



バージョン 2 用ユーザーガイド

AWS Command Line Interface



AWS Command Line Interface: バージョン 2 用ユーザーガイド

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

Table of Contents

.....	xv
について AWS CLI	1
AWS CLI バージョン 2 について	1
SDK メジャーバージョンのメンテナンスとサポート	2
アマゾン ウェブ サービスについて	2
の例について	2
その他のドキュメントとリソース	4
AWS CLI ドキュメントとリソース	4
その他 AWS SDKs および ツール	4
使用を開始する	6
前提条件	7
IAM または IAM Identity Center 管理アカウントを作成する	7
次のステップ	8
インストール/更新	8
AWS CLI インストールと更新の手順	9
AWS CLI インストールとアンインストールのエラーのトラブルシューティング	29
次のステップ	29
過去のリリース	30
AWS CLI インストールおよびアンインストールエラーのトラブルシューティング	49
次のステップ	49
ソースからのビルドとインストール	49
ソースからビルドする理由	50
クイックステップ	51
ステップ 1: すべての要件の設定	53
ステップ 2: AWS CLI ソースインストールの構成	58
ステップ 3: AWS CLI の構築	64
ステップ 4: AWS CLI のインストール	65
ステップ 5: インストールの検証	67
ワークフローの例	67
AWS CLI インストールおよびアンインストールエラーのトラブルシューティング	70
次のステップ	70
Amazon ECR Public/Docker	71
前提条件	71
Amazon ECR Public と Docker Hub 間の決定	71

公式イメージを実行する	72
公式イメージのインターフェイスと下位互換性に関する注意事項	73
特定のバージョンとタグの使用	73
最新の公式イメージに更新する	74
ホストファイル、認証情報、環境変数、構成を共有する	75
docker run コマンドの短縮	81
セットアップ	84
プログラムによるアクセス用の認証情報を収集する	84
新しい設定と認証情報のセットアップ	86
既存の設定と認証情報ファイルの使用	94
の設定 AWS CLI	95
設定と認証情報の優先順位	95
このセクションのその他のトピック	96
の設定と認証情報ファイルの設定 AWS CLI	97
設定ファイルと認証ファイルの形式	97
構成設定はどこに保存されていますか?	106
名前を指定されたプロファイルを使用する	107
コマンドを使用した構成設定の設定と表示	107
新しい設定と認証情報を設定するコマンドの例	110
サポートされる config ファイル設定	113
環境可変	132
環境変数の設定方法	132
AWS CLI サポートされている環境変数	134
のコマンドラインオプション AWS CLI	144
コマンドラインオプションの使用方法	145
AWS CLI サポートされているグローバルコマンドラインオプション	145
コマンドラインオプションの一般的な用途	150
でのコマンド完了の設定 AWS CLI	151
仕組み	151
Linux または macOS でのコマンド補完の設定。	152
Windows でのコマンド補完の設定。	155
再試行	157
使用可能な再試行モード	157
再試行モードの設定	160
再試行ログの表示	161
のHTTPプロキシの使用 AWS CLI	162

例の使用	162
プロキシを認証する	163
Amazon EC2インスタンスでのプロキシの使用	164
トラブルシューティング	164
エンドポイント	164
1つのコマンドのエンドポイントを設定する	165
すべての のグローバルエンドポイントを設定する AWS のサービス	165
すべての にFIPsエンドポイントを使用するように を設定する AWS のサービス	167
すべての AWS のサービスでデュアルスタックのエンドポイントを使用するように設定	168
サービス固有のエンドポイントを設定します。	169
エンドポイント設定と設定の優先順位	173
認証とアクセス認証情報	174
設定と認証情報の優先順位	175
このセクションのその他のトピック	176
IAM Identity Center 認証	176
前提条件	177
aws configure sso ウィザードでプロファイルを設定する	178
aws configure sso-session ウィザードで sso-session セクションのみを設定する	181
config ファイルを使用した手動設定	181
IAM Identity Center セッションにサインインする	184
IAM Identity Center プロファイルでコマンドを実行する	185
IAM Identity Center セッションからサインアウトする	185
トラブルシューティング	186
関連リソース	186
IAM Identity Center の概念	186
チュートリアル: AWS IAM Identity Center および Amazon S3	190
短期の認証情報	196
IAM ロール	197
前提条件	198
IAM ロールの使用の概要	198
ロールの設定と使用	199
の使用 MFA	201
クロスアカウントロールと外部 ID	203
監査を容易にするためのロールセッション名の指定	203
ウェブ ID を使用したロールの継承	204

キャッシュされた認証情報のクリア	206
IAM ユーザー	206
ステップ 1: IAM ユーザーを作成する	207
ステップ 2: アクセスキーを取得する	207
の設定 AWS CLI	207
Amazon EC2 インスタンスメタデータを の認証情報として使用する AWS CLI	210
前提条件	210
Amazon EC2 メタデータのプロファイルの設定	211
外部認証情報	212
の使用 AWS CLI	215
ヘルプが必要な場合	216
組み込みの AWS CLI help コマンド	216
AWS CLI リファレンスガイド	221
API ドキュメント	221
エラーのトラブルシューティング	222
その他のヘルプ	222
コマンド構造	222
コマンド構造	222
Wait コマンド	223
パラメータ値を指定する	225
一般的なパラメータタイプ	226
文字列を含む引用符	231
ファイルからのパラメータ	235
CLI スケルトンテンプレートを生成する	238
短縮構文	250
自動プロンプト	253
仕組み	253
自動プロンプト機能	254
自動プロンプトモード	257
自動プロンプトの設定	257
コマンド出力を制御する	258
機密出力	258
サーバー側の出力オプションとクライアント側の出力オプション	259
[Output Format] (出力形式)	259
ページ分割	269
の出力をフィルタリングする	274

リターンコード	298
ウィザード	300
仕組み	300
エイリアス	302
前提条件	302
ステップ 1: エイリアスファイルを作成する	302
ステップ 2: エイリアスを作成する	303
ステップ 3: エイリアスを呼び出す	307
エイリアスリポジトリの例	309
リソース	310
コードの例	311
ガイド付きコマンドの例	311
DynamoDB	312
Amazon EC2	316
S3 Glacier	335
IAM	342
Amazon S3	346
Amazon SNS	366
コマンド例	368
ACM	375
API ゲートウェイ	386
API ゲートウェイHTTPと WebSocket API	445
API ゲートウェイ管理 API	492
App Mesh	495
App Runner	540
AWS AppConfig	575
Application Auto Scaling	611
Application Discovery Service	629
AppRegistry	635
Athena	647
Auto Scaling	682
Auto Scaling Plans	753
AWS Backup	761
AWS Batch	767
AWS Budgets	782
Amazon Chime	793

クラウドコントロール API	867
AWS Cloud Map	874
AWS Cloud9	884
AWS CloudFormation	893
CloudFront	944
Amazon CloudSearch	1014
CloudTrail	1015
CloudWatch	1032
CloudWatch ログ	1047
CloudWatch ネットワークモニタリング	1053
CodeArtifact	1066
CodeBuild	1095
CodeCommit	1161
CodeDeploy	1237
CodeGuru レビューアー	1279
CodePipeline	1298
AWS CodeStar 通知	1331
CodeConnections	1342
Amazon Cognito ID	1351
Amazon Cognito Identity Provider	1357
Amazon Comprehend	1427
Amazon Comprehend Medical	1566
AWS Config	1602
Amazon Connect	1626
AWS Cost and Usage Report	1643
Cost Explorer サービス	1646
Firehose	1654
Amazon Data Lifecycle Manager	1658
AWS Data Pipeline	1664
DataSync	1674
DAX	1678
検出	1698
Device Farm	1710
AWS Direct Connect	1715
AWS Directory Service	1767
AWS DMS	1770

Amazon DocumentDB	1814
DynamoDB	1872
DynamoDB Streams	1970
Amazon EC2	1977
Amazon EC2 Instance Connect	2655
Amazon ECR	2656
Amazon ECR Public	2688
Amazon ECS	2696
Amazon EFS	2783
Amazon EKS	2791
Elastic Beanstalk	2870
Elastic Load Balancing - バージョン 1	2901
Elastic Load Balancing - バージョン 2	2930
Elastic Transcoder	2985
ElastiCache	3013
MediaStore	3120
Amazon EMR	3138
EMR での Amazon EKS	3187
EventBridge	3189
Firewall Manager	3195
AWS FIS	3206
Amazon GameLift	3225
Global Accelerator	3259
AWS Glue	3299
GuardDuty	3321
AWS Health	3340
HealthImaging	3347
HealthLake	3375
HealthOmics	3386
IAM	3457
IAM Access Analyzer	3601
Image Builder	3638
Incident Manager	3680
Incident Manager の連絡先	3703
Amazon Inspector	3727
AWS IoT	3772

AWS IoT 1-Click デバイス	3959
AWS IoT 1-Click プロジェクト	3969
AWS IoT Analytics	3981
Device Advisor	4009
AWS IoT data	4024
AWS IoT Events	4027
AWS IoT Events-Data	4053
AWS IoT Greengrass	4080
AWS IoT Greengrass V2	4168
AWS IoT Jobs SDK release	4194
AWS IoT SiteWise	4198
AWS IoT Things Graph	4249
AWS IoT Wireless	4277
Amazon IVS	4315
Amazon IVS Chat	4356
Amazon IVS リアルタイムストリーミング	4369
Amazon Kendra	4400
Kinesis	4409
AWS KMS	4429
Lake Formation	4499
Lambda	4552
License Manager	4595
Lightsail	4609
Macie	4736
Amazon Managed Grafana	4742
MediaConnect	4743
MediaConvert	4760
MediaLive	4785
MediaPackage	4792
MediaPackage VOD	4807
MediaStore データプレーン	4820
MediaTailor	4825
MemoryDB	4831
Amazon MSK	4869
Network Manager	4878
Nimble Studio	4917

OpenSearch サービス	4935
AWS OpsWorks	4950
AWS OpsWorks CM	5009
組織	5024
AWS Outposts	5063
AWS Payment Cryptography	5068
AWS Payment Cryptography データプレーン	5089
Amazon Pinpoint	5098
Amazon Polly	5122
AWS の料金表	5129
AWS Private CA	5134
AWS Proton	5142
QLDB	5154
Amazon RDS	5178
Amazon RDS Data Service	5379
Amazon RDS Performance Insights	5383
Amazon Redshift	5387
Amazon Rekognition	5469
AWS RAM	5547
Resource Explorer	5571
リソースグループ	5594
Resource Groups のタグ付け API	5608
AWS RoboMaker	5612
Route 53	5649
Route 53 ドメイン登録	5664
Route 53 プロファイル	5690
Route 53 Resolver	5703
Amazon S3	5748
Amazon S3 コントロール	5844
S3 Glacier	5860
Secrets Manager	5883
Security Hub	5913
Security Lake	5993
AWS Serverless Application Repository	6029
Service Catalog	6031
Service Quotas	6063

Amazon SES	6074
Shield	6087
Signer	6103
Snowball	6113
Amazon SNS	6115
Amazon SQS	6137
Storage Gateway	6159
AWS STS	6162
AWS Support	6171
Amazon SWF	6185
Systems Manager	6201
Amazon Textract	6379
Amazon Transcribe	6391
Amazon Translate	6434
Trusted Advisor	6436
Verified Permissions	6456
VPC 格子	6483
AWS WAF Classic	6512
AWS WAF Classic Regional	6517
AWS WAFV2	6524
Amazon WorkDocs	6570
Amazon WorkMail	6603
Amazon WorkMail メッセージフロー	6628
WorkSpaces	6630
X-Ray	6646
Bash スクリプトの例	6664
DynamoDB	6665
Amazon EC2	6737
HealthImaging	6844
IAM	6853
Amazon S3	6909
AWS STS	6933
セキュリティ	6936
データ保護	6937
データ暗号化	6938
Identity and Access Management	6938

対象者	6939
アイデンティティを使用した認証	6939
ポリシーを使用したアクセスの管理	6943
での の AWS のサービス 仕組み IAM	6945
AWS ID とアクセスのトラブルシューティング	6945
コンプライアンス検証	6947
耐障害性	6949
インフラストラクチャセキュリティ	6949
最小TLSバージョンの強制	6950
エラーのトラブルシューティング	6952
最初に試す一般的なトラブルシューティング	6952
AWS CLI コマンドのフォーマットを確認する	6953
AWS CLI コマンドが を使用している AWS リージョン ことを確認します。	6953
最新バージョンの AWS CLI を実行していることを確認する	6954
--debug オプションを使用する	6954
AWS CLI コマンド履歴ログを有効にして確認する	6960
AWS CLI が設定されていることを確認します。	6960
コマンドが見つからないエラー	6961
「aws --version」コマンドが、インストールしたのとは異なるバージョンを返す	6964
aws --version 「」コマンドは、AWS CLI	6965
は、パラメータ名が不完全なコマンド AWS CLI を処理しました	6966
アクセス拒否エラー	6967
無効な認証情報とキーエラー	6968
署名がエラーと一致しない	6970
SSL 証明書エラー	6971
無効なJSONエラー	6973
追加リソース	6975
移行ガイド	6976
新機能と変更点	6976
AWS CLI バージョン 2 の新機能	6977
AWS CLI バージョン 1 と AWS CLI バージョン 2 の変更のブレーク	6978
移行手順	6986
バージョン 1 をバージョン 2 に置き換える	6986
Side-by-side インストール	6987
アンインストール	6989
AWS CLI インストールおよびアンインストールエラーのトラブルシューティング	6993

ドキュメント履歴	6994
----------------	------

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。

AWS Command Line Interface とはどのようなものですか。

AWS Command Line Interface (AWS CLI) は、コマンドラインシェルのコマンドを使用して AWS サービスとやり取りできるオープンソースツールです。最小限の設定で、を使用すると、AWS CLI ターミナルプログラムのコマンドプロンプト AWS Management Console から、ブラウザベースの提供される機能と同等の機能を実装するコマンドの実行を開始できます。

- Linux シェル - Linux または macOS でコマンドを実行するには [bash](#)、[zsh](#)、[tcsh](#) などの一般的なシェルプログラムを使用します。
- Windows コマンドライン - Windows では、Windows コマンドプロンプトまたは でコマンドを実行します PowerShell。
- リモート - PuTTY や などのリモートターミナルプログラム、または を使用して SSH、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスでコマンドを実行します AWS Systems Manager。

のすべての IaaS (サービスとしてのインフラストラクチャ) AWS のガバナンス、管理、アクセス機能は AWS Management Console、および で AWS API 利用できます AWS CLI。新しい AWS IaaS 機能とサービスは、および起動 CLI 時 API、または起動から 180 日以内に完全な AWS Management Console 機能を提供します。

AWS CLI は、APIs AWS サービスの一般ユーザーに直接アクセスできるようにします。を使用してサービスの機能を調べ AWS CLI、シェルスクリプトを開発して リソースを管理できます。低レベルの API と同等のコマンドに加えて、いくつかの AWS サービスでは をカスタマイズできます AWS CLI。カスタマイズには、複雑な でサービスの使用を簡素化する高レベルのコマンドを含めることができます API。

AWS CLI バージョン 2 について

AWS CLI バージョン 2 は の最新のメジャーバージョン AWS CLI であり、すべての最新機能をサポートしています。バージョン 2 で導入された一部の機能は、バージョン 1 との下位互換性がないため、これらの機能にアクセスするには、アップグレードする必要があります。バージョン 1 には、スクリプトを変更することが必要となる場合がある「互換性を損なう」変更があります。バージョン 2 の変更点の一覧については、「[AWS CLI バージョン 1 から AWS CLI バージョン 2 への移行](#)」をご参照ください。

AWS CLI バージョン 2 は、バンドルされたインストーラとしてのみインストールできます。パッケージマネージャーに表示される場合がありますが、これらはサポートされていない非公式パッケー

ジであり、によって作成または管理されません AWS。このガイドに記載されているように、公式の AWS 配布ポイント AWS CLI からのみ をインストールすることをお勧めします。

AWS CLI バージョン 2 をインストールするには、「」を参照してください [the section called “インストール/更新”](#)。

現在インストールされているバージョンを確認するには、以下のコマンドを使用します。

```
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

バージョン履歴については、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してください [GitHub](#)。

SDK メジャーバージョンのメンテナンスとサポート

SDK メジャーバージョンとその基盤となる依存関係のメンテナンスとサポートについては、「[AWS SDKs および ツール リファレンス ガイド](#)」の以下を参照してください。

- [AWS SDKs および ツールのメンテナンスポリシー](#)
- [AWS SDKs および ツールのバージョンサポートマトリックス](#)

アマゾン ウェブ サービスについて

アマゾン ウェブ サービス (AWS) は、デベロッパーがアプリケーションの開発時に利用できるデジタルインフラストラクチャサービスの集合体です。サービスには、コンピューティング、ストレージ、データベース、アプリケーションの同期 (メッセージングとキューイング) が含まれます。は pay-as-you-go サービスモデル AWS を使用します。料金が発生するのは、ユーザー (すなわちユーザーのアプリケーション) が実際に使用したサービスの分のみです。また、プロトタイピングと実験のプラットフォームとして AWS より親しみやすくするために、[は無料利用枠 AWS を提供しています](#)。この枠では、サービスを利用しても一定のレベル以下であれば無料です。AWS コストと 無料利用枠の詳細については、「[AWS 無料利用枠](#)」を参照してください。AWS アカウントを取得するには、[AWS ホームページ](#)を開き、AWS アカウントの作成 を選択します。

AWS CLI ユーザーガイドの例について

このガイドの AWS Command Line Interface (AWS CLI) 例は、次の規則を使用してフォーマットされています。

- プロンプト – コマンドプロンプトは Linux プロンプトを使用し、(\$) として表示されます。Windows 固有のコマンドの場合、C:\> がプロンプトとして使用されます。コマンドを入力した場合はプロンプトを含めないでください。
- ディレクトリ – 特定のディレクトリからコマンドを実行する必要がある場合は、プロンプト記号の前にディレクトリ名が表示されます。
- ユーザー入力 – コマンドラインに入力するコマンドテキストは、**user input** としてフォーマットされます。
- 置換可能なテキスト – 選択したリソースの名前、またはコマンドに含める必要がある AWS サービスによってIDs生成された可変テキストは、次のようにフォーマットされます。**replaceable text**。特定のキーボード入力が必要な複数行のコマンドまたはコマンドでは、キーボードコマンドを置き換え可能なテキストとして表示することもできます。
- 出力 – AWS サービスによって返される出力は、ユーザー入力の下に表示され、としてフォーマットされます **computer output**。

以下の **aws configure** コマンド例は、ユーザー入力、置き換え可能なテキスト、および出力を表しています。

1. コマンドラインに **aws configure** を入力し、Enter キーを押します。
2. はテキスト行を AWS CLI 出力し、追加情報の入力を求めます。
3. 各アクセスキーを順に入力して、[Enter] キーを押します。
4. 次に、表示された形式で AWS リージョン名を入力し、Enter を押してから、Enter a final time を押して出力形式設定をスキップします。
5. 最後の Enter コマンドは、その行にユーザー入力がないため、置き換え可能なテキストとして表示されます。

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: ENTER
```

次の例は、簡単なコマンドと出力を示しています。この例を使用するには、コマンドのフルテキスト (プロンプトの後の強調表示されたテキスト) を入力し、Enter キーを押します。セキュリティグループの名前。 **my-sg** は、目的のセキュリティグループ名に置き換えることができます。中括弧を含む

JSONドキュメントが出力されます。をテキスト形式またはテーブル形式で出力CLIするように設定した場合、出力の形式は異なります。[JSON](#) はデフォルトの出力形式です。

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group"
{
  "GroupId": "sg-903004f8"
}
```

に関する追加のドキュメントとリソース AWS CLI

AWS CLI ドキュメントとリソース

このユーザーガイドに加えて、を使用する際の貴重なオンラインリソースを以下に示します AWS CLI。

- [AWS CLI バージョン 2 リファレンスガイド](#)
- [AWS CLI Bash スクリプティングコード例リポジトリ](#)。オープンソースの bash スクリプティングの例。Bash スクリプトの例は、の [AWS Code Examples Repository](#) でホストされています GitHub。
- [AWS CLI GitHub リポジトリ](#)。AWS CLI のソースコードを表示およびフォークできます GitHub。ユーザーのコミュニティに参加GitHubして、フィードバックの提供、機能のリクエスト、独自の投稿の送信を行います。これには、AWS CLI ドキュメントのコマンド例の表示と提供が含まれます。
- [AWS CLI エイリアスサンプルリポジトリ](#) で AWS CLI エイリアスサンプルを表示およびフォークできます GitHub。
- [AWS CLI バージョン 2 Changelog](#)

その他 AWS SDKs および ツール

ユースケースによっては、ニーズにより適した AWS SDKs または ツールのいずれかを選択することもできます。

- [AWS SDKs および ツールリファレンスガイド](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)

- [AWS SDK for JavaScript](#)
- [AWS SDK for Kotlin](#)
- [AWS SDK for .NET](#)
- [AWS SDK for Python \(Boto\)](#)
- [AWS SDK for PHP](#)
- [AWS Tools for PowerShell](#)
- [AWS SDK for Ruby](#)
- [AWS SDK for Rust](#)
- [AWS SDK for SAP ABAP](#)
- [AWS SDK for Swift](#)
- [AWS Amplify](#)

の開始方法 AWS CLI

この章では、AWS Command Line Interface (AWS CLI) のバージョン 2 を開始する手順と、関連する手順へのリンクについて説明します。

1. [すべての前提条件を満たす](#) - で AWS サービスにアクセスするには AWS CLI、少なくとも AWS アカウントと IAM 認証情報が必要です。AWS アカウントのセキュリティを強化するため、ルートアカウントの認証情報を使用しないことをお勧めします。AWS で実行するタスクへのアクセス認証情報を提供するには、最小特権を持つユーザーを作成する必要があります。
2. 次のいずれかの方法 AWS CLI を使用して、をインストールまたはアクセスします。
 - (推奨) [the section called “インストール/更新”](#)。
 - [the section called “過去のリリース”](#)。特定バージョンのインストールは、チームで特定バージョンのツールが必要な場合に行います。
 - [the section called “ソースからのビルドとインストール”](#)。ソース AWS CLI から GitHub を構築することは、構築済みのインストーラーで直接サポートしていないプラットフォームで作業するお客様が主に使用する、より詳細な方法です。
 - [the section called “Amazon ECR Public/Docker”](#)。
 - を使用して、ブラウザから AWS コンソールで AWS CLI バージョン 2 にアクセスします AWS CloudShell。詳細については、[AWS CloudShell ユーザーガイド](#)をご参照ください。
3. [にアクセスしたら AWS CLI、を初めて使用するために IAM 認証情報で を設定します AWS CLI](#)。

i インストーラまたは設定エラーのトラブルシューティング
のインストール、アンインストール、または設定後に問題が発生した場合は、トラブルシューティングの手順[エラーのトラブルシューティング](#)については AWS CLI、「」を参照してください。

トピック

- [AWS CLI バージョン 2 を使用するための前提条件](#)
- [の最新バージョンへのインストールまたは更新 AWS CLI](#)
- [AWS CLI バージョン 2 の過去のリリースのインストール](#)
- [ソース AWS CLI からの の構築とインストール](#)

- [の公式 Amazon ECR Public または Docker イメージの実行 AWS CLI](#)
- [のセットアップ AWS CLI](#)

AWS CLI バージョン 2 を使用するための前提条件

を使用して AWS サービスにアクセスするには AWS CLI、AWS アカウントと IAM 認証情報が必要です。AWS CLI コマンドを実行する場合、はこれらの AWS 認証情報にアクセスできる AWS CLI 必要があります。AWS アカウントのセキュリティを強化するには、ルートアカウントの認証情報を使用しないことをお勧めします。で実行するタスクへのアクセス認証情報を提供する最小権限を持つユーザーを作成する必要があります AWS。

トピック

- [IAM または IAM Identity Center 管理アカウントを作成する](#)
- [次のステップ](#)

IAM または IAM Identity Center 管理アカウントを作成する

を設定する前に AWS CLI、IAM または IAM Identity Center アカウントを作成する必要があります。

管理者ユーザーを作成するには、以下のいずれかのオプションを選択します。

管理者を管理する方法を 1 つ選択します	目的	方法	以下の操作も可能
IAM Identity Center で (推奨)	短期の認証情報を使用して AWS にアクセスします。 これはセキュリティのベストプラクティスと一致しています。ベストプラクティ	AWS IAM Identity Center ユーザーガイドの「 開始方法 」の手順に従います。	AWS Command Line Interface ユーザーガイド で使用する AWS CLI ようにを設定 AWS IAM Identity Center して、プログラムによるアクセスを設定します。

管理者を管理する方法を1つ選択します	目的	方法	以下の操作も可能
	<p>の詳細については、IAM「ユーザーガイド」の「セキュリティのベストプラクティスIAM」を参照してください。</p>		
で IAM (非推奨)	<p>長期認証情報を使用して AWS にアクセスする。</p>	<p>IAM ユーザーガイドの「最初のIAM管理者ユーザーとユーザーグループの作成」の手順に従います。</p>	<p>IAM ユーザーガイドの IAM「ユーザーのアクセスキーを管理する」を使用して、プログラムによるアクセスを設定します。</p>

次のステップ

AWS アカウント および IAM 認証情報を作成した後、を使用するには、次のいずれか AWS CLI を実行できます。

- AWS CLI バージョン 2 の最新 [リリース](#) をコンピュータにインストールします。
- お使いのコンピュータに AWS CLI バージョン 2 の [過去のリリースをインストールする](#)。
- Docker イメージを使用して、コンピュータから AWS CLI バージョン 2 にアクセスします。 [???](#)
- を使用して、ブラウザから AWS コンソールで AWS CLI バージョン 2 にアクセスします AWS CloudShell。詳細については、AWS CloudShell ユーザーガイドを参照してください。

の最新バージョンへのインストールまたは更新 AWS CLI

このトピックでは、サポートされているオペレーティングシステムに AWS Command Line Interface (AWS CLI) の最新リリースをインストールまたは更新する方法について説明します。の最新リリースについては AWS CLI、[AWS CLI のバージョン 2 の変更ログ](#) を参照してください GitHub。

の過去のリリースをインストールするには AWS CLI、「」を参照してください[the section called “過去のリリース”](#)。アンインストール手順については、「[アンインストール](#)」を参照してください。

⚠ Important

AWS CLI バージョン 1 と 2 は同じawsコマンド名を使用します。以前に AWS CLI バージョン 1 をインストールした場合は、「」を参照してください[AWS CLI バージョン 1 から AWS CLI バージョン 2 への移行](#)。

トピック

- [AWS CLI インストールと更新の手順](#)
- [AWS CLI インストールとアンインストールのエラーのトラブルシューティング](#)
- [次のステップ](#)

AWS CLI インストールと更新の手順

インストール手順については、お使いのオペレーティングシステムのセクションを展開します。

Linux

インストールと更新の要件

- ダウンロードしたパッケージを抽出または解凍できる必要があります。オペレーティングシステムに組み込み unzip コマンドがない場合は、同等のコマンドを使用します。
- は、glibc、groff、および AWS CLI を使用しますless。これらは、Linux のほとんどの主要なディストリビューションにデフォルトで含まれています。
- CentOS AWS CLI 、Fedora、Ubuntu、Amazon Linux 1、Amazon Linux 2、Amazon Linux 2023、および Linux の最近のディストリビューションの 64 ビットバージョンで をサポートしています ARM。
- AWS は 以外のサードパーティーリポジトリを保持していないためsnap、最新バージョンの が含まれていることを保証することはできません AWS CLI。

のインストールまたは更新 AWS CLI

Warning

Amazon Linux で初めて更新する場合は、の最新バージョンをインストールするには AWS CLI、次のコマンドを使用してプリインストールされたyumバージョンをアンインストールする必要があります。

```
$ sudo yum remove awscli
```

yum のインストール AWS CLI が削除されたら、以下の Linux インストール手順に従ってください。

をインストールするには AWS CLI、次のいずれかの方法を使用します。

- コマンドラインインストーラは、インストールするバージョンを指定できるため、バージョン管理に適しています。このオプションは自動更新されません。以前のバージョンを上書きするには、更新するたびに新しいインストーラをダウンロードする必要があります。
- 公式にサポートされているsnapパッケージは、スナップパッケージが自動的に更新 AWS CLI されるため、常にの最新バージョンを持つのに適しています。のマイナーバージョンを選択するための組み込みサポートがないため AWS CLI、チームがバージョンを固定する必要がある場合は最適なインストール方法ではありません。

Command line installer - Linux x86 (64-bit)

の現在のインストールを更新するには AWS CLI、更新するたびに新しいインストーラをダウンロードして以前のバージョンを上書きします。コマンドラインから以下の手順に従って、Linux AWS CLI に をインストールします。

以下は、基本的なインストールを提供する単一のコピーアンドペーストグループでの簡単なインストール手順です。ガイド付き手順については、以下のステップを参照してください。

Note

(オプション) 次のコマンドブロックは、ダウンロードの整合性を最初に検証 AWS CLI することなく、 をダウンロードしてインストールします。ダウンロードの整合性を検証するには、以下の詳しい手順に従ってください。

AWS CLIをインストールするには、次のコマンドを実行します。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

AWS CLIの現在のインストールを更新するには、既存のシンボリックリンクとインストーラの情報を追加して `--bin-dir`、`--install-dir`、および `--update` パラメータで `install` コマンドを作成します。次のコマンドブロックは、のシンボリックリンクの例を使用しています。/`usr/local/bin` および のインストーラの場所の例 `/usr/local/aws-cli`。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
```

ガイド付きインストール手順

1. 次のいずれかの方法でインストールファイルをダウンロードします。

- `curl` コマンドの使用 `--o` オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。以下のコマンドの例のオプションを使用すると、ダウンロードしたファイルが現在のディレクトリにローカル名 `awscliv2.zip` で書き込まれます。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

- からのダウンロード URL - ブラウザでインストーラをダウンロードするには、次のを使用しますURL。 https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip

2. (オプション) ダウンロードした zip ファイルの整合性を確認する

上記のステップ .zip で AWS CLI インストーラパッケージを手動でダウンロードすることを選択した場合は、以下のステップを使用して、GnuPG ツールを使用して署名を検証できます。

AWS CLI インストーラパッケージ .zip ファイルは、PGP 署名を使用して暗号的に署名されます。ファイルの破損や変更がある場合、この検証は失敗します。その場合、インストールを続行しないでください。

- a. ご利用のパッケージマネージャーを使用し、`gpg` コマンドをダウンロードしてインストールします。GnuPG の詳細については、[GnuPG のウェブサイト](#)を参照してください。
- b. パブリックキーファイルを作成するには、テキストファイルを作成し、次のテキストに貼り付けます。

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgus0Jl7ENSyumXh85z0TRV0xJorM2B/JL0kH0yigQ1uUG
ZMLhENAG0bYatdrKP+3H91lvK050pXwn0/R7fB/FSTouki4ciIx50uLlnJZIxSzx
PqG10mkxImLNBGwoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+Lb1HAgEt5G
TfnxEKJ8soPLYWmwDH6HWcnjZ/aIQRBTIQ05uVeEoYxSh6w0ai7ss/KveoSNBbYz
gbdzoqI2Y8cgh2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU00UCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcqJ0z3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3Tqgvdx4bdkhf5cH+7NtW0
lrFj6UwAsGukBTA0xC0l/dnSmZhJ7Z1KmEwilro/g0rjt0xqRQut1IqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZwC4XKCa3mprEGC3IbJTBFqglXmZ719yWG
EEUJY01b2XrSuPwml39beWdKM8kzr10jnl0m6+lpTRCBfo0wa9F8YZRhHPAkWkKX
XDe0GpWRj4oh0x0d2GWkyV5xyN14p2tQ0Cd00Dmz80yUTgRpPVQUt0EhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbG1AYW1hem9uLmNvbT6JAlQEewEIAD4CGwMF
CwkIBwIGFQoJCAcCBYCAwEChgECF4AWIQT7Xbd/1cEYUaurraimMQrMRnJHXAUC
ZqFYbwUJCv/c0gAKCRCmMQrMRnJHXKYuEAC+wtZ611qQt0l0t5spM9SWZuszbcyA
0xBAJq2pncnp6wdC0kuAPu4/R3UCIoD2C49MkLj9Y0Yvue8CCF60IJ8L+fKBv2DI
yWZGmHL0p9wa/X8NCKQrKxK1gq5PuCzi3f3SqwfbZuZGeK/ubnmtttWXPuUu/Iz
VR0u/0sAy3j4uTGKh2cX7XnZbSqqJhUk9H324mIjISwzvw1Ker6xtH/LwdBeJCck
bVBdh3LZis4zuD4IzeB01vRvjot30q4xadUv5RSPATg7T1kivrtLCnwwqc6L4LnF
00kNysk94L3LQSHyQW2kQS1cVwr+yGUSiSp+VvMbAobAapmMJWP6e/dKYAUGIX6+
2waLdbBs2U7MXznx/2ayCLPH7qCY9cenbdj5JhG9ibVvFWqqhSo22B/URQE/CMrG
+3xXwtHEBoMyWEATr1tWwn2yyQGbkUGANneSDFiTFeoQvKNyyCFTF01F2XKCCuDs
19nj34PE2TJi1TG2QR1Mr4D0NgwLLAMg2Los1CK6nXWnImYHKuaKS9LVaCoC8vu7
IRBik1NX6SjrQnftk0M9dY+s0ZbAN1gbdjZ8H3qlbl/4TxMdr87m8LP4FZIIo261
Eycv34pVkcPzPiP+dgamEiQJ7IL4Zario9mv6HbDGV6mLY45+16/0EzCwkI5IyIf
BfWC9s/USgxchg==
=ptgS
-----END PGP PUBLIC KEY BLOCK-----
```

参考までに、パブリックキーの詳細を以下に示します。

```
Key ID:          A6310ACC4672475C
Type:           RSA
Size:           4096/4096
```

```
Created:          2019-09-18
Expires:         2025-07-24
User ID:         AWS CLI Team <aws-cli@amazon.com>
Key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

- c. 次のコマンドを使用して AWS CLI パブリックキーをインポートし、置き換えます。 *public-key-file-name* 作成したパブリックキーのファイル名。

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
imported
gpg: Total number processed: 1
gpg:             imported: 1
```

- d. ダウンロードしたパッケージ AWS CLI の署名ファイルをダウンロードします。これは、対応する .zip ファイルと同じパスと名前を持っていますが、拡張子は .sig です。次の例では、現在のディレクトリに awscliv2.sig という名前のファイルとして保存します。

の最新バージョンでは AWS CLI、次のコマンドブロックを使用します。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip.sig
```

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 *2.0.30* は次のコマンド awscli-exe-linux-x86_64-*2.0.30*.zip.sig になります。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip.sig
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

- e. 署名を検証し、.sig ファイル名と .zip ファイル名の両方をパラメータとして gpg コマンドに渡します。

```
$ gpg --verify awscliv2.sig awscliv2.zip
```

出力は以下の例のようになります。

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC
    4672 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the
    owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511  ADA8 A631 0ACC 4672 475C
```

⚠ Important

出力内の警告は予想されており、問題を示すものではありません。これは、パーソナルPGPキー (持っている場合) とキーの間に信頼チェーンがないために AWS CLI PGP発生します。詳細については、「[信用の輪 \(Web of Trust\)](#)」を参照してください。

3. インストーラを解凍します。Linux ディストリビューションに組み込み unzip コマンドがない場合は、同等のコマンドを使用して解凍します。次のコマンド例では、パッケージを解凍し、現在のディレクトリの下に aws という名前のディレクトリを作成します。

```
$ unzip awscliv2.zip
```

i Note

以前のバージョンからアップデートする場合、unzip コマンドでは、既存のファイルを上書きするように求められます。スクリプトのオートメーションなどでこのプロンプトをスキップする場合は、unzip の `-u` 更新フラグを使用します。このフラグは、既存のファイルを自動的に更新し、必要に応じて新しいファイルを作成します。

```
$ unzip -u awscliv2.zip
```

4. インストールプログラムを実行します。インストールコマンドは、新しく解凍された `install` ディレクトリにある `aws` という名前のファイルを使用します。デフォルトでは、すべてのファイルが `/usr/local/aws-cli` にインストールされ、シンボリックリンクが /

`usr/local/bin` に作成されます。コマンドには、これらのディレクトリへの書き込み権限を付与するための `sudo` が含まれます。

```
$ sudo ./aws/install
```

すでに書き込み権限を持つディレクトリを指定した場合は、`sudo` なしでインストールできます。次の `install` コマンド用の手順を使用して、インストール場所を指定します。

- `-i` および `-b` パラメータに指定するパス内のボリューム名やディレクトリ名に、スペース文字などの空白文字が含まれないことを確認します。スペースが含まれていると、インストールは失敗します。
- `--install-dir` または `-i` — このオプションは、すべてのファイルのコピー先ディレクトリを指定します。

デフォルト値は `/usr/local/aws-cli` です。

- `--bin-dir` または `-b` — このオプションは、インストールディレクトリのメイン `aws` プログラムが、指定されたパスのファイル `aws` にシンボリックにリンクされることを指定します。指定したディレクトリへの書き込み権限が必要です。パスにすでに存在するディレクトリへのシンボリックリンクを作成すると、インストールディレクトリをユーザーの `$PATH` 変数に追加する必要がなくなります。

デフォルト値は `/usr/local/bin` です。

```
$ ./aws/install -i /usr/local/aws-cli -b /usr/local/bin
```

Note

の現在のインストールを更新するには AWS CLI、既存の `symlink` とインストーラ情報を追加して、`--update` パラメータを使用して `install` コマンドを構築します。

```
$ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
```

既存のシンボリックリンクとインストールディレクトリを検索するには、次の手順に従います。

1. `which` コマンドを使用して、シンボリックリンクを見つけます。これにより、`--bin-dir` パラメータで使用するパスが指定されます。

```
$ which aws
/usr/local/bin/aws
```

2. `ls` コマンドを使用して、シンボリックリンクが指すディレクトリを見つけます。これにより、`--install-dir` パラメータで使用するパスが指定されます。

```
$ ls -l /usr/local/bin/aws
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/aws-cli/v2/current/bin/aws
```

5. 以下のコマンドを使用して、インストールを確認します。

```
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/2.4.5
```

`aws` コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Command line - Linux ARM

の現在のインストールを更新するには AWS CLI、更新するたびに新しいインストーラをダウンロードして以前のバージョンを上書きします。コマンドラインから以下の手順に従って、Linux AWS CLI に をインストールします。

以下は、基本的なインストールを提供する単一のコピーアンドペーストグループでの簡単なインストール手順です。ガイド付き手順については、以下のステップを参照してください。

Note

(オプション) 次のコマンドブロックは、ダウンロードの整合性を最初に検証 AWS CLI することなく、 をダウンロードしてインストールします。ダウンロードの整合性を検証するには、以下の詳しい手順に従ってください。

をインストールするには AWS CLI、次のコマンドを実行します。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

AWS CLIの現在のインストールを更新するには、既存のシンボリックリンクとインストーラの情報を追加して `--bin-dir`、`--install-dir`、および `--update` パラメータで `install` コマンドを作成します。次のコマンドブロックは、のシンボリックリンクの例を使用しています。 `/usr/local/bin` および のインストーラの場所の例 `/usr/local/aws-cli`。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --update
```

ガイド付きインストール手順

1. 次のいずれかの方法でインストールファイルをダウンロードします。

- `curl` コマンドの使用 `--o` オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。以下のコマンドの例のオプションを使用すると、ダウンロードしたファイルが現在のディレクトリにローカル名 `awscliv2.zip` で書き込まれます。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o "awscliv2.zip"
```

- からのダウンロード URL - ブラウザでインストーラをダウンロードするには、次のを使用しますURL。 <https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip>

2. (オプション) ダウンロードした zip ファイルの整合性を確認する

上記のステップ .zip で AWS CLI インストーラパッケージを手動でダウンロードすることを選択した場合は、以下のステップを使用して、GnuPG ツールを使用して署名を検証できます。

AWS CLI インストーラパッケージ .zip ファイルは、PGP 署名を使用して暗号的に署名されます。ファイルの破損や変更がある場合、この検証は失敗します。その場合、インストールを続行しないでください。

- a. ご利用のパッケージマネージャーを使用し、`gpg` コマンドをダウンロードしてインストールします。GnuPG の詳細については、[GnuPG のウェブサイト](#)を参照してください。
- b. パブリックキーファイルを作成するには、テキストファイルを作成し、次のテキストに貼り付けます。

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgus0Jl7ENSyumXh85z0TRV0xJorM2B/JL0kH0yigQ1uUG
ZMLhENAG0bYatdrKP+3H91lvK050pXwn0/R7fB/FSTouki4ciIx50uLlnJZIxSzx
PqG10mkxImLNBGwoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+Lb1HAgEt5G
TfnxEKJ8soPLYWmwDH6HWcnjZ/aIQRBTIQ05uVeEoYxSh6w0ai7ss/KveoSNBbYz
gbdzoqI2Y8cgh2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU00UCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcqJ0z3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3Tqgvdx4bdkhf5cH+7NtW0
lrFj6UwAsGukBTA0xC0l/dnSmZhJ7Z1KmEwilro/g0rjt0xqRQut1IqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZwC4XKCa3mprEGC3IbJTBFqg1XmZ719yWG
EEUJY01b2XrSuPwml39beWdKM8kzr10jnl0m6+lpTRCBfo0wa9F8YZRhHPAkWkKX
XDe0GpWRj4oh0x0d2GWkyV5xyN14p2tQ0Cd00Dmz80yUTgRpPVQUt0EhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbG1AYW1hem9uLmNvbT6JAlQEEwEIAD4CGwMF
CwkIBwIGFQoJCAcCBYCAwEChgECF4AWIQT7Xbd/1cEYUaurraimMQrMRnJHXAUC
ZqFYbwUJCv/c0gAKCRCmMQrMRnJHXKYuEAC+wtZ611qQt0l0t5spM9SWZuszbcyA
0xBAJq2pncnp6wdC0kuAPu4/R3UCIoD2C49MkLj9Y0Yvue8CCF60IJ8L+fKBv2DI
yWZGmHL0p9wa/X8NCKQrKxK1gq5PuCzi3f3SqwfbZuZGeK/ubnmtttWXPuUu/Iz
VR0u/0sAy3j4uTGKh2cX7XnZbSqqJhUk9H324mIjiSwzvw1Ker6xtH/LwdBeJCck
bVBdh3LZis4zuD4IzeB01vRvjot30q4xadUv5RSPATg7T1kivrtLCnwwqc6L4LnF
00kNysk94L3LQSHyQW2kQS1cVwr+yGUSiSp+VvMbAobAapmMJWP6e/dKYAUGIX6+
2waLdbBs2U7MXznx/2ayCLPH7qCY9cenbdj5JhG9ibVvFWqqhSo22B/URQE/CMrG
+3xXwtHEBoMyWEATr1tWwn2yyQGbkUGANneSDFiTFeoQvKNyyCFTF01F2XKCCuDs
19nj34PE2TJi1TG2QR1Mr4D0NgwLLAMg2Los1CK6nXWnImYHKuaKS9LVaCoC8vu7
IRBik1NX6SjrQnftk0M9dY+s0ZbAN1gbdjZ8H3q1bl/4TxMdr87m8LP4FZIIo261
Eycv34pVkcPzPiP+dgamEiQJ7IL4Zario9mv6HbDGV6mLY45+16/0EzCwkI5IyIf
BfWC9s/USgxchg==
=ptgS
-----END PGP PUBLIC KEY BLOCK-----
```

参考までに、パブリックキーの詳細を以下に示します。

```
Key ID:          A6310ACC4672475C
Type:           RSA
Size:           4096/4096
```

```
Created:          2019-09-18
Expires:         2025-07-24
User ID:         AWS CLI Team <aws-cli@amazon.com>
Key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

- c. 次のコマンドを使用して AWS CLI パブリックキーをインポートし、置き換えます。 *public-key-file-name* 作成したパブリックキーのファイル名。

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
imported
gpg: Total number processed: 1
gpg:             imported: 1
```

- d. ダウンロードしたパッケージ AWS CLI の署名ファイルをダウンロードします。これは、対応する .zip ファイルと同じパスと名前を持っていますが、拡張子は .sig です。次の例では、現在のディレクトリに awscliv2.sig という名前のファイルとして保存します。

の最新バージョンでは AWS CLI、次のコマンドブロックを使用します。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip.sig
```

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 *2.0.30* は次のコマンド awscli-exe-linux-aarch64-2.0.30.zip.sig になります。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip.sig
```

バージョンのリストについては、[AWS CLI のバージョン 2 の変更ログ](#)を参照してくださいGitHub。

- e. 署名を検証し、.sig ファイル名と .zip ファイル名の両方をパラメータとして gpg コマンドに渡します。

```
$ gpg --verify awscliv2.sig awscliv2.zip
```


出力は以下の例のようになります。

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC
4672 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the
owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

⚠ Important

出力内の警告は予想されており、問題を示すものではありません。これは、パーソナルPGPキー (持っている場合) とキーの間に信頼チェーンがないために AWS CLI PGP発生します。詳細については、「[信用の輪 \(Web of Trust\)](#)」を参照してください。

3. インストーラを解凍します。Linux ディストリビューションに組み込み unzip コマンドがない場合は、同等のコマンドを使用して解凍します。次のコマンド例では、パッケージを解凍し、現在のディレクトリの下に aws という名前のディレクトリを作成します。

```
$ unzip awscliv2.zip
```

i Note

以前のバージョンからアップデートする場合、unzip コマンドでは、既存のファイルを上書きするように求められます。スクリプトのオートメーションなどでこのプロンプトをスキップする場合は、unzip の `-u` 更新フラグを使用します。このフラグは、既存のファイルを自動的に更新し、必要に応じて新しいファイルを作成します。

```
$ unzip -u awscliv2.zip
```

4. インストールプログラムを実行します。インストールコマンドは、新しく解凍された `install` ディレクトリにある `aws` という名前のファイルを使用します。デフォルトでは、すべてのファイルが `/usr/local/aws-cli` にインストールされ、シンボリックリンクが /

`usr/local/bin` に作成されます。コマンドには、これらのディレクトリへの書き込み権限を付与するための `sudo` が含まれます。

```
$ sudo ./aws/install
```

すでに書き込み権限を持つディレクトリを指定した場合は、`sudo` なしでインストールできます。次の `install` コマンド用の手順を使用して、インストール場所を指定します。

- `-i` および `-b` パラメータに指定するパス内のボリューム名やディレクトリ名に、スペース文字などの空白文字が含まれないことを確認します。スペースが含まれていると、インストールは失敗します。
- `--install-dir` または `-i` — このオプションは、すべてのファイルのコピー先ディレクトリを指定します。

デフォルト値は `/usr/local/aws-cli` です。

- `--bin-dir` または `-b` — このオプションは、インストールディレクトリのメイン `aws` プログラムが、指定されたパスのファイル `aws` にシンボリックにリンクされることを指定します。指定したディレクトリへの書き込み権限が必要です。パスにすでに存在するディレクトリへのシンボリックリンクを作成すると、インストールディレクトリをユーザーの `$PATH` 変数に追加する必要がなくなります。

デフォルト値は `/usr/local/bin` です。

```
$ ./aws/install -i /usr/local/aws-cli -b /usr/local/bin
```

Note

の現在のインストールを更新するには AWS CLI、既存の `symlink` とインストーラ情報を追加して、`--update` パラメータを使用して `install` コマンドを構築します。

```
$ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/  
aws-cli --update
```

既存のシンボリックリンクとインストールディレクトリを検索するには、次の手順に従います。

1. `which` コマンドを使用して、シンボリックリンクを見つけます。これにより、`--bin-dir` パラメータで使用するパスが指定されます。

```
$ which aws
/usr/local/bin/aws
```

2. `ls` コマンドを使用して、シンボリックリンクが指すディレクトリを見つけます。これにより、`--install-dir` パラメータで使用するパスが指定されます。

```
$ ls -l /usr/local/bin/aws
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /
usr/local/aws-cli/v2/current/bin/aws
```

5. 以下のコマンドを使用して、インストールを確認します。

```
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/2.4.5
```

`aws` コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Snap package

AWS CLI では、の公式 AWS サポートバージョンを提供しています `snap`。システムに常に最新バージョンの AWS CLI をインストールする場合は、スナップパッケージが自動更新時にこれを提供します。のマイナーバージョンを選択するための組み込みサポートがないため AWS CLI、チームがバージョンを固定する必要がある場合は最適なインストール方法ではありません。の特定のマイナーバージョンをインストールする場合は AWS CLI、コマンドラインインストーラを使用することをお勧めします。

1. Linux プラットフォームがまだ `snap` インストールされていない場合は、プラットフォーム `snap` に をインストールします。
 - a. のインストールの詳細については `snap`、「[Snap ドキュメント](#)」の「[デーモンのインストール](#)」を参照してください。

- b. PATH 変数が正しく更新されるように、システムを再起動する必要がある場合があります。インストールに問題がある場合は、「Snap ドキュメント」の「[一般的な問題を修正する](#)」の手順に従ってください。
- c. が正しくインストールsnapされていることを確認するには、次のコマンドを実行します。

```
$ snap version
```

2. に対して次のsnap installコマンドを実行します AWS CLI。

```
$ snap install aws-cli --classic
```

アクセス許可によっては、コマンドsudoに を追加する必要がある場合があります。

```
$ sudo snap install aws-cli --classic
```

Note

のスナップリポジトリを表示するには AWS CLI、追加のsnap手順を含め、Canonical Snapcraft ウェブサイトの[aws-cli](#)「」ページを参照してください。

3. が正しく AWS CLI インストールされていることを確認します。

```
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/2.4.5
```

エラーが表示される場合は、「[のエラーのトラブルシューティング AWS CLI](#)」を参照してください。

macOS

インストールと更新の要件

- macOS AWS CLI バージョン 10.15 以降のをサポートしています。詳細については、AWS デベロッパーツールブログの[AWS CLI 「v2 の macOS サポートポリシーの更新](#)」を参照してください。

- AWS はサードパーティリポジトリを保持していないため、の最新バージョンが含まれていることを保証することはできません AWS CLI。

AWS CLIをインストールまたは更新する

最新バージョンに更新する場合は、現在のバージョンで使用したのと同じインストール方法を使用します。macOS AWS CLI に をインストールするには、次の方法があります。

GUI installer

次の手順では、標準の macOS ユーザーインターフェイスとブラウザ AWS CLI を使用して の最新バージョンをインストールする方法を示します。

1. ブラウザで、macOS pkg ファイルをダウンロードします: <https://awscli.amazonaws.com/AWSCLIV2.pkg>
2. ダウンロードしたファイルを実行し、その後は画面の指示に従います。のインストールは、AWS CLI 次の方法で選択できます。
 - コンピュータ上のすべてのユーザーが対象 (sudo が必要)
 - 任意のフォルダにインストールするか、推奨されるデフォルトのフォルダ `/usr/local/aws-cli` を選択できます。
 - インストーラによって、選択したインストールフォルダ内のメインプログラムにリンクするシンボリックリンクが `/usr/local/bin/aws` に自動的に作成されます。
 - 現在のユーザーのみが対象 (sudo は不要)
 - 書き込みアクセス許可がある任意のフォルダにインストールできます。
 - 標準のユーザー権限により、インストーラの完了後に、コマンドプロンプトで次のコマンドを使用して、aws および aws_completer プログラムを指すシンボリックリンク ファイルを \$PATH で手動で作成する必要があります。\$PATH に書き込み可能なフォルダが含まれている場合、そのフォルダをターゲットのパスとして指定すると、sudo なしで以下のコマンドを実行できます。\$PATH に書き込み可能なフォルダが含まれていない場合は、コマンドで sudo を使用して、指定したターゲットフォルダに対する書き込みアクセス許可を取得する必要があります。シンボリックリンクのデフォルトの場所は `/usr/local/bin/` です。

```
$ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
$ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/
aws_completer
```

Note

インストーラの任意の場所で `Cmd + L` キーを押すと、インストールのデバッグログを表示できます。これにより、ログをフィルター処理して保存できるログペインが開きます。また、ログファイルは `/var/log/install.log` に自動的に保存されます。

3. シェルが `aws` で `$PATH` コマンドを検索して実行できることを確認するには、次のコマンドを使用します。

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Darwin/23.3.0 botocore/2.4.5
```

`aws` コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Command line installer - All users

`sudo` アクセス許可がある場合は、コンピュータ上のすべてのユーザーを対象に AWS CLI をインストールできます。1 つの簡単な手順でグループのコピーと貼り付けができます。以下のステップの各行の説明を参照してください。

```
$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
$ sudo installer -pkg AWSCLIV2.pkg -target /
```

ガイド付きインストール手順

1. `curl` コマンドを使用して、ファイルをダウンロードします。`-o` オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。この例では、ファイルは現在のフォルダ内の `AWSCLIV2.pkg` に書き込まれます。

```
$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
```

2. ダウンロードした `.pkg` ファイルをソースとして指定して、標準の macOS installer プログラムを実行します。`-pkg` パラメータを使用して、インストールするパッケージの

名前、およびパッケージをインストールするドライブの `-target` / パラメータを指定します。ファイルは `/usr/local/aws-cli` にインストールされ、シンボリックリンクが `/usr/local/bin` に自動的に作成されます。これらのフォルダに対する書き込みアクセス許可を付与するには、コマンドに `sudo` を含める必要があります。

```
$ sudo installer -pkg ./AWSCLIV2.pkg -target /
```

インストールが完了すると、デバッグログが `/var/log/install.log` に書き込まれます。

3. シェルが `aws` で `$PATH` コマンドを検索して実行できることを確認するには、次のコマンドを使用します。

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Darwin/23.3.0 botocore/2.4.5
```

`aws` コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Command line - Current user

1. AWS CLI がインストールされているフォルダを指定するには、任意のファイル名の XML ファイルを作成する必要があります。このファイルは、次の例のような XML 形式のファイルです。パスを置き換える必要がある以外は、図のようにすべての値のままにします `/Users/myusername` 9 行目に、AWS CLI をインストールするフォルダへのパスを入力します。フォルダはすでに存在している必要があります。存在しない場合、コマンドは失敗します。次の XML 例では、`choices.xml` は、フォルダ AWS CLI に をインストールするインストーラを指定します。ここで `/Users/myusername`、という名前のフォルダが作成されます `aws-cli`。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <array>
    <dict>
      <key>choiceAttribute</key>
```

```
<string>customLocation</string>
<key>attributeSetting</key>
<string>/Users/myusername</string>
<key>choiceIdentifier</key>
<string>default</string>
</dict>
</array>
</plist>
```

2. `curl` コマンドを使用して `pkg` インストーラをダウンロードします。`-o` オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。この例では、ファイルは現在のフォルダ内の `AWSCLIV2.pkg` に書き込まれます。

```
$ curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
```

3. 以下のオプションを使用して、標準の macOS installer プログラムを実行できます。
 - `-pkg` パラメータで、インストールするパッケージの名前を指定します。
 - `-target` パラメータを `CurrentUserHomeDirectory` に設定して、現在のユーザーのみへのインストールを指定します。
 - `-applyChoiceChangesXML` パラメータで作成した XML ファイルのパス (現在のフォルダに関連) と名前を指定します。

次の例では、フォルダ `AWS CLI` に をインストールします `/Users/myusername/aws-cli`。

```
$ installer -pkg AWSCLIV2.pkg \
            -target CurrentUserHomeDirectory \
            -applyChoiceChangesXML choices.xml
```

4. 通常、標準のユーザーアクセス許可では `$PATH` 内のフォルダへの書き込みが許可されないため、このモードのインストーラは `aws` および `aws_completer` プログラムにシンボリックリンクを追加しようとしません。を正しく実行する AWS CLI には、インストーラの完了後、シンボリックリンクを手動で作成する必要があります。`$PATH` に書き込み可能なフォルダが含まれていて、そのフォルダをターゲットのパスとして指定すると、`sudo` なしで以下のコマンドを実行できます。`$PATH` に書き込み可能なフォルダが含まれていない場合は、`sudo` を使用して、指定したターゲットフォルダに対する書き込みアクセス許可を取得する必要があります。シンボリックリンクのデフォルトの場所は `/usr/local/bin/` です。 `folder/installed` を、AWS CLI のインストールへのパスに置き換えます。


```
$ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
$ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/
aws_completer
```

インストールが完了すると、デバッグログが `/var/log/install.log` に書き込まれます。

5. シェルが `aws` で `$PATH` コマンドを検索して実行できることを確認するには、次のコマンドを使用します。

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Darwin/23.3.0 botocore/2.4.5
```

`aws` コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Windows

インストールと更新の要件

- Microsoft がサポートするバージョンの 64 ビット Windows AWS CLI では がサポートされていません。
- ソフトウェアをインストールするための管理者権限

AWS CLIをインストールまたは更新する

Windows AWS CLI での の現在のインストールを更新するには、以前のバージョンを上書きするために更新するたびに新しいインストーラをダウンロードします。AWS CLI は定期的に更新されます。最新バージョンがいつリリースされたかを確認するには、「」の[AWS CLI 「バージョン 2 の変更ログ](#)」を参照してくださいGitHub。

1. Windows (64 ビット): 用のインストーラをダウンロードして実行します AWS CLI MSI。

<https://awscli.amazonaws.com/AWSCLIV2.msi>

または、`msiexec` コマンドを実行してMSIインストーラを実行できます。

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```

msiexec で使用できるさまざまなパラメータについては、Microsoft Docs ウェブサイトの [msiexec](#) を参照してください。例えば、/qn フラグを使用してサイレントインストールを行うことができます。

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi /qn
```

2. インストールを確認するには、[スタート] メニューを開き、cmd を検索してコマンドプロンプトウィンドウを開いて、コマンドプロンプトで `aws --version` コマンドを使用します。

```
C:\> aws --version
aws-cli/2.17.20 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

Windows がプログラムを見つけることができない場合、パスを更新するためにコマンドプロンプトウィンドウを閉じて再度開くか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

AWS CLI インストールとアンインストールのエラーのトラブルシューティング

のインストールまたはアンインストール後に問題が発生した場合は AWS CLI、トラブルシューティングの手順 [エラーのトラブルシューティング](#) については、「」を参照してください。最も関連性の高いトラブルシューティングステップについては、「[the section called “コマンドが見つからないエラー”](#)」、「[the section called “aws --version コマンドが、インストールしたのとは異なるバージョンを返す”](#)」、および「[the section called “aws --version 「」 コマンドは、AWS CLI”](#)」を参照してください。

次のステップ

が正常にインストールされたら AWS CLI、ダウンロードしたインストーラファイルを安全に削除できます。の手順を完了 [the section called “前提条件”](#) し、 をインストールしたら AWS CLI、 を実行する必要があります [the section called “セットアップ”](#)。

AWS CLI バージョン 2 の過去のリリースのインストール

このトピックでは、サポートされているオペレーティングシステムに AWS Command Line Interface バージョン 2 (AWS CLI) の過去のリリースをインストールする方法について説明します。AWS CLI バージョン 2 リリースの詳細については、[AWS CLI のバージョン 2 の変更ログ](#)を参照してください GitHub。

AWS CLI バージョン 2 のインストール手順：

Linux

インストール要件

- インストールする AWS CLI バージョン 2 のリリースがわかります。バージョンのリストについては、[AWS CLI のバージョン 2 の変更ログ](#)を参照してください GitHub。
- ダウンロードしたパッケージを抽出または解凍できる必要があります。オペレーティングシステムに組み込み unzip コマンドがない場合は、同等のコマンドを使用します。
- AWS CLI バージョン 2 では glibc、groff、および less を使用します。これらは、Linux のほとんどの主要なディストリビューションにデフォルトで含まれています。
- CentOS、Fedora、Ubuntu、Amazon Linux 1、Amazon Linux 2、Linux の最近のディストリビューションの 64 ビットバージョンで AWS CLI バージョン 2 がサポートされています ARM。
- AWS はサードパーティーリポジトリを保持していないため、の最新バージョンが含まれていることを保証することはできません AWS CLI。

インストール手順

コマンドラインから以下の手順に従って、Linux AWS CLI に をインストールします。

64 ビット Linux と Linux のどちらを使用するかに基づいて、コピーと貼り付けが簡単な 1 つのグループにステップが用意されています ARM。以下のステップの各行の説明を参照してください。

Linux x86 (64-bit)

Note

(オプション) 次のコマンドブロックは、ダウンロードの整合性を最初に検証 AWS CLI することなく、 をダウンロードしてインストールします。ダウンロードの整合性を検証するには、以下の詳しい手順に従ってください。

バージョンのリストについては、「」の[AWS CLI 「バージョン 2 の変更ログ」](#)を参照してくださいGitHub。

をインストールするには AWS CLI、次のコマンドを実行します。

バージョンを指定する場合は、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンドawscli-exe-linux-x86_64-2.0.30.zipになります。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o
"awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

AWS CLIの現在のインストールを更新するには、既存のシンボリックリンクとインストーラの情報を追加して `--bin-dir`、`--install-dir`、および `--update` パラメータで `install` コマンドを作成します。次のコマンドブロックは、のシンボリックリンクの例を使用しています。`/usr/local/bin` および のインストーラの場所の例 `/usr/local/aws-cli`。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o
"awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --
update
```

Linux ARM

Note

(オプション) 次のコマンドブロックは、ダウンロードの整合性を最初に検証 AWS CLI することなく、をダウンロードしてインストールします。ダウンロードの整合性を検証するには、以下の詳しい手順に従ってください。

バージョンのリストについては、「」の[AWS CLI 「バージョン 2 の変更ログ」](#)を参照してくださいGitHub。

をインストールするには AWS CLI、次のコマンドを実行します。

バージョンを指定する場合は、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `awscli-exe-linux-aarch64-2.0.30.zip` になります。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip" -o
"awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

AWS CLIの現在のインストールを更新するには、既存のシンボリックリンクとインストーラの情報を追加して `--bin-dir`、`--install-dir`、および `--update` パラメータで `install` コマンドを作成します。次のコマンドブロックは、のシンボリックリンクの例を使用しています。 `/usr/local/bin` および のインストーラの場所の例 `/usr/local/aws-cli`。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip" -o
"awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-cli --
update
```

1. 次のいずれかの方法でインストールファイルをダウンロードします。

Linux x86 (64-bit)

- `curl` コマンドの使用 `--o` オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。以下のコマンドの例のオプションを使用すると、ダウンロードしたファイルが現在のディレクトリにローカル名 `awscliv2.zip` で書き込まれます。

バージョンを指定する場合は、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `awscli-exe-linux-x86_64-2.0.30.zip` になります。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip" -o
"awscliv2.zip"
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

- からのダウンロード URL-

ブラウザで、ファイル名にハイフンとバージョン番号 AWS CLI を追加して、特定のバージョンのをダウンロードします。

```
https://awscli.amazonaws.com/awscli-exe-linux-x86_64-version.number.zip
```

この例では、バージョンのファイル名 **2.0.30** は awscli-exe-linux-x86_64-2.0.30.zip になり、次のリンクになります。 https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip

Linux ARM

- **curl** コマンドの使用 `--o` オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。以下のコマンドの例のオプションを使用すると、ダウンロードしたファイルが現在のディレクトリにローカル名 `awscliv2.zip` で書き込まれます。

バージョンを指定する場合は、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `awscli-exe-linux-aarch64-2.0.30.zip` になります。

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip" -o  
"awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install
```

- からのダウンロード URL-

ブラウザで、ファイル名にハイフンとバージョン番号 AWS CLI を追加して、特定のバージョンのをダウンロードします。

```
https://awscli.amazonaws.com/awscli-exe-linux-aarch64-version.number.zip
```

この例では、バージョンのファイル名 **2.0.30** は以下のリンク `awscli-exe-linux-aarch64-2.0.30.zip` になります。 <https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip>

2. (オプション) ダウンロードした zip ファイルの整合性を確認する

上記のステップ `.zip` で AWS CLI インストーラパッケージを手動でダウンロードすることを選択した場合は、以下のステップを使用して、GnuPG ツールを使用して署名を検証できます。

AWS CLI インストーラパッケージ.zipファイルは、PGP署名を使用して暗号的に署名されます。ファイルの破損や変更がある場合、この検証は失敗します。その場合、インストールを続行しないでください。

- ご利用のパッケージマネージャーを使用し、`gpg` コマンドをダウンロードしてインストールします。GnuPG の詳細については、[GnuPG のウェブサイト](#)を参照してください。
- パブリックキーファイルを作成するには、テキストファイルを作成し、次のテキストに貼り付けます。

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgus0Jl7ENSyUmXh85z0TRV0xJorM2B/JL0kH0yigQ1uUG
ZMLhEnAG0bYatdrKP+3H911vK050pXwn0/R7fB/FSTouki4ciIx50uLlnJZIxSzx
PqG10mkxImLNBGwoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+Lb1HAGEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6w0ai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2bnfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU00UCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcgJ0z3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtW0
lrFj6UwAsGukBTA0xC01/dnSmZhJ7Z1KmEWilro/g0rjt0xqRQut1IqG22TaqoPG
fYVN+en3ZwbT97kcgZDwqbuykNt64oZwc4XKCa3mprEGC3IbJTBFqg1XmZ719ywG
EEUJY01b2XrSuPwm139beWdKM8kzr10jn10m6+lpTRCBfo0wa9F8YZRhHPAkWkKX
XDe0GpWRj4oh0x0d2GWkyV5xyN14p2tQ0Cd00Dmz80yUTgRpPVQUt0EHXQARAQAB
tCFBV1MgQ0xJIFR1YW0gPGF3cy1jbG1AYW1hem9uLmNvbT6JAlQEEwEiAD4CGwMF
CwkIBWIGFQoJCAwEChgECF4AWIQT7Xbd/1cEYuAURraimMQrMRnJHXAUC
ZqFYbwUJCv/c0gAKCRCmMQrMRnJHXYuEAC+wtZ611qQt010t5spM9SWZuszbcyA
0xBAJq2pncnp6wdC0kuAPu4/R3UCIoD2C49MkLj9Y0Yvue8CCF60IJ8L+fKBv2DI
yWZGmHL0p9wa/X8NCKQrKxK1gq5PuCzi3f3SqwfbZuZGeK/ubnmtttWXPuU/Iz
VR0u/0sAy3j4uTGKh2cX7XnZbSqqJhUk9H324mIjSwzvw1Ker6xtH/LwdBeJCck
bVBdh3LZis4zuD4IZeB01vRvjot30q4xadUv5RSPATg7T1kivrtLCnwvqc6L4LnF
00kNysk94L3LQSHyQW2kQS1cVwr+yGUSiSp+VvMbAobAapmMJP6e/dKyAUGIX6+
2waLdbBs2U7MXznx/2ayCLPH7qCY9cenbdj5JhG9ibVvFWqqhSo22B/URQE/CMrG
+3xXwtHEBoMyWEATr1tWwn2yyQGbKUGANneSDFiTFeoQvKNyyCFTF01F2XKCcuDs
19nj34PE2TJi1TG2QR1Mr4D0NgwLLAMg2Los1CK6nXWnImYHKuaKS9LVaCoC8vu7
IRBik1NX6SjrQnftk0M9dY+s0ZbAN1gbdjZ8H3qlb1/4TxMdr87m8LP4FZIIo261
Eycv34pVkcPZiP+dgamEiQJ7IL4Zario9mv6HbDGV6mLY45+16/0EzCwkI5IyIf
BfWC9s/USgxchg==
=ptgS
-----END PGP PUBLIC KEY BLOCK-----
```

参考までに、パブリックキーの詳細を以下に示します。

```
Key ID:          A6310ACC4672
Type:           RSA
Size:          4096/4096
Created:       2019-09-18
Expires:       2025-07-24
User ID:       AWS CLI Team <aws-cli@amazon.com>
Key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

- c. 次のコマンドを使用して AWS CLI パブリックキーをインポートし、置き換えます。 *public-key-file-name* 作成したパブリックキーのファイル名。

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
imported
gpg: Total number processed: 1
gpg:             imported: 1
```

- d. ダウンロードしたパッケージ AWS CLI の署名ファイルをダウンロードします。これは、対応する .zip ファイルと同じパスと名前を持っていますが、拡張子は .sig です。次の例では、現在のディレクトリに awscliv2.sig という名前のファイルとして保存します。

Linux x86 (64-bit)

の最新バージョンでは AWS CLI、次のコマンドブロックを使用します。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip.sig
```

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 *2.0.30* は次のコマンド awscli-exe-linux-x86_64-2.0.30.zip.sig になります。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip.sig
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

Linux ARM

の最新バージョンでは AWS CLI、次のコマンドブロックを使用します。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip.sig
```

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `awscli-exe-linux-aarch64-2.0.30.zip.sig` になります。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip.sig
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

- e. 署名を検証し、`.sig` ファイル名と `.zip` ファイル名の両方をパラメータとして `gpg` コマンドに渡します。

```
$ gpg --verify awscliv2.sig awscliv2.zip
```

出力は以下の例のようになります。

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672
 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

Important

出力内の警告は予想されており、問題を示すものではありません。これは、パーソナルPGPキー (持っている場合) と AWS CLI PGPキーの間に信頼チェーンがないために発生します。詳細については、「[信用の輪 \(Web of Trust\)](#)」を参照してください。

3. インストーラを解凍します。Linux ディストリビューションに組み込み unzip コマンドがない場合は、同等のコマンドを使用して解凍します。次のコマンド例では、パッケージを解凍し、現在のディレクトリの下に aws という名前のディレクトリを作成します。

```
$ unzip awscliv2.zip
```

4. インストールプログラムを実行します。インストールコマンドは、新しく解凍された install ディレクトリにある aws という名前のファイルを使用します。デフォルトでは、すべてのファイルが /usr/local/aws-cli にインストールされ、シンボリックリンクが /usr/local/bin に作成されます。コマンドには、これらのディレクトリへの書き込み権限を付与するための sudo が含まれます。

```
$ sudo ./aws/install
```

すでに書き込み権限を持つディレクトリを指定した場合は、sudo なしでインストールできます。次の install コマンド用の手順を使用して、インストール場所を指定します。

- -i および -b パラメータに指定するパス内のボリューム名やディレクトリ名に、スペース文字などの空白文字が含まれないことを確認します。スペースが含まれていると、インストールは失敗します。
- --install-dir または -i — このオプションは、すべてのファイルのコピー先ディレクトリを指定します。

デフォルト値は /usr/local/aws-cli です。

- --bin-dir または -b — このオプションは、インストールディレクトリのメイン aws プログラムが、指定されたパスのファイル aws にシンボリックにリンクされることを指定します。指定したディレクトリへの書き込み権限が必要です。パスにすでに存在するディレクトリへのシンボリックリンクを作成すると、インストールディレクトリをユーザーの \$PATH 変数に追加する必要がなくなります。

デフォルト値は /usr/local/bin です。

```
$ ./aws/install -i /usr/local/aws-cli -b /usr/local/bin
```

Note

AWS CLI バージョン 2 の現在のインストールを新しいバージョンに更新するには、既存のシンボリックリンクとインストーラ情報を追加して、`--update`パラメータを使用して `install` コマンドを構築します。

```
$ sudo ./aws/install --bin-dir /usr/local/bin --install-dir /usr/local/aws-  
cli --update
```

既存のシンボリックリンクとインストールディレクトリを検索するには、次の手順に従います。

1. `which` コマンドを使用して、シンボリックリンクを見つけます。これにより、`--bin-dir` パラメータで使用するパスが指定されます。

```
$ which aws  
/usr/local/bin/aws
```

2. `ls` コマンドを使用して、シンボリックリンクが指すディレクトリを見つけます。これにより、`--install-dir` パラメータで使用するパスが指定されます。

```
$ ls -l /usr/local/bin/aws  
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/  
local/aws-cli/v2/current/bin/aws
```

5. 以下のコマンドを使用して、インストールを確認します。

```
$ aws --version  
aws-cli/2.17.20 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/2.4.5
```

`aws` コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

(オプション) ダウンロードした zip ファイルの整合性を確認する

上記のステップ .zip で AWS CLI バージョン 2 インストーラパッケージを手動でダウンロードすることを選択した場合は、ツールを使用して署名を検証するために以下のステップを使用できませんGnuPG。

AWS CLI バージョン 2 インストーラパッケージ.zipファイルは、PGP署名を使用して暗号的に署名されます。ファイルの破損や変更がある場合、この検証は失敗します。その場合、インストールを続行しないでください。

1. ご利用のパッケージマネージャーを使用し、gpg コマンドをダウンロードしてインストールします。GnuPG の詳細については、[GnuPG のウェブサイト](#)を参照してください。
2. パブリックキーファイルを作成するには、テキストファイルを作成し、次のテキストに貼り付けます。

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBF2Cr7UBEADJZHcgus0Jl7ENSyumXh85z0TRV0xJorM2B/JL0kH0yigQ1uUG
ZMLhEnAG0bYatdrKP+3H911vK050pXwn0/R7fB/FSTouki4ciIx50uLlnJZIxSzx
PqG10mkxImLNBGwoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+Lb1HAGet5G
TfnxEKJ8soPLyWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6w0ai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2bnfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU00UCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcqJ0z3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3TqgvdX4bdkhf5cH+7NtW0
lrFj6UwAsGukBTA0xC01/dnSmZhJ7Z1KmEWilro/g0rjt0xqRQut1IqG22TaqoPG
fYVN+en3ZwbT97kcgZDwqbuykNt64oZwC4XKCa3mprEGC3IbJTBFqglXmZ719ywG
EEUJY01b2XrSuPwm139beWdKM8kzr10jn10m6+lpTRCBfo0wa9F8YZRhHPAkWkKX
XDe0GpWrrj4oh0x0d2GWkyV5xyN14p2tQ0Cd00Dmz80yUTgRpPVQUt0EhXQARAQAB
tCFBV1MgQ0xJIFRlYW0gPGF3cy1jbG1AYW1hem9uLmNvbT6JAlQEEwEIAID4CGwMF
CwkIBWIGFQoJCAwEChgECF4AWIQT7Xbd/1cEYUauraimMQrMRnJHXAUC
ZqFYbwUJCv/c0gAKCRCmMQrMRnJHXYuEAC+wtZ611qQt010t5spM9SWZuszbcyA
0xBAJq2pncnp6wdC0kuAPu4/R3UCIoD2C49MkLj9Y0Yvue8CCF60IJ8L+fKBv2DI
yWZGmHL0p9wa/X8NCKQrKxK1gq5PuCzi3f3SqwfbZuZGeK/ubnmtttWxpUtuU/Iz
VR0u/0sAy3j4uTGKh2cX7XnZbSqqJhUk9H324mIjiSwzvw1Ker6xtH/LwdBeJCck
bVBdh3LZis4zuD4IZEB01vRvjot30q4xadUv5RSPATg7T1kivrtLCnwvqc6L4LnF
00kNysk94L3LQSHyQW2kQS1cVwr+yGUSiSp+VvMbAobAapmMJWP6e/dKyAUGIX6+
2waLdbBs2U7MXznx/2ayCLPH7qCY9cenbdj5JhG9ibVvFWqqhSo22B/URQE/CMrG
+3xXwtHEBoMyWEATr1tWwn2yyQgbkUGANneSDFiTFeoQvKNyyCFTF01F2XKCcuDs
19nj34PE2TJi1TG2QR1Mr4D0NgwLLAMg2Los1CK6nXWnImYHKuaKS9LVaCoC8vu7
IRBik1NX6SjrQnftk0M9dY+s0ZbAN1gbdjZ8H3qlbl/4TxMdr87m8LP4FZIIo261
Eycv34pVkcPZiP+dgamEiQJ7IL4Zario9mv6HbDGV6mLY45+16/0EzCwkI5IyIf
BfWC9s/USgxchg==
=ptgS
-----END PGP PUBLIC KEY BLOCK-----
```

参考までに、パブリックキーの詳細を以下に示します。

```
Key ID:           A6310ACC4672
Type:             RSA
Size:             4096/4096
Created:          2019-09-18
Expires:          2025-07-24
User ID:          AWS CLI Team <aws-cli@amazon.com>
Key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

3. 次のコマンドを使用して AWS CLI パブリックキーをインポートし、置き換えます。*public-key-file-name* 作成したパブリックキーのファイル名。

```
$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>" imported
gpg: Total number processed: 1
gpg:             imported: 1
```

4. ダウンロードしたパッケージ AWS CLI の署名ファイルをダウンロードします。これは、対応する .zip ファイルと同じパスと名前を持っていますが、拡張子は .sig です。次の例では、現在のディレクトリに awscliv2.sig という名前のファイルとして保存します。

Linux x86 (64-bit)

AWS CLI の最新バージョンの場合は、次のコマンドブロックを使用します。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip.sig
```

AWS CLI の特定バージョンの場合は、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド awscli-exe-linux-x86_64-2.0.30.zip.sig になります。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86_64-2.0.30.zip.sig
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

Linux ARM

の最新バージョンでは AWS CLI、次のコマンドブロックを使用します。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip.sig
```

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `awscli-exe-linux-aarch64-2.0.30.zip.sig` になります。

```
$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-aarch64-2.0.30.zip.sig
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

- 署名を検証し、`.sig` ファイル名と `.zip` ファイル名の両方をパラメータとして `gpg` コマンドに渡します。

```
$ gpg --verify awscliv2.sig awscliv2.zip
```

出力は以下の例のようになります。

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:          using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

Important

出力内の警告は予想されており、問題を示すものではありません。これは、パーソナル PGP キー (持っている場合) とキーの間に信頼チェーンがないために AWS CLI PGP 発生します。詳細については、「[信用の輪 \(Web of Trust\)](#)」を参照してください。

macOS

インストール要件

- インストールする AWS CLI バージョン 2 のリリースがわかります。バージョンのリストについては、「」の[AWS CLI 「バージョン 2 の変更ログ」](#)を参照してくださいGitHub。
- Apple がサポートする AWS CLI 64 ビット macOS のバージョンでは、バージョン 2 がサポートされています。
- AWS はサードパーティーリポジトリを保持していないため、の最新バージョンが含まれていることを保証することはできません AWS CLI。

インストール手順

AWS CLI バージョン 2 は、次の方法で macOS にインストールできます。

GUI installer

次の手順では、標準の macOS ユーザーインターフェイスとブラウザを使用して、バージョン AWS CLI 2 の最新バージョンをインストールまたは更新する方法を示します。最新バージョンに更新する場合は、現在のバージョンに使用したのと同じインストール方法を使用します。

1. ブラウザで、ファイル名にハイフンとバージョン番号を追加して、AWS CLI の特定のバージョンをダウンロードします。

```
https://awscli.amazonaws.com/AWSCLIV2-version.number.pkg
```

この例では、バージョンのファイル名 **2.0.30** は次のリンク `AWSCLIV2-2.0.30.pkg` になります: <https://awscli.amazonaws.com/AWSCLIV2-2.0.30.pkg>。

2. ダウンロードしたファイルを実行し、その後は画面の指示に従います。AWS CLI バージョン 2 は、次の方法でインストールできます。
 - コンピュータ上のすべてのユーザーが対象 (sudo が必要)
 - 任意のフォルダにインストールするか、推奨されるデフォルトのフォルダ `/usr/local/aws-cli` を選択できます。
 - インストーラによって、選択したインストールフォルダ内のメインプログラムにリンクするシンボリックリンクが `/usr/local/bin/aws` に自動的に作成されます。
 - 現在のユーザーのみが対象 (sudo は不要)

- 書き込みアクセス許可がある任意のフォルダにインストールできます。
- 標準のユーザー権限により、インストーラの完了後に、コマンドプロンプトで次のコマンドを使用して、aws および aws_completer プログラムを指すシンボリックリンク ファイルを \$PATH で手動で作成する必要があります。\$PATH に書き込み可能なフォルダが含まれている場合、そのフォルダをターゲットのパスとして指定すると、sudo なしで以下のコマンドを実行できます。\$PATH に書き込み可能なフォルダが含まれていない場合は、コマンドで sudo を使用して、指定したターゲットフォルダに対する書き込みアクセス許可を取得する必要があります。シンボリックリンクのデフォルトの場所は /usr/local/bin/ です。

```
$ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws
$ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/
aws_completer
```

Note

インストーラの任意の場所で Cmd + L キーを押すと、インストールのデバッグログを表示できます。これにより、ログをフィルター処理して保存できるログペインが開きます。また、ログファイルは /var/log/install.log に自動的に保存されます。

3. シェルが aws で \$PATH コマンドを検索して実行できることを確認するには、次のコマンドを使用します。

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Darwin/23.3.0 botocore/2.4.5
```

aws コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Command line installer - All users

sudo 許可がある場合は、コンピュータ上のすべてのユーザー用に AWS CLI バージョン 2 をインストールできます。1 つの簡単な手順でグループのコピーと貼り付けができます。以下のステップの各行の説明を参照してください。

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `AWSCLI2-2.0.30.pkg` になります。

```
$ curl "https://awscli.amazonaws.com/AWSCLI2-2.0.30.pkg" -o "AWSCLI2.pkg"
$ sudo installer -pkg AWSCLI2.pkg -target /
```

1. `curl` コマンドを使用して、ファイルをダウンロードします。`-o` オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。この例では、ファイルは現在のフォルダ内の `AWSCLI2.pkg` に書き込まれます。

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `AWSCLI2-2.0.30.pkg` になります。

```
$ curl "https://awscli.amazonaws.com/AWSCLI2-2.0.30.pkg" -o "AWSCLI2.pkg"
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

2. ダウンロードした `.pkg` ファイルをソースとして指定して、標準の macOS installer プログラムを実行します。`-pkg` パラメータを使用して、インストールするパッケージの名前、およびパッケージをインストールするドライブの `-target /` パラメータを指定します。ファイルは `/usr/local/aws-cli` にインストールされ、シンボリックリンクが `/usr/local/bin` に自動的に作成されます。これらのフォルダに対する書き込みアクセス許可を付与するには、コマンドに `sudo` を含める必要があります。

```
$ sudo installer -pkg ./AWSCLI2.pkg -target /
```

インストールが完了すると、デバッグログが `/var/log/install.log` に書き込まれます。

3. シェルが `aws` で `$PATH` コマンドを検索して実行できることを確認するには、次のコマンドを使用します。

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Darwin/23.3.0 botocore/2.4.5
```

aws コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Command line - Current user

1. AWS CLI がインストールされているフォルダを指定するには、XML ファイルを作成する必要があります。このファイルは、次の例のようなXML形式のファイルです。パスを置き換える必要がある以外は、図のようにすべての値のままにします */Users/myusername* AWS CLI バージョン 2 をインストールするフォルダへのパスを含む 9 行目。フォルダはすでに存在している必要があります。存在しない場合、コマンドは失敗します。この例では、インストーラがフォルダ AWS CLI に をインストールし/*Users/myusername*、という名前のフォルダを作成することXMLを指定しますaws-cli。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <array>
    <dict>
      <key>choiceAttribute</key>
      <string>customLocation</string>
      <key>attributeSetting</key>
      <string>/Users/myusername</string>
      <key>choiceIdentifier</key>
      <string>default</string>
    </dict>
  </array>
</plist>
```

2. curl コマンドを使用して pkg インストーラをダウンロードします。-o オプションは、ダウンロードしたパッケージを書き込むファイル名を指定します。この例では、ファイルは現在のフォルダ内の AWSCLIV2.pkg に書き込まれます。

の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。この例では、バージョンのファイル名 **2.0.30** は次のコマンド `AWSCLI2-2.0.30.pkg` になります。

```
$ curl "https://awscli.amazonaws.com/AWSCLI2-2.0.30.pkg" -o "AWSCLI2.pkg"
```

バージョンのリストについては、「」の [AWS CLI 「バージョン 2 の変更ログ」](#) を参照してくださいGitHub。

- 以下のオプションを使用して、標準の macOS installer プログラムを実行できます。
 - pkg パラメータで、インストールするパッケージの名前を指定します。
 - target パラメータを `CurrentUserHomeDirectory` に設定して、現在のユーザーのみへのインストールを指定します。
 - applyChoiceChangesXML パラメータで作成したXMLファイルのパス (現在のフォルダに関連) と名前を指定します。

次の例では、フォルダ `AWS CLI` に をインストールします `/Users/myusername/aws-cli`。

```
$ installer -pkg AWSCLI2.pkg \  
            -target CurrentUserHomeDirectory \  
            -applyChoiceChangesXML choices.xml
```

- 通常、標準のユーザーアクセス許可では `$PATH` 内のフォルダへの書き込みが許可されないため、このモードのインストーラは `aws` および `aws_completer` プログラムにシンボリックリンクを追加しようとしません。を正しく実行する AWS CLI には、インストーラの完了後、シンボリックリンクを手動で作成する必要があります。`$PATH` に書き込み可能なフォルダが含まれていて、そのフォルダをターゲットのパスとして指定すると、`sudo` なしで以下のコマンドを実行できます。`$PATH` に書き込み可能なフォルダが含まれていない場合は、`sudo` を使用して、指定したターゲットフォルダに対する書き込みアクセス許可を取得する必要があります。シンボリックリンクのデフォルトの場所は `/usr/local/bin/` です。

```
$ sudo ln -s /folder/installed/aws-cli/aws /usr/local/bin/aws  
$ sudo ln -s /folder/installed/aws-cli/aws_completer /usr/local/bin/  
aws_completer
```

インストールが完了すると、デバッグログが `/var/log/install.log` に書き込まれます。

5. シェルが `aws` で `$PATH` コマンドを検索して実行できることを確認するには、次のコマンドを使用します。

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Darwin/23.3.0 botocore/2.4.5
```

`aws` コマンドが見つからない場合は、端末を再起動するか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

Windows

インストール要件

- インストールする AWS CLI バージョン 2 のリリースがわかります。バージョンのリストについては、[AWS CLI のバージョン 2 の変更ログ](#)を参照してくださいGitHub。
- Microsoft がサポートするバージョンの 64 ビット Windows AWS CLI では がサポートされています。
- ソフトウェアをインストールするための管理者権限

インストール手順

Windows で AWS CLI バージョン 2 の現在のインストールを更新するには、以前のバージョンを上書きするために更新するたびに新しいインストーラをダウンロードします。AWS CLI は定期的に更新されます。最新バージョンがいつリリースされたかを確認するには、「」の[AWS CLI 「バージョン 2 の変更ログ](#)」を参照してくださいGitHub。

1. Windows (64 ビット) 用インストーラをダウンロードして、次のいずれかの方法で実行 AWS CLI MSIします。
 - MSI インストーラのダウンロードと実行：特定のバージョンの のダウンロードリンクを作成するには AWS CLI、ファイル名にハイフンとバージョン番号を追加します。

```
https://awscli.amazonaws.com/AWSCLIV2-version.number.msi
```

この例では、バージョンのファイル名 **2.0.30** は次のリンク `AWSCLIV2-2.0.30.msi` になります: <https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi>。

- `msiexec` コマンドを使用する: または、`msiexec` コマンドにリンクを追加することでMSI インストーラを使用できます。の特定のバージョンでは AWS CLI、ファイル名にハイフンとバージョン番号を追加します。

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2-version.number.msi
```

この例では、バージョンのファイル名 **2.0.30** は次のリンク `AWSCLIV2-2.0.30.msi` になります <https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi>。

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi
```

`msiexec` で使用できるさまざまなパラメータについては、Microsoft Docs ウェブサイトの [msiexec](#) を参照してください。

バージョンのリストについては、[AWS CLI のバージョン 2 の変更ログ](#) を参照してくださいGitHub。

2. インストールを確認するには、[スタート] メニューを開き、`cmd` を検索してコマンドプロンプトウィンドウを開いて、コマンドプロンプトで `aws --version` コマンドを使用します。

```
C:\> aws --version  
aws-cli/2.17.20 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

Windows がプログラムを見つけることができない場合、パスを更新するためにコマンドプロンプトウィンドウを閉じて再度開くか、「[エラーのトラブルシューティング](#)」のトラブルシューティングに従ってください。

AWS CLI インストールおよびアンインストールエラーのトラブルシューティング

のインストールまたはアンインストール後に問題が発生した場合は AWS CLI、トラブルシューティングの手順[エラーのトラブルシューティング](#)については、「」を参照してください。最も関連性の高いトラブルシューティングステップについては、「[the section called “コマンドが見つからないエラー”](#)」、「[the section called “「aws --version」コマンドが、インストールしたのとは異なるバージョンを返す”](#)」、および「[the section called “aws --version 「」コマンドは、AWS CLI”](#)」を参照してください。

次のステップ

の手順を完了[the section called “前提条件”](#)し、 をインストールしたら AWS CLI、 を実行する必要があります[the section called “セットアップ”](#)。

ソース AWS CLI からの の構築とインストール

このトピックでは、サポートされているオペレーティングシステムに AWS Command Line Interface (AWS CLI) の最新リリースをインストールまたは更新する方法について説明します。

の最新リリースについては AWS CLI、[AWS CLI のバージョン 2 の変更ログ](#)を参照してください
GitHub。

Important

AWS CLI バージョン 1 と 2 は同じawsコマンド名を使用します。AWS CLI バージョン 1 を以前にインストールした場合は、「」を参照してください[AWS CLI バージョン 1 から AWS CLI バージョン 2 への移行](#)。

トピック

- [ソースからビルドする理由](#)
- [クイックステップ](#)
- [ステップ 1: すべての要件の設定](#)
- [ステップ 2: AWS CLI ソースインストールの構成](#)

- [ステップ 3: AWS CLI の構築](#)
- [ステップ 4: AWS CLI のインストール](#)
- [ステップ 5: インストールの検証](#)
- [ワークフローの例](#)
- [AWS CLI インストールおよびアンインストールエラーのトラブルシューティング](#)
- [次のステップ](#)

ソースからビルドする理由

AWS CLI は、ほとんどのプラットフォームと環境、および Docker イメージ用に [構築済みのインストーラ](#)として使用できます。

通常、これらのインストーラーはほとんどのユースケースに対応しています。ソースからインストールする手順は、インストーラーがカバーしていないユースケースに役立てるためのものです。ユースケースには次のようなものがあります。

- ビルド済みインストーラーは、使用中の環境をサポートしていません。例えば、32 ARM ビットは事前構築済みのインストーラではサポートされていません。
- ビルド済みインストーラーには、使用中の環境にはない依存関係があります。例えば、Alpine Linux では [musl](#) を使用していますが、現在のインストーラーでは glibc が必要になるため、ビルド済みインストーラーがすぐに動作しません。
- ビルド済みインストーラーには、使用中の環境によってアクセスが制限されているリソースが必要です。例えば、セキュリティが強化されたシステムでは、共有メモリへのアクセス許可が与えられない場合があります。これはフリーズした aws インストーラーに必要です。
- コードとパッケージのビルドプロセスは完全に制御されることが望ましいため、ビルド済みインストーラーは、パッケージマネージャーのメンテナンス担当者にとって障害となることがよくあります。ソースから構築することで、ディストリビューションメンテナーはより合理化されたプロセスで AWS CLI を更新できます。メンテナーを有効にすると brew、`、`、などのサードパーティーのパッケージマネージャーから をインストールする AWS CLI ときに yum、より多くの up-to-date バージョンの がお客様に提供されます apt。
- パッチ AWS CLI 機能を使用するお客様は、ソース AWS CLI から を構築およびインストールする必要があります。これは、AWS CLI GitHub リポジトリに変更を加える前に、ソースに加えた変更をテストしたいコミュニティメンバーにとって特に重要です。

クイックステップ

Note

すべてのコード例は、ソースディレクトリのルートから実行されることを前提としています。

ソース AWS CLI から を構築してインストールするには、このセクションの手順に従ってください。は [GNU Autotools](#) AWS CLI を活用してソースから をインストールします。最も単純なケースでは、GitHub リポジトリのルートからデフォルトのサンプルコマンドを実行して、 を AWS CLI ソースからインストール AWS CLI できます。

1. [環境のすべての要件を設定します](#)。これには、[GNUAutotools](#) で生成されたファイルを実行できることや、Python 3.8 以降がインストールされていることが含まれます。
2. ターミナルで、AWS CLI ソースフォルダの最上位に移動し、`./configure` コマンドを実行します。このコマンドは、必要なすべての依存関係についてシステムをチェックし、検出および指定された設定 AWS CLI に基づいて を構築およびインストールMakefileするための を生成します。

Linux and macOS

次の`./configure`コマンド例では、デフォルト設定 AWS CLI を使用して のビルド設定を設定します。

```
$ ./configure
```

Windows PowerShell

を呼び出すコマンドを実行する前にMSYS2、現在の作業ディレクトリを保持する必要があります。

```
PS C:\> $env:CHERE_INVOKING = 'yes'
```

次に、次の`./configure`コマンド例を使用して、Python 実行可能ファイルへのローカルパス AWS CLI を使用して のビルド設定を設定し、`C:\Program Files\`に をインストールし AWSCLI、すべての依存関係をダウンロードします。


```
PS C:\> C:\msys64\usr\bin\bash -lc " PYTHON='C:\path\to\python.exe' ./configure --prefix='C:\Program Files\AWSCLI' --with-download-deps "
```

詳細、使用可能な構成オプション、およびデフォルト設定情報については、「[the section called “ステップ 2: AWS CLI ソースインストールの構成”](#)」セクションを参照してください。

3. `make` コマンドを実行します。このコマンドは、設定 AWS CLI に従って を構築します。

次の `make` コマンド例は、既存の `./configure` 設定を使用してデフォルトオプションでビルドされます。

Linux and macOS

```
$ make
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "make"
```

詳細と使用可能なビルドオプションについては、「[the section called “ステップ 3: AWS CLI の構築”](#)」セクションを参照してください。

4. `make install` コマンドを実行します。このコマンドは、ビルドされた AWS CLI をご使用のシステム上で構成された場所にインストールします。

次の `make install` コマンド例では、デフォルトのコマンド設定を使用して、ビルドされた AWS CLI をインストールし、構成された場所にシンボリックリンクを作成します。

Linux and macOS

```
$ make install
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "make install"
```

インストール後、以下 AWS CLI を使用して へのパスを追加します。

```
PS C:\> $Env: PATH +=";C:\Program Files\AWSCLI\bin\"
```

詳細と使用可能なインストールオプションについては、「[the section called “ステップ 4: AWS CLIのインストール”](#)」セクションを参照してください。

5. 次のコマンドを使用して、正常にインストール AWS CLI された を確認します。

```
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

インストールエラーのトラブルシューティングの手順については、「[the section called “AWS CLI インストールおよびアンインストールエラーのトラブルシューティング”](#)」セクションを参照してください。

ステップ 1: すべての要件の設定

ソース AWS CLI から を構築するには、事前に以下を完了する必要があります。

Note

すべてのコード例は、ソースディレクトリのルートから実行されることを前提としています。

1. AWS CLI GitHub リポジトリを削除するか、AWS CLI ソース tarball をダウンロードして、ソースをダウンロードします。手順は、以下のいずれかです。
 - から[AWS CLI リポジトリ](#)をフォークしてクローンしますGitHub。詳細については、GitHubDocs の「[リポジトリをフォークする](#)」を参照してください。
 - 次のコマンドを使用して、最新のソース tarball を <https://awscli.amazonaws.com/awscli.tar.gz> でダウンロードし、コンテンツを抽出します。

```
$ curl -o awscli.tar.gz https://awscli.amazonaws.com/awscli.tar.gz
$ tar -xzf awscli.tar.gz
```

Note

特定のバージョンをダウンロードするには、次のリンク形式を使用します: <https://awscli.amazonaws.com/awscli-versionnumber.tar.gz>

例えば、バージョン 2.10.0 の場合、リンクは次のようになります: <https://awscli.amazonaws.com/awscli-2.10.0.tar.gz>

ソースバージョンは、AWS CLI の 2.10.0 のバージョン以降で使用できます。

(オプション) 以下の手順を完了してダウンロードした zip ファイルの整合性を確認する

1. GnuPG ツールを使用してこの署名を検証するには、次のステップを行います。

AWS CLI インストーラパッケージ .zip ファイルは、PGP 署名を使用して暗号的に署名されます。ファイルの破損や変更がある場合、この検証は失敗します。その場合、インストーラを続行しないでください。

2. ご利用のパッケージマネージャーを使用し、`gpg` コマンドをダウンロードしてインストールします。GnuPG の詳細については、[GnuPG のウェブサイト](#)を参照してください。
3. パブリックキーファイルを作成するには、テキストファイルを作成し、次のテキストに貼り付けます。

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
mQINBF2Cr7UBEADJZHcgus0Jl17ENSyumXh85z0TRV0xJorM2B/JL0kH0yigQ1uUG
ZMLhENaG0bYatdrKP+3H911vK050pXwn0/R7fB/FSTouki4ciIx50uLlnJZIxSzx
PqG10mkxImLNBGwoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+Lb1HAgEt5G
TfNxEKJ8soPLYWmwDH6HWCnjZ/aIQRBTIQ05uVeEoYxSh6w0ai7ss/KveoSNBbYz
gbdzoqI2Y8cgH2nbfgp3DSasaLZEdCSsIsK1u05CinE7k2qZ7KgKAUIcT/cR/grk
C6VwsnDU00UCideXcQ8WeHutqvgZH1JgKDbznoIzeQHJD238GEu+eKhRHcz8/jeG
94zkcqJ0z3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3Tqgvdx4bdkhf5cH+7NtW0
1rFj6UwAsGukBTA0xC01/dnSmZhJ7Z1KmEWi1ro/g0rjt0xqRQut1IqG22TaqoPG
fYVN+en3Zwbt97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3IbJTBfqq1XmZ719yWg
EEUJY01b2XrSuPwm139beWdKM8kzr10jn10m6+1pTRCBfo0wa9F8YZRhHPAkwKkX
XDe0GpWRj4oh0x0d2GWkyV5xyN14p2tQ0Cd00Dmz80yUTgRpPVQUt0EhXQARAQAB
tCFBV1MgQ0xJIFR1YW0gPGF3cy1jbG1AYW1hem9uLmNvbT6JAlQEEwEiAD4WIQT7
Xbd/1cEYUauraimMQrMRnJHXAUCXYKvtQIbAwUJB4TOAAULCQgHAgYVCgkICwIE
FgIDAQIeAQIXgAAKCRcmMQrMRnJHXJIXEACHLUIkg80uPUkGjE3jejvQSA1aWuAM
zy6fdpd1RUz6M6nmsUh0ExjVivibEJpzK5mhuSZ41b0vJ2ZUPgCv4zs2nBd7BGJ
MxKiWgBREgVtdqZ0SzyYH4PYCJSE732x/Fw9hfnh1dMTXNcrQXzw0mmFNNegG00x
au+Vnpr5Kz3smiTrIwZbRudo1ijhCYPQ7t5CMp9kjC6b0bvy1hSIg2xNbMAN/Do
```

```

ikebA136uA6Y/Uczjj3GxZW4ZWeFirMidKbtqvUz2y0UFszobjiBSqZZHCreC34B
hw9bFNpuWC/0SrxXgohdsc6vK50pDGdV5kM2qo9tMQ/izsAwTh/d/GzZv8H41V9e0
tEis+EpR497PaxKKh9tJf0N6Q1YLRHof5xePZt0I1S3gfvsh5hXA3HJ9yIxb8T0H
QYmVr3aIUes20i6meI3fuV36VFupwfrTKaL7VXnsrK2fq5cRvyJLNzXucg0WAjPF
RrAGLzY7nP1xeg1a0aeP+pdsqjq1PJom80CWc1+6DWbg0jsC74WoesAqgBIt0DMB
rsal1y/q+bPzpsnWjzHV8+1/EtZmSc8ZUGSJ0Pkfc7h0bnfk118h+1QtKTjZme4d
H17gsBJr+opwJw/Zio2LMjQB0qlm3K1A4zFTh7wBC7He6KPQea1p2XAMgtvATtNe
YLZATHZKTJyiqA==
=vY0k
-----END PGP PUBLIC KEY BLOCK-----

```

参考までに、パブリックキーの詳細を以下に示します。

```

Key ID:          A6310ACC4672
Type:           RSA
Size:           4096/4096
Created:        2019-09-18
Expires:        2023-09-17
User ID:        AWS CLI Team <aws-cli@amazon.com>
Key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C

```

4. 次のコマンドを使用して AWS CLI パブリックキーをインポートし、置き換えます。 *public-key-file-name* 作成したパブリックキーのファイル名。

```

$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>"
imported
gpg: Total number processed: 1
gpg:          imported: 1

```

5. <https://awscli.amazonaws.com/awscli.tar.gz.sig> でダウンロードしたパッケージ AWS CLI の署名ファイルをダウンロードします。これは、対応する tarball ファイルと同じパスと名前を持っていますが、拡張子は .sig です。tarball ファイルと同じパスに保存します。または、次のコマンドブロックを使用します。

```

$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli.tar.gz.sig

```

6. 署名を検証し、.sig ファイル名と .zip ファイル名の両方をパラメータとして gpg コマンドに渡します。

```
$ gpg --verify awscliv2.sig awscli.tar.gz
```

出力は以下の例のようになります。

```
gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:                using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672
475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
```

Important

出力内の警告は予想されており、問題を示すものではありません。これは、パーソナルPGPキー (持っている場合) と AWS CLI PGPキーの間に信頼チェーンがないために発生します。詳細については、「[信用の輪 \(Web of Trust\)](#)」を参照してください。

2. `configure` やなどの [GNU Autotools](#) 生成ファイルを実行できる環境がありますMakefile。これらのファイルは、POSIXプラットフォーム間で広く移植されます。

Linux and macOS

Autotools がご使用の環境にまだインストールされていない場合、または Autotools を更新する必要がある場合は、GNUドキュメントの「[Autotools \(ユーザーとして\) をインストールする方法](#)」または「[基本インストール](#)」にあるインストール手順に従ってください。

Windows PowerShell

Warning

Windows 環境の場合は、ビルド済みインストーラーを使用することをお勧めします。ビルド済みインストーラーのインストール手順については、「[the section called “インストール/更新”](#)」を参照してください

Windows には POSIX 準拠のシェルが付属していないため、ソース AWS CLI から をインストールするために追加のソフトウェアをインストールする必要があります。[MSYS2](#) は、特に Autotools が使用する POSIX ベースのスクリプト用に、Windows ソフトウェアの構築とインストールに役立つツールとライブラリのコレクションを提供します。

1. をインストールします MSYS2。のインストールと使用の詳細については MSYS2、MSYS2 ドキュメントの「[インストールと使用方法](#)」を参照してください。
2. MSYS2 ターミナルを開き、次のコマンドを使用して Autotools をインストールします。

```
$ pacman -S autotools
```

Note

このガイドの Windows 用設定、ビルド、およびインストールコードの例を使用する場合、のデフォルトの MSYS2 インストールパス `C:\msys64\usr\bin\bash` が前提となります。MSYS2 内で PowerShell を呼び出すときは、bash コマンドを引用符で囲んだ次の形式を使用します。

```
PS C:\> C:\msys64\usr\bin\bash -lc "command example"
```

次のコマンド例では、`./configure` コマンドを呼び出します。

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure"
```

3. Python 3.8 以降のインタプリタがインストールされています。必要な最小 Python バージョンは、[および Tools の AWS SDKs 公式 Python サポートポリシー](#)と同じタイムラインに従います。インタプリタは、その日付から 6 か月後にのみサポートされます end-of-support。
4. (オプション) AWS CLI のビルドとランタイム Python ライブラリの依存関係をすべてインストールします。`./configure` コマンドは、不足している依存関係がないかどうか、また、そのインストール方法について通知します。

これらの依存関係は、構成で自動的にインストールして使用できます。詳細については、「[the section called “依存関係のダウンロード”](#)」を参照してください。

ステップ 2: AWS CLI ソースインストールの構成

を構築およびインストールするための設定 AWS CLI は、`configure` スクリプトを使用して指定されます。すべての構成オプションを文書化するには、`--help` オプションを使用して `configure` スクリプトを実行します。

Linux and macOS

```
$ ./configure --help
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure --help"
```

最も重要なオプションは次のとおりです。

- [インストール場所](#)
- [Python インタープリタ](#)
- [依存関係のダウンロード](#)
- [インストールタイプ](#)

インストール場所

のソースインストール AWS CLI では、2 つの設定可能なディレクトリを使用して をインストールします AWS CLI。

- `libdir` - AWS CLI がインストールされる親ディレクトリ。AWS CLI インストールへのパスは `<libdir-value>/aws-cli`。Linux と macOS のデフォルト `libdir` 値は `/usr/local/lib` で、デフォルトのインストールディレクトリを `/usr/local/lib/aws-cli` にします
- `bindir` - AWS CLI 実行可能ファイルがインストールされているディレクトリ。デフォルトの場所は `/usr/local/bin` です。

以下の `configure` オプションは使用するディレクトリを制御します。

- `--prefix` - インストールに使用するディレクトリプレフィックスを設定します。Linux と macOS のデフォルト値は `/usr/local` です。

- `--libdir` - AWS CLIのインストールに使用する `libdir` を設定します。デフォルト値は `<prefix-value>/lib` です。`--libdir` と `--prefix` が両方とも指定されていない場合、Linux と macOS のデフォルトは `/usr/local/lib/` です。
- `--bindir` - および `aws_completer` 実行可能ファイルのインストール `bindir` に使用する AWS CLI `aws` を設定します。デフォルト値は `<prefix-value>/bin` です。`bindir` と `--prefix` が両方とも指定されていない場合、Linux と macOS のデフォルトは `/usr/local/bin/` です。

Linux and macOS

次のコマンド例では、`--prefix` オプションを使用して AWS CLIをローカルユーザーにインストールします。このコマンドは、AWS CLI に `をインストール$HOME/.local/lib/aws-cli` し、`に実行可能ファイルをインストールします$HOME/.local/bin`。

```
$ ./configure --prefix=$HOME/.local
```

次のコマンド例では、`--libdir` オプションを使用して、`を アドオンアプリケーション AWS CLI` として `/opt` ディレクトリにインストールします。このコマンドは、AWS CLI `を にインストール/opt/aws-cli` し、実行可能ファイルをデフォルトの場所である `にインストールします/usr/local/bin`。

```
$ ./configure --libdir=/opt
```

Windows PowerShell

次のコマンド例では、`--prefix` オプションを使用して AWS CLIをローカルユーザーにインストールします。このコマンドは、AWS CLI に `をインストール$HOME/.local/lib/aws-cli` し、`に実行可能ファイルをインストールします$HOME/.local/bin`。

```
$ C:\msys64\usr\bin\bash -lc ".\configure --prefix='C:\Program Files\AWSCLI'"
```

次のコマンド例では、`--libdir` オプションを使用して、`を アドオンアプリケーション AWS CLI` として `/opt` ディレクトリにインストールします。このコマンドは、AWS CLI `を にインストールしますC:\Program Files\AWSCLI\opt\aws-cli`。

Python インタープリタ

Note

Windows にインストールする場合は、Python インタープリタを指定することを強くお勧めします。

`./configure` スクリプトは、Autoconf マクロ AWS CLI を使用した の構築と実行に使用するインストール済みの Python 3.8 [AM_PATH_PYTHON](#) 以降のインタープリタを自動的に選択します。

使用する Python インタープリタは、`configure` スクリプトの実行時に `PYTHON` 環境変数を使用して明示的に設定できます。

Linux and macOS

```
$ PYTHON=/path/to/python ./configure
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "PYTHON='C:\i>path\to\python' ./configure"
```

依存関係のダウンロード

デフォルトでは、AWS CLI のすべてのビルドとランタイムの依存関係がシステムに既にインストールされている必要があります。これには Python ライブラリの依存関係も含まれます。`configure` スクリプトの実行時にすべての依存関係がチェックされ、システムに Python の依存関係がない場合、`configure` スクリプトはエラーになります。

次のコード例は、システムに依存関係がない場合にエラーになります。

Linux and macOS

```
$ ./configure
checking for a Python interpreter with version >= 3.8... python
checking for python... /Users/username/.envs/env3.11/bin/python
checking for python version... 3.11
checking for python platform... darwin
checking for GNU default python prefix... ${prefix}
```

```
checking for GNU default python exec_prefix... ${exec_prefix}
checking for python script directory (pythondir)... ${PYTHON_PREFIX}/lib/python3.11/
site-packages
checking for python extension module directory (pyexecdir)... ${PYTHON_EXEC_PREFIX}/
lib/python3.11/site-packages
checking for --with-install-type... system-sandbox
checking for --with-download-deps... Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
125, in <module>
    main()
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
121, in main
    parsed_args.func(parsed_args)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
49, in validate
    validate_env(parsed_args.artifact)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/validate_env.py",
line 68, in validate_env
    raise UnmetDependenciesException(unmet_deps, in_venv)
validate_env.UnmetDependenciesException: Environment requires following Python
dependencies:

awscrt (required: ('>=0.12.4', '<0.17.0')) (version installed: None)

We recommend using --with-download-deps flag to automatically create a virtualenv
and download the dependencies.

If you want to manage the dependencies yourself instead, run the following pip
command:
/Users/username/.envs/env3.11/bin/python -m pip install --prefer-binary
'awscrt>=0.12.4,<0.17.0'

configure: error: "Python dependencies not met."
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure"
checking for a Python interpreter with version >= 3.8... python
checking for python... /Users/username/.envs/env3.11/bin/python
checking for python version... 3.11
checking for python platform... darwin
```

```
checking for GNU default python prefix... ${prefix}
checking for GNU default python exec_prefix... ${exec_prefix}
checking for python script directory (pythondir)... ${PYTHON_PREFIX}/lib/python3.11/
site-packages
checking for python extension module directory (pyexecdir)... ${PYTHON_EXEC_PREFIX}/
lib/python3.11/site-packages
checking for --with-install-type... system-sandbox
checking for --with-download-deps... Traceback (most recent call last):
  File "<frozen runpy>", line 198, in _run_module_as_main
  File "<frozen runpy>", line 88, in _run_code
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
125, in <module>
    main()
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
121, in main
    parsed_args.func(parsed_args)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/__main__.py", line
49, in validate
    validate_env(parsed_args.artifact)
  File "/Users/username/aws-code/aws-cli/./backends/build_system/validate_env.py",
line 68, in validate_env
    raise UnmetDependenciesException(unmet_deps, in_venv)
validate_env.UnmetDependenciesException: Environment requires following Python
dependencies:

awsCRT (required: ('>=0.12.4', '<0.17.0')) (version installed: None)

We recommend using --with-download-deps flag to automatically create a virtualenv
and download the dependencies.

If you want to manage the dependencies yourself instead, run the following pip
command:
/Users/username/.envs/env3.11/bin/python -m pip install --prefer-binary
'awsCRT>=0.12.4,<0.17.0'

configure: error: "Python dependencies not met."
```

必要な Python 依存関係を自動的にインストールするには、`--with-download-deps` オプションを使用してください。このフラグを使用すると、ビルドプロセスは次の処理を行います。

- Python ライブラリの依存関係チェックをスキップします。

- 必要なすべての Python 依存関係をダウンロードし、ダウンロードされた依存関係のみを使用してビルド AWS CLI 中に make を構築するように設定します。

以下の configure コマンドの例では、`--with-download-deps` オプションを使用して Python の依存関係をダウンロードおよび使用しています。

Linux and macOS

```
$ ./configure --with-download-deps
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure --with-download-deps"
```

インストールタイプ

ソースインストールのプロセスでは、次のインストールタイプがサポートされます。

- `system-sandbox` - (デフォルト) 分離された Python 仮想環境を作成し、AWS CLI を仮想環境にインストールし、仮想環境の `aws aws_completer` および 実行可能ファイルへのシンボリックリンクを行います。このインストール AWS CLI は、ランタイム用に選択した Python インタープリタに直接依存します。

これは、システムに AWS CLI をインストールするための軽量インストールメカニズムであり、仮想環境でインストールをサンドボックス化することで Python のベストプラクティスに従います。このインストールは、Python のインストールと組み合わせたインストールで、可能な限りスムーズな方法でソース AWS CLI から をインストールしたいお客様を対象としています。

- `portable-exe` - をフリーズ AWS CLI して、同様のアーキテクチャの環境に分散できるスタンドアロン実行可能ファイルにします。これは、AWS CLI の公式ビルド済み実行ファイルを生成するのと同じプロセスです。`portable-exe` は、AWS CLI のランタイムに使用する `configure` ステップで選択した Python インタープリタのコピーをフリーズします。これにより、Python インタープリタがない他のマシンに移動させることができます。

このタイプのビルドは、AWS CLI インストールが環境のインストール済み Python バージョンに結合されていないことを確認し、Python がまだインストールされていない他のシステムにビルドを分散できるため便利です。これにより、使用する AWS CLI 実行可能ファイルの依存関係とセキュリティを制御できます。

インストールタイプを構成するには、`--with-install-type` オプションを使用して `portable-exe` または `system-sandbox` の値を指定します。

次の `./configure` コマンド例は、`portable-exe` の値を指定します。

Linux and macOS

```
$ ./configure --with-install-type=portable-exe
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "./configure --with-install-type=portable-exe"
```

ステップ 3: AWS CLIの構築

`make` コマンドを使用して、設定 AWS CLI を使用して を構築します。

Linux and macOS

```
$ make
```

Windows PowerShell

```
PS C:\> C:\msys64\usr\bin\bash -lc "make"
```

Note

`make` コマンドを使用すると、次の手順がバックグラウンドで完了します。

1. 仮想環境は、Python [venv](#) モジュールを使用してビルドディレクトリに作成されます。仮想環境は、[Python 標準ライブラリにベンダリングされているバージョンの pip](#) でブートストラップされます。
2. Python ライブラリの依存関係をコピーします。configure コマンドで `--with-download-deps` フラグが指定されているかどうかに応じて、このステップは次のいずれかを実行します。

- `--with-download-deps` は指定されます。Python の依存関係は `pip` でインストールされます。これには、`wheel`、`setuptools`、およびすべての AWS CLI ランタイム依存関係が含まれます。`portable-exe` をビルドしている場合は、`pyinstaller` がインストールされます。これらの要件はすべて、[pip-compile](#) から生成されるロックファイルで指定されています。
 - `--with-download-deps` は指定されません。Python インタープリタのサイトパッケージの Python ライブラリとスクリプト (例: `pyinstaller`) は、ビルドに使用されている仮想環境にコピーされます。
3. AWS CLI コードベースで `pip install` 直接実行して、オフラインのツリー内ビルドを実行し、ビルド仮想環境に をインストール AWS CLI します。このインストールでは、`pip` フラグ [--no-build-isolation](#)、[--use-feature=in-tree-build](#)、[--no-cache-dir](#)、および を使用します [--no-index](#)。
 4. (オプション) `configure` コマンドで `--install-type` が `portable-exe` に設定されている場合、[pyinstaller](#) を使用してスタンドアロン実行ファイルをビルドします。

ステップ 4: AWS CLI のインストール

`make install` コマンドは、ビルドされた AWS CLI をシステム上の設定済みの場所にインストールします。

Linux and macOS

次のコマンド例では、設定とビルド設定 AWS CLI を使用して をインストールします。

```
$ make install
```

Windows PowerShell

次のコマンド例では、設定とビルド設定 AWS CLI を使用して をインストールし、 のパスを持つ環境変数を追加します AWS CLI。

```
PS C:\> C:\msys64\usr\bin\bash -lc " make install "  
PS C:\> $Env: PATH +=";C:\Program Files\AWSCLI\bin\"
```

make install ルールは [DESTDIR](#) 変数をサポートします。この変数を指定すると、AWS CLI をインストールするときに、既に構成されているインストールパスの前に、指定されたパスが追加されます。デフォルトでは、この変数には値が設定されていません。

Linux and macOS

次のコード例では、`--prefix=/usr/local` フラグを使用してインストール場所を構成し、`DESTDIR=/tmp/stage` を `make install` コマンドに使用してその宛先を変更します。これらのコマンドを実行すると、AWS CLI が `/tmp/stage/usr/local/lib/aws-cli` にインストールされ、その実行ファイルが `/tmp/stage/usr/local/bin` に配置されます。

```
$ ./configure --prefix=/usr/local
$ make
$ make DESTDIR=/tmp/stage install
```

Windows PowerShell

次のコード例では、`--prefix=\awscli` フラグを使用してインストール場所を構成し、`DESTDIR=C:\Program Files` を `make install` コマンドに使用してその宛先を変更します。これらのコマンドを実行すると、AWS CLI が `C:\Program Files\awscli` にインストールされます。

```
$ ./configure --prefix=\awscli
$ make
$ make DESTDIR='C:\Program Files' install
```

Note

`make install` コマンドを実行すると、次の手順がバックグラウンドで完了します。

1. 次のいずれかを構成済みのインストールディレクトリに移動します。
 - インストールタイプが `system-sandbox` の場合、ビルドした仮想環境を移動します。
 - インストールタイプが `portable-exe` の場合、ビルドしたスタンドアロン実行ファイルを移動します。

2. 構成した bin ディレクトリで、aws および aws_completer 実行ファイルの両方にシンボリックリンクを作成します。

ステップ 5: インストールの検証

次のコマンドを使用して、正常にインストール AWS CLI された を確認します。

```
$ aws --version  
aws-cli/2.17.20 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

aws コマンドが認識されない場合は、新しいシンボリックリンクを更新するために、ターミナルの再起動が必要になる場合があります。のインストールまたはアンインストール後に追加の問題が発生した場合は AWS CLI、一般的なトラブルシューティング手順[エラーのトラブルシューティング](#)については、「」を参照してください。

ワークフローの例

このセクションでは、ソースからインストールするための基本的なワークフローの例をいくつか紹介します。

Linux および macOS の基本的なインストール

次の例は、AWS CLI が のデフォルトの場所にインストールされる基本的なインストールワークフローです/usr/local/lib/aws-cli。

```
$ cd path/to/cli/repository/  
$ ./configure  
$ make  
$ make install
```

Windows 自動インストール

Note

このワークフローを使用するには、管理者 PowerShell として を実行する必要があります。

MSYS2 は CI 設定で自動で使用できます。「MSYS2ドキュメント」の「[CI MSYS2で使用する](#)」を参照してください。

Downloaded Tarball

awscli.tar.gz ファイルをダウンロードし、解凍して、AWS CLIをインストールします。次のコマンドを使用する場合は、次のパスを置き換えます。

- C:\msys64\usr\bin\bash MSYS2パスの場所。
- .\awscli-2.x.x\ を解凍した awscli.tar.gz フォルダ名にします。
- PYTHON='C:\path\to\python.exe' をローカルの Python パスにします。

次のコード例では、PowerShell を使用して AWS CLI から の構築とインストールを自動化し MSYS2、使用する Python のローカルインストールを指定します。

```
PS C:\> curl -o awscli.tar.gz https://awscli.amazonaws.com/awscli.tar.gz #
Download the awscli.tar.gz file in the current working directory
PS C:\> tar -xvzf .\awscli.tar.gz # Extract awscli.tar.gz file
PS C:\> cd .\awscli-2.x.x\ # Navigate to the root of the extracted files
PS C:\> $env:CHERE_INVOKING = 'yes' # Preserve the current working directory
PS C:\> C:\msys64\usr\bin\bash -lc " PYTHON='C:\path\to\python.exe' ./configure --
prefix='C:\Program Files\AWSCLI' --with-download-deps "
PS C:\> C:\msys64\usr\bin\bash -lc "make"
PS C:\> C:\msys64\usr\bin\bash -lc "make install"
PS C:\> $Env:PATH +=";C:\Program Files\AWSCLI\bin\"
PS C:\> aws --version
aws-cli/2.17.20 Python/3.11.6 Windows/10 source-sandbox/AMD64 prompt/off
```

GitHub Repository

awscli.tar.gz ファイルをダウンロードし、解凍して、AWS CLIをインストールします。次のコマンドを使用する場合は、次のパスを置き換えます。

- C:\msys64\usr\bin\bash MSYS2パスの場所。
- C:\path\to\cli\repository\ からクローンされた[AWS CLI リポジトリ](#)へのパス。
GitHub詳細については、GitHub「Docs」の「[リポジトリをフォークする](#)」を参照してください。
- PYTHON='C:\path\to\python.exe' をローカルの Python パスにします。

次のコード例では、PowerShell を使用して AWS CLI から の構築とインストールを自動化し MSYS2、使用する Python のローカルインストールを指定します。

```
PS C:\> cd C:path\to\cli\repository\  
PS C:\> $env:CHERE_INVOKING = 'yes' # Preserve the current working directory  
PS C:\> C:\msys64\usr\bin\bash -lc " PYTHON='C:\path\to\python.exe' ./configure --  
prefix='C:\Program Files\AWSCLI' --with-download-deps "  
PS C:\> C:\msys64\usr\bin\bash -lc "make"  
PS C:\> C:\msys64\usr\bin\bash -lc "make install"  
PS C:\> $Env:PATH +=";C:\Program Files\AWSCLI\bin\  
PS C:\> aws --version
```

Alpine Linux コンテナ

以下は、Alpine [用の事前構築済みバイナリの代替として、Alpine](#) Linux コンテナ AWS CLI での作業 インストールを取得するために使用できる Dockerfile の例です。この例では、`AWSCLI_VERSION` 必要な AWS CLI バージョン番号 :

```
FROM python:3.8-alpine AS builder  
  
ENV AWSCLI_VERSION=2.10.1  
  
RUN apk add --no-cache \  
    curl \  
    make \  
    cmake \  
    gcc \  
    g++ \  
    libc-dev \  
    libffi-dev \  
    openssl-dev \  
    && curl https://awscli.amazonaws.com/awscli-${AWSCLI_VERSION}.tar.gz | tar -xz \  
    && cd awscli-${AWSCLI_VERSION} \  
    && ./configure --prefix=/opt/aws-cli/ --with-download-deps \  
    && make \  
    && make install  
  
FROM python:3.8-alpine  
  
RUN apk --no-cache add groff
```

```
COPY --from=builder /opt/aws-cli/ /opt/aws-cli/  
  
ENTRYPOINT ["/opt/aws-cli/bin/aws"]
```

このイメージはビルドされ、Amazon Linux 2 上に構築されているのと同様のコンテナから AWS CLI 呼び出されます。

```
$ docker build --tag awscli-alpine .  
$ docker run --rm -it awscli-alpine --version  
aws-cli/2.2.1 Python/3.8.11 Linux/5.10.25-linuxkit source-sandbox/x86_64.alpine.3  
prompt/off
```

このイメージの最終サイズは、公式の AWS CLI Docker イメージのサイズよりも小さくなります。公式の Docker イメージについては、「[the section called “Amazon ECR Public/Docker”](#)」を参照してください。

AWS CLI インストールおよびアンインストールエラーのトラブルシューティング

インストールエラーのトラブルシューティング手順については、一般的なトラブルシューティング手順を「[エラーのトラブルシューティング](#)」で参照してください。最も関連性の高いトラブルシューティングステップについては、「[the section called “コマンドが見つからないエラー”](#)」、「[the section called “「aws --version」コマンドが、インストールしたのとは異なるバージョンを返す”](#)」、および「[the section called “aws --version 「」コマンドは、AWS CLI”](#)」を参照してください。

トラブルシューティングガイドで説明されていない問題については、の [AWS CLI リポジトリ](#) で source-distribution ラベルの問題を検索します GitHub。既存の問題がエラーをカバーしていない場合は、[新しい問題を作成して](#)、AWS CLI メンテナンス担当者からサポートを受けてください。

次のステップ

をインストールしたら AWS CLI、 を実行する必要があります [the section called “セットアップ”](#)。

の公式 Amazon ECR Public または Docker イメージの実行 AWS CLI

このトピックでは、公式の Amazon Elastic Container Registry Public (Amazon ECR Public) または Docker Hub イメージを使用して、Docker で AWS CLI バージョン 2 を実行、バージョン管理、設定する方法について説明します。Docker の使用方法の詳細については、[Docker のドキュメント](#)を参照してください。

公式イメージは、AWS 直接サポートおよび維持する分離、移植性、セキュリティを提供します。これにより、自分でインストールを管理することなく、コンテナベースの環境で AWS CLI バージョン 2 を使用できます。

トピック

- [前提条件](#)
- [Amazon ECR Public と Docker Hub 間の決定](#)
- [公式 AWS CLI バージョン 2 イメージを実行する](#)
- [公式イメージのインターフェイスと下位互換性に関する注意事項](#)
- [特定のバージョンとタグの使用](#)
- [最新の公式イメージに更新する](#)
- [ホストファイル、認証情報、環境変数、構成を共有する](#)
- [docker run コマンドの短縮](#)

前提条件

Docker がインストールされている必要があります。インストール手順については、[Docker のウェブサイト](#)を参照してください。

Docker のインストールを確認するには、次のコマンドを実行し、出力があることを確認します。

```
$ docker --version
Docker version 19.03.1
```

Amazon ECR Public と Docker Hub 間の決定

AWS CLI イメージには Docker Hub ではなく Amazon ECR Public を使用することをお勧めします。Docker Hubでは、一般消費者向けのレート制限が厳しくなっているため、スロットリングの問

題が発生する可能性があります。さらに、Amazon ECR Public は複数のリージョンにイメージをレプリケートし、強力な可用性を提供し、リージョンの停止問題に対処します。

Docker Hub のレート制限の詳細については、Docker ウェブサイトの「[Understanding Docker Hub Rate Limiting \(Docker Hub のレート制限とは\)](#)」を参照してください。

公式 AWS CLI バージョン 2 イメージを実行する

`docker run` コマンドを初めて使用すると、最新のイメージがコンピュータにダウンロードされます。それ以降の `docker run` コマンドの使用は、ローカルコピーから実行されます。

AWS CLI バージョン 2 の Docker イメージを実行するには、`docker run` コマンドを使用します。

Amazon ECR Public

公式 AWS CLI バージョン 2 Amazon ECR Public イメージは、リポジトリの Amazon ECR Public [aws-cli/aws-cli](#) でホストされます。

```
$ docker run --rm -it public.ecr.aws/aws-cli/aws-cli command
```

Docker Hub

公式 AWS CLI バージョン 2 の Docker イメージは、`amazon/aws-cli` リポジトリの Docker Hub でホストされます。

```
$ docker run --rm -it amazon/aws-cli command
```

コマンドの機能は次のとおりです。

- `docker run --rm -it repository/name aws` 実行可能ファイルと同等です。このコマンドを実行するたびに、Docker はダウンロードしたイメージのコンテナをスピンアップし、`aws` コマンドを実行します。デフォルトでは、イメージはバージョン 2 AWS CLI の最新バージョンを使用します。

例えば、Docker で `aws --version` コマンドを呼び出すには、以下を実行します。

Amazon ECR Public

```
$ docker run --rm -it public.ecr.aws/aws-cli/aws-cli --version
```

```
aws-cli/2.17.20 Python/3.7.3 Linux/4.9.184-linuxkit boto-core/2.4.5dev10
```

Docker Hub

```
$ docker run --rm -it amazon/aws-cli --version
aws-cli/2.17.20 Python/3.7.3 Linux/4.9.184-linuxkit boto-core/2.4.5dev10
```

- `--rm` - コマンドが終了後にコンテナをクリーンアップすることを指定します。
- `-it` - で擬似 TTY を開くことを指定します `stdin`。これにより、`aws configure` および `aws help` コマンドなどを使用して、コンテナで実行されているときに AWS CLI バージョン 2 に入力を提供できます。`-it` を省略する場合は、以下の点を考慮してください。
 - スクリプトを実行している場合は、`-it` は不要です。
 - スクリプトでエラーが発生する場合は、Docker 呼び出しから `-it` を省略することで問題が解決する場合があります。
 - パイプ出力を実行しようとする `-it` でエラーの原因となることがあり、Docker 呼び出しから `-it` を省略することで、この問題が解決する場合があります。`-it` フラグを保持したままパイプ出力を実行する場合は、デフォルトで使用されている [クライアント側のページャー](#) AWS CLI を無効にすることで問題が解決できます。

`docker run` コマンドの詳細については、[Docker reference guide](#) を参照してください。

公式イメージのインターフェイスと下位互換性に関する注意事項

- イメージでサポートされる唯一のツールは AWS CLI です。必ず、`aws` 実行可能ファイルは直接実行する必要があります。例えば、`less` と `groff` がイメージに明示的にインストールされている場合でも、AWS CLI コマンドの外部で直接実行しないでください。
- `/aws` 作業ディレクトリは、ユーザーによってコントロールします。AWS CLI コマンドの実行についてユーザーから指示がない限り、イメージはこのディレクトリに書き込まれません。
- 最新のタグに依存する場合、下位互換性は保証されません。下位互換性を保証するには、特定の `<major.minor.patch>` タグに固定する必要があります。このタグはイミュータブルであり、1 回のみプッシュされます。

特定のバージョンとタグの使用

公式 AWS CLI バージョン 2 イメージには、バージョン から始まる複数のバージョンがあります 2.0.6。AWS CLI バージョン 2 の特定のバージョンを実行するには、適切なタグを `docker`

run コマンドに追加します。初めてタグを指定して `docker run` コマンドを使用すると、そのタグの最新のイメージがコンピュータにダウンロードされます。それ以降、そのタグで `docker run` コマンドを使用する場合は、ローカルコピーから実行されます。

次の 2 種類のタグを使用できます。

- `latest` – イメージのバージョン 2 AWS CLI の最新バージョンを定義します。AWS CLI バージョン 2 の最新バージョンを使用する場合は、`latest` タグを使用することをお勧めします。ただし、このタグに依存する場合の下位互換性は保証されません。`latest` タグは、`docker run` コマンドでデフォルトで使用されます。`latest` タグを明示的に使用するには、タグをコンテナイメージ名に追加します。

Amazon ECR Public

```
$ docker run --rm -it public.ecr.aws/aws-cli/aws-cli:latest command
```

Docker Hub

```
$ docker run --rm -it amazon/aws-cli:latest command
```

- `<major.minor.patch>` – イメージのバージョン 2 AWS CLI の特定のバージョンを定義します。本番環境で公式イメージを使用する場合は、バージョン 2 AWS CLI の特定のバージョンを使用して下位互換性を確保することをお勧めします。例えば、バージョン 2.0.6 を実行するには、コンテナイメージ名にバージョンを追加します。

Amazon ECR Public

```
$ docker run --rm -it public.ecr.aws/aws-cli/aws-cli:2.0.6 command
```

Docker Hub

```
$ docker run --rm -it amazon/aws-cli:2.0.6 command
```

最新の公式イメージに更新する

最新のイメージは、`docker run` コマンドの初回使用時にのみコンピュータにダウンロードされるため、更新したイメージは手動でプルする必要があります。手動で最新バージョンに更新するには、`latest` タグ付きイメージをプルすることをお勧めします。イメージをプルすると、コンピュータに最新バージョンがダウンロードされます。

Amazon ECR Public

```
$ docker pull public.ecr.aws/aws-cli/aws-cli:latest
```

Docker Hub

```
$ docker pull amazon/aws-cli:latest
```

ホストファイル、認証情報、環境変数、構成を共有する

AWS CLI バージョン 2 はコンテナで実行されるため、デフォルトでは、CLIは設定と認証情報を含むホストファイルシステムにアクセスできません。ホストファイルシステム、認証情報、および設定をコンテナと共有するには、ホストシステムの ~/.aws ディレクトリを /root/.aws のコンテナにマウントして、-v コマンドに docker run フラグを付けます。これにより、コンテナで実行されている AWS CLI バージョン 2 がホストファイル情報を見つけることができます。

Amazon ECR Public

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws public.ecr.aws/aws-cli/aws-cli command
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%\aws:/root/.aws public.ecr.aws/aws-cli/aws-cli command
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\aws:/root/.aws public.ecr.aws/aws-cli/aws-cli command
```

Docker Hub

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws amazon/aws-cli command
```


Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%\aws:/root/.aws amazon/aws-cli command
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\aws:/root/.aws amazon/aws-cli command
```

-v フラグとマウントの詳細については、[Docker reference guide](#) を参照してください。

Note

config および credentials の詳細については、「[the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)」を参照してください。

例 1: 認証情報と設定の提供

この例では、Amazon Simple Storage Service (Amazon S3) でバケットを一覧表示する `s3 ls` コマンドの実行時に、ホストの認証情報および設定を提供しています。次の例では、AWS CLI 認証情報と設定ファイルのデフォルトの場所を使用して、別の場所を使用するには、ファイルパスを変更します。

Amazon ECR Public

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws public.ecr.aws/aws-cli/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%\aws:/root/.aws public.ecr.aws/aws-cli/aws-
cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws public.ecr.aws/aws-cli/
aws-cli s3 ls
```

Docker Hub

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws amazon/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%.aws:/root/.aws amazon/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws amazon/aws-cli s3 ls
```

特定システムの環境変数は、`-e` フラグを使用して呼び出すことができます。環境変数を使用するには、変数を名前呼び出します。

Amazon ECR Public

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e ENVVAR_NAME public.ecr.aws/aws-cli/
aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%.aws:/root/.aws -e ENVVAR_NAME
public.ecr.aws/aws-cli/ aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -e ENVVAR_NAME  
public.ecr.aws/aws-cli/aws-cli s3 ls
```

Docker Hub

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e ENVVAR_NAME amazon/aws-cli s3 ls  
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%.aws:/root/.aws -e ENVVAR_NAME amazon/aws-cli  
s3 ls  
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\.aws:/root/.aws -e ENVVAR_NAME amazon/  
aws-cli s3 ls
```

例 2: ホストシステムへの Amazon S3 ファイルのダウンロード

一部の AWS CLI バージョン 2 コマンドでは、コンテナ内のホストシステムからファイルを読み取るか、コンテナからホストシステムにファイルを書き込むことができます。

この例では、現在の作業ディレクトリをコンテナの S3 ディレクトリにマウントして、s3://aws-cli-docker-demo/hello オブジェクト /aws をローカルファイルシステムにダウンロードします。hello オブジェクトをコンテナの /aws ディレクトリにダウンロードすると、ファイルはホストシステムの現在の作業ディレクトリにも保存されます。

Amazon ECR Public

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd)/aws public.ecr.aws/aws-cli/  
aws-cli s3 cp s3://aws-cli-docker-demo/hello .  
download: s3://aws-cli-docker-demo/hello to ./hello
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%\aws:/root/.aws -v %cd%:/aws public.ecr.aws/  
aws-cli/aws-cli s3 cp s3://aws-cli-docker-demo/hello .  
download: s3://aws-cli-docker-demo/hello to ./hello
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\aws:/root/.aws -v $pwd\aws:/aws  
public.ecr.aws/aws-cli/aws-cli s3 cp s3://aws-cli-docker-demo/hello .
```

Docker Hub

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws amazon/aws-cli s3 cp s3://  
aws-cli-docker-demo/hello .  
download: s3://aws-cli-docker-demo/hello to ./hello
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%\aws:/root/.aws -v %cd%:/aws amazon/aws-cli  
s3 cp s3://aws-cli-docker-demo/hello .  
download: s3://aws-cli-docker-demo/hello to ./hello
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\aws:/root/.aws -v $pwd\aws:/aws  
amazon/aws-cli s3 cp s3://aws-cli-docker-demo/hello .
```

ダウンロードしたファイルがローカルファイルシステムに存在することを確認するには、以下を実行します。

Linux および macOS

```
$ cat hello  
Hello from Docker!
```

Windows PowerShell

```
$ type hello
Hello from Docker!
```

例 3: AWS_PROFILE環境変数の使用

特定システムの環境変数は、`-e`フラグを使用して呼び出すことができます。使用したいそれぞれの環境変数を呼び出してください。この例では、ホスト認証情報、設定、および `AWS_PROFILE` `s3 ls` コマンドを実行して Amazon Simple Storage Service (Amazon S3) にバケットを一覧表示するときの環境変数。

Amazon ECR Public

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e AWS_PROFILE public.ecr.aws/aws-cli/
aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%\aws:/root/.aws -e AWS_PROFILE
public.ecr.aws/aws-cli/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\aws:/root/.aws -e AWS_PROFILE
public.ecr.aws/aws-cli/aws-cli s3 ls
```

Docker Hub

Linux および macOS

```
$ docker run --rm -it -v ~/.aws:/root/.aws -e AWS_PROFILE amazon/aws-cli s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows コマンドプロンプト

```
$ docker run --rm -it -v %userprofile%\aws:/root/.aws -e AWS_PROFILE amazon/aws-cli
s3 ls
2020-03-25 00:30:48 aws-cli-docker-demo
```

Windows PowerShell

```
C:\> docker run --rm -it -v $env:userprofile\aws:/root/.aws -e AWS_PROFILE amazon/
aws-cli s3 ls
```

docker run コマンドの短縮

docker run コマンドを短縮するには、オペレーティングシステムの機能を使用して Linux および macOS で [symbolic link](#) (シンボリックリンク) または [alias](#) を作成するか、Windows で [doskey](#) を作成することをお勧めします。aws エイリアスを設定するには、次のいずれかのコマンドを実行します。

- aws コマンドへの基本的なアクセスについては、以下を実行します。

Amazon ECR Public

Linux および macOS

```
$ alias aws='docker run --rm -it public.ecr.aws/aws-cli/aws-cli'
```

Windows コマンドプロンプト

```
C:\> doskey aws=docker run --rm -it public.ecr.aws/aws-cli/aws-cli $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it public.ecr.aws/aws-cli/aws-cli $args}
Set-Alias -Name aws -Value AWSCLI
```

Docker Hub

Linux および macOS

```
$ alias aws='docker run --rm -it amazon/aws-cli'
```

Windows コマンドプロンプト

```
C:\> doskey aws=docker run --rm -it amazon/aws-cli $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it amazon/aws-cli $args}  
Set-Alias -Name aws -Value AWSCLI
```

- aws コマンド使用時にホストファイルシステムおよび構成設定にアクセスする場合は、以下を実行します。

Amazon ECR Public

Linux および macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws  
public.ecr.aws/aws-cli/aws-cli'
```

Windows コマンドプロンプト

```
C:\> doskey aws=docker run --rm -it -v %userprofile%\aws:/root/.aws -v %cd%:/aws  
public.ecr.aws/aws-cli/aws-cli $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\aws:/root/.aws -v  
$pwd\aws:/aws public.ecr.aws/aws-cli/aws-cli $args}  
Set-Alias -Name aws -Value AWSCLI
```

Docker Hub

Linux および macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws amazon/aws-  
cli'
```

Windows コマンドプロンプト

```
C:\> doskey aws=docker run --rm -it -v %userprofile%\aws:/root/.aws -v %cd%:/aws
amazon/aws-cli $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\aws:/root/.aws -v
$pwd\aws:/aws amazon/aws-cli $args}
Set-Alias -Name aws -Value AWSCLI
```

- aws エイリアスで使用する特定のバージョンを割り当てるには、バージョンタグを追加します。

Amazon ECR Public

Linux および macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws
public.ecr.aws/aws-cli/aws-cli:2.0.6'
```

Windows コマンドプロンプト

```
C:\> doskey aws=docker run --rm -it -v %userprofile%\aws:/root/.aws -v %cd%:/aws
public.ecr.aws/aws-cli/aws-cli:2.0.6 $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\aws:/root/.aws -v
$pwd\aws:/aws public.ecr.aws/aws-cli/aws-cli:2.0.6 $args}
Set-Alias -Name aws -Value AWSCLI
```

Docker Hub

Linux および macOS

```
$ alias aws='docker run --rm -it -v ~/.aws:/root/.aws -v $(pwd):/aws amazon/aws-
cli:2.0.6'
```

Windows コマンドプロンプト


```
C:\> doskey aws=docker run --rm -it -v %userprofile%\aws:/root/.aws -v %cd%:/aws
amazon/aws-cli:2.0.6 $*
```

Windows PowerShell

```
C:\> Function AWSCLI {docker run --rm -it -v $env:userprofile\aws:/root/.aws -v
$pwd\aws:/aws amazon/aws-cli:2.0.6 $args}
Set-Alias -Name aws -Value AWSCLI
```

エイリアスを設定したら、バージョン AWS CLI 2 をコンテナ内からホストシステムにインストールされているかのように実行できます。

```
$ aws --version
aws-cli/2.17.20 Python/3.7.3 Linux/4.9.184-linuxkit botocore/2.4.5dev10
```

のセットアップ AWS CLI

このトピックでは、AWS Command Line Interface (AWS CLI) がとやり取りするために使用する基本設定をすばやく設定する方法について説明します AWS。これには、セキュリティ認証情報、デフォルトの出力形式、デフォルトの AWS リージョンが含まれます。

トピック

- [プログラムによるアクセス用の認証情報を収集する](#)
- [新しい設定と認証情報のセットアップ](#)
- [既存の設定と認証情報ファイルの使用](#)

プログラムによるアクセス用の認証情報を収集する

の AWS 外部とやり取りする場合は、プログラムによるアクセスが必要です AWS Management Console。認証と認証情報の説明については、次のいずれかのオプションを選択します。

[Authentication type] (認証タイプ)	目的	手順
IAM Identity Center ワークフォースユーザーの短期認証情報	<p>(推奨) IAM Identity Center ワークフォースユーザーには短期認証情報を使用します。</p> <p>セキュリティのベストプラクティスは、 を IAM Identity Center AWS Organizations で使用することです。短期認証情報を、組み込みの IAM Identity Center ディレクトリや Active Directory などのユーザーディレクトリと組み合わせます。</p>	<p>the section called “IAM Identity Center 認証”</p>
IAM ユーザー短期認証情報	<p>IAM ユーザーの短期認証情報を使用します。これは、長期認証情報よりも安全です。認証情報が侵害された場合、有効期限が切れる前に使用できる時間は限られています。</p>	<p>the section called “短期の認証情報”</p>
IAM Amazon EC2インスタンスの または IAM Identity Center ユーザー。	<p>Amazon EC2インスタンスメタデータを使用して、Amazon EC2インスタンスに割り当てられたロールを使用して一時的な認証情報をクエリします。</p>	<p>the section called “Amazon EC2インスタンスメタデータを の認証情報として使用する AWS CLI”</p>
アクセス許可のロールを引き受ける	<p>別の認証情報メソッドをペアにして、ユーザーへの一時的なアクセスのロールを受け AWS のサービスと、にアクセスできない可能性があります。</p>	<p>the section called “IAM ロール”</p>

[Authentication type] (認証タイプ)	目的	手順
IAM ユーザー長期認証情報	(推奨されません) 有効期限のない長期的な認証情報を使用します。	the section called “IAM ユーザー”
IAM または IAM Identity Center ワークフォースユーザーの外部ストレージ	(推奨されません) 別の認証情報メソッドをペアリングしますが、認証情報の値はの外部に保存します AWS CLI。この方法は、認証情報が保存されている外部の場所と同じくらい安全です。	the section called “外部認証情報”

新しい設定と認証情報のセットアップ

は、設定と認証情報を credentials および config ファイルのプロファイル (設定のコレクション) に AWS CLI 保存します。

すばやくセットアップするには、主に 2 つの方法があります。

- [AWS CLI コマンドを使用した設定](#)
- [認証情報と設定ファイルの手動編集](#)

以下の例では、各認証方法でサンプル値を使用しています。サンプル値を自分の値に置き換えてください。

AWS CLI コマンドを使用した設定

一般的には、任意のターミナルの `aws configure` または `aws configure sso` コマンドを使用するのが、AWS CLI のインストールをセットアップするための最も簡単な方法です。希望する認証情報方法に基づいて、は関連情報の入力 AWS CLI を求めます。デフォルトでは、このプロファイルの情報は、使用するプロファイルを明示的に指定しない AWS CLI コマンドを実行するときに使用されます。

credentials ファイルと config ファイルの詳細については、「[の設定と認証情報ファイルの設定 AWS CLI](#)」を参照してください。

IAM Identity Center (SSO)

この例では、`aws configure sso` ウィザード AWS IAM Identity Center を使用します。詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

```
$ aws configure sso
SSO session name (Recommended): my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

Attempting to automatically open the SSO authorization page in your default browser.

There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (111122223333)
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

CLI default client Region [None]: us-west-2
CLI default output format [None]: json
CLI profile name [123456789011_ReadOnly]: user1
```

IAM Identity Center (Legacy SSO)

この例では、`aws configure sso` ウィザード AWS IAM Identity Center を使用する従来の方法を示しています。レガシーを使用するには SSO、セッション名を空白のままにします。詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

```
$ aws configure sso
SSO session name (Recommended):
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

SSO authorization page has automatically been opened in your default browser.
Follow the instructions in the browser to complete this authorization request.

There are 2 AWS accounts available to you.
```



```
$ aws configure set output json
```

Amazon EC2 instance metadata credentials

この例では、ホストする Amazon EC2 インスタンスメタデータから取得した認証情報を使用します。このプロセス用のウィザードは存在しないため、各値の設定には `aws configure set` コマンドを使用します。詳細については、「[the section called “Amazon EC2 インスタンスメタデータを の認証情報として使用する AWS CLI”](#)」を参照してください。

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole  
$ aws configure set credential_source Ec2InstanceMetadata  
$ aws configure set region us-west-2  
$ aws configure set output json
```

Long-term credentials

Warning

セキュリティ上のリスクを回避するため、専用ソフトウェアの開発時や実際のデータの使用時に、認証に IAM ユーザーを使用しないでください。代わりに、[AWS IAM Identity Center](#) などの ID プロバイダーとのフェデレーションを使用してください。

この例は、AWS Identity and Access Management の長期の認証情報用です。詳細については、「[the section called “IAM ユーザー”](#)」を参照してください。

```
$ aws configure  
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE  
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY  
Default region name [None]: us-west-2  
Default output format [None]: json
```

認証方法と認証情報メソッドの詳細については、「[認証とアクセス認証情報](#)」を参照してください。

認証情報と設定ファイルの手動編集

情報をコピーして貼り付ける場合は、`config` ファイルと `credentials` ファイルを手動で編集することをお勧めします。希望する認証情報メソッドに応じて、ファイルは異なる方法で設定されます。

ファイルはホームディレクトリの `.aws` フォルダの下に保存されます。ホームディレクトリの場所はオペレーティングシステムによって異なりますが、環境変数 `%UserProfile%` (Windows の場合) および `$HOME` または `チルド ~` (Unix ベースのシステムの場合) を使用して参照されます。これらの設定が保存される場所の詳細については、「[the section called “構成設定はどこに保存されていますか?”](#)」を参照してください。

次の例は、`default` プロファイルと `user1` という名前のプロファイルを示しており、サンプル値を使用しています。サンプル値を自分の値に置き換えてください。`credentials` ファイルと `config` ファイルの詳細については、「[の設定と認証情報ファイルの設定 AWS CLI](#)」を参照してください。

IAM Identity Center (SSO)

この例は用です AWS IAM Identity Center。詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

認証情報ファイル

`credentials` ファイルは、この認証方法には使用しません。

設定ファイル

```
[default]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = readOnly
region = us-west-2
output = text

[profile user1]
sso_session = my-sso
sso_account_id = 444455556666
sso_role_name = readOnly
region = us-east-1
output = json

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```



```
output=text
```

Amazon EC2 instance metadata credentials

この例では、ホストする Amazon EC2 インスタンスメタデータから取得した認証情報を使用します。詳細については、「[the section called “Amazon EC2 インスタンスメタデータを の認証情報として使用する AWS CLI”](#)」を参照してください。

認証情報ファイル

credentials ファイルは、この認証方法には使用しません。

設定ファイル

```
[default]
role_arn=arn:aws:iam::123456789012:role/defaultrole
credential_source=Ec2InstanceMetadata
region=us-west-2
output=json

[profile user1]
role_arn=arn:aws:iam::777788889999:role/user1role
credential_source=Ec2InstanceMetadata
region=us-east-1
output=text
```

Long-term credentials

Warning

セキュリティ上のリスクを回避するため、専用ソフトウェアの開発時や実際のデータの使用時に、認証に IAM ユーザーを使用しないでください。代わりに、[AWS IAM Identity Center](#) などの ID プロバイダーとのフェデレーションを使用してください。

この例は、AWS Identity and Access Management の長期の認証情報用です。詳細については、「[the section called “IAM ユーザー”](#)」を参照してください。

認証情報ファイル

```
[default]
```

```
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

[user1]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

設定ファイル

```
[default]
region=us-west-2
output=json

[profile user1]
region=us-east-1
output=text
```

認証方法と認証情報メソッドの詳細については、「[認証とアクセス認証情報](#)」を参照してください。

既存の設定と認証情報ファイルの使用

既存の設定および認証情報ファイルがある場合は、AWS CLIにこれらのファイルを使用できます。

config ファイルと credentials ファイルを使用するには、これらのファイルをホームディレクトリの .aws という名前の付いたフォルダに移動します。ホームディレクトリの場所はオペレーティングシステムによって異なりますが、環境変数 %UserProfile% (Windows の場合) および \$HOME またはチルド ~ (Unix ベースのシステムの場合) を使用して参照されます。

AWS_CONFIG_FILE と AWS_SHARED_CREDENTIALS_FILE の環境変数を別のローカルパスに設定することで、config ファイルと credentials ファイルの場所をデフォルト以外の場所に指定できます。詳細については、「[の環境変数の設定 AWS CLI](#)」を参照してください。

設定と認証情報ファイルの設定の詳細については、「[the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)」を参照してください。

の設定 AWS CLI

このセクションでは、AWS Command Line Interface (AWS CLI) がとやり取りするために使用する設定を設定する方法について説明します AWS。これには以下が含まれます。

- 認証情報は、 を呼び出すユーザーを識別しますAPI。アクセス認証情報は、AWS サーバーへのリクエストを暗号化して ID を確認し、関連するアクセス許可ポリシーを取得するために使用されます。これらのアクセス許可によって、実行できるアクションが決まります。認証情報の設定の詳細については、「[認証とアクセス認証情報](#)」を参照してください。
- デフォルトの出力形式やデフォルトの AWS リージョンなど、リクエストの処理 AWS CLI 方法を示すその他の設定の詳細。

Note

AWS では、すべての受信リクエストが暗号化で署名されている必要があります。AWS CLI はこれをユーザーに代わって行います。「signature」には、AWS サービスによってdate/time stamp. Therefore, you must ensure that your computer's date and time are set correctly. If you don't, and the date/time in the signature is too far off of the date/time認識された が含まれ、AWS リクエストを拒否します。

設定と認証情報の優先順位

認証情報と設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。特定の場所が他の場所よりも優先されます。AWS CLI 認証情報と設定は、次の順序で優先されます。

1. [コマンドラインオプション](#) - `---region`、`--output`、`--profile` パラメータなど、他の任意の場所にある設定を上書きします。
2. [環境変数](#) - システムの環境変数に値を保存できます。
3. [ロールを引き受ける](#) - 設定または [aws sts assume-role](#) コマンドを使用してIAMロールのアクセス許可を引き受けます。
4. [ウェブ ID でロールを引き受ける](#) - 設定または [aws sts assume-role](#) コマンドを使用して、ウェブ ID を使用してIAMロールのアクセス許可を引き受けます。

5. [AWS IAM Identity Center](#) – config ファイルに保存されている IAM Identity Center 設定は、aws configure sso コマンドの実行時に更新されます。その後、aws sso login コマンドを実行すると認証情報が認証されます。「config」ファイルは、Linux または macOS では「~/.aws/config」、Windows では「C:\Users**USERNAME**\.aws\config」にあります。
6. [認証情報ファイル](#) – コマンド aws configure を実行すると、credentials ファイルと config ファイルが更新されます。「credentials」ファイルは、Linux または macOS では「~/.aws/credentials」、Windows では「C:\Users**USERNAME**\.aws\credentials」にあります。
7. [カスタムプロセス](#) – 外部ソースから認証情報を取得します。
8. [設定ファイル](#) - コマンド aws configure を実行すると、credentials ファイルと config ファイルが更新されます。「config」ファイルは、Linux または macOS では「~/.aws/config」、Windows では「C:\Users**USERNAME**\.aws\config」にあります。
9. [コンテナ認証情報](#) – IAM ロールを各 Amazon Elastic Container Service (Amazon ECS) タスク定義に関連付けることができます。関連付けられると、そのロールの一時認証情報は、そのタスクのコンテナで使用できるようになります。詳細については、[IAM 「Amazon Elastic Container Service デベロッパーガイド」の「タスクのロール」](#)を参照してください。
10. [Amazon EC2 インスタンスプロファイルの認証情報](#) – IAM ロールを Amazon Elastic Compute Cloud (Amazon EC2) の各インスタンスに関連付けることができます。関連付けられると、そのロールの一時認証情報は、インスタンスで実行中のコードで使用できるようになります。認証情報は Amazon EC2 メタデータサービスを介して配信されます。詳細については、[IAM 「Amazon ユーザーガイド」の「Amazon のロール EC2」](#)と IAM 「ユーザーガイド」の「[インスタンスプロファイルの使用](#)」を参照してください。 EC2

このセクションのその他のトピック

- [the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)
- [the section called “環境可変”](#)
- [the section called “のコマンドラインオプション AWS CLI”](#)
- [the section called “でのコマンド完了の設定 AWS CLI”](#)
- [the section called “再試行”](#)
- [the section called “のHTTPプロキシの使用 AWS CLI”](#)

の設定と認証情報ファイルの設定 AWS CLI

頻繁に利用される構成設定および認証情報を AWS CLI が維持するファイルに保存することができます。

ファイルは profiles に分割されます。デフォルトでは、`default` という名前のプロファイルにある設定 AWS CLI を使用します。替わりの設定を使用するには、追加のプロファイルを作成して参照できます。

サポートされた環境変数のいずれかの設定を使用するか、あるいはコマンドラインパラメータを使用して、個別の設定を上書きすることもできます。構成設定の優先順位の詳細については、「[の設定 AWS CLI](#)」を参照してください。

Note

認証情報の設定の詳細については、「[認証とアクセス認証情報](#)」を参照してください。

トピック

- [設定ファイルと認証ファイルの形式](#)
- [構成設定はどこに保存されていますか?](#)
- [名前を指定されたプロファイルを使用する](#)
- [コマンドを使用した構成設定の設定と表示](#)
- [新しい設定と認証情報を設定するコマンドの例](#)
- [サポートされる config ファイル設定](#)

設定ファイルと認証ファイルの形式

config ファイルと credentials ファイルは、セクションにまとめられています。セクションには、プロファイル、SSO セッション、およびサービスが含まれます。セクションは、設定の名前付きコレクションであり、別のセクション定義の行が検出されるまで続きます。複数のプロファイルとセクションを config ファイルおよび credentials ファイルに保存できます。

これらのファイルは、次の形式を使用するプレーンテキストファイルです。

- セクション名は、`[default]`、`[profile user1]`、`[sso-session]` などの括弧 `[]` で囲まれています。

- セクション内のすべてのエントリは、`setting_name=value` の一般的な形式になります。
- 行の先頭にハッシュ文字 (#) を付けると、行をコメントアウトできます。

`config` および `credentials` ファイルには以下のセクションタイプが含まれます。

- [セクションタイプ: profile](#)
- [セクションタイプ: sso-session](#)
- [セクションタイプ: services](#)

セクションタイプ: **profile**

AWS CLI ストア

ファイルに応じて、セクション名の形式は次のとおりです。

- 設定ファイル: `[default] [profile user1]`
- 認証情報ファイル: `[default] [user1]`

`credentials` ファイルにエントリを作成するときは、`profile` という単語を使用しないでください。

各プロファイルは異なる認証情報を指定できます。また、別の AWS リージョンおよび出力形式を指定することもできます。`config` ファイルでプロファイルに名前を付けるときは、プレフィックス「`profile`」を含めますが、`credentials` ファイルには含めないでください。

次の例は、2つのプロファイル、リージョン、出力が指定されている `credentials` ファイルと `config` ファイルを示しています。最初の `[default]` はプロファイルを指定せずに AWS CLI コマンドを実行するときに使用します。2番目の `--profile user1` パラメータで AWS CLI コマンドを実行する場合に使用されます。

IAM Identity Center (SSO)

この例は用です AWS IAM Identity Center。詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

認証情報ファイル

`credentials` ファイルは、この認証方法には使用しません。

設定ファイル

```
[default]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = readOnly
region = us-west-2
output = text

[profile user1]
sso_session = my-sso
sso_account_id = 444455556666
sso_role_name = readOnly
region = us-east-1
output = json

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

IAM Identity Center (Legacy SSO)

この例では、のレガシーメソッド用です AWS IAM Identity Center。詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

認証情報ファイル

credentials ファイルは、この認証方法には使用しません。

設定ファイル

```
[default]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
sso_account_id = 111122223333
sso_role_name = readOnly
region = us-west-2
output = text

[profile user1]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-east-1
```



```
role_arn=arn:aws:iam::777788889999:role/user1role
credential_source=Ec2InstanceMetadata
region=us-east-1
output=text
```

Long-term credentials

Warning

セキュリティ上のリスクを回避するため、専用ソフトウェアの開発時や実際のデータの使用時に、認証にIAMユーザーを使用しないでください。代わりに、[AWS IAM Identity Center](#) などの ID プロバイダーとのフェデレーションを使用してください。

この例は、AWS Identity and Access Managementの長期の認証情報用です。詳細については、「[the section called “IAM ユーザー”](#)」を参照してください。

認証情報ファイル

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

[user1]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

設定ファイル

```
[default]
region=us-west-2
output=json

[profile user1]
region=us-east-1
output=text
```

詳細および追加の認可/認証情報メソッドについては、「[the section called “IAM ユーザー”](#)」を参照してください。

セクションタイプ: **sso-session**

config ファイルの sso-session セクションは、SSO アクセストークンを取得するための設定変数をグループ化するために使用されます。これにより、AWS 認証情報を取得できます。次の設定を使用します。

- (必須) [sso_start_url](#)
- (必須) [sso_region](#)
- [sso_account_id](#)
- [sso_role_name](#)
- [sso_registration_scopes](#)

sso-session セクションを定義してプロファイルに関連付けます。sso_region と sso_start_url は sso-session セクション内に設定する必要があります。通常、が SSO 認証情報を SDK リクエストできるように、profile セクションで sso_account_id とを設定 sso_role_name する必要があります。

次の例では、SSO 認証情報をリクエスト SDK するようにを設定し、トークンの自動更新をサポートしています。

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

これにより、sso-session 設定を複数のプロファイルでも再利用できるようになります。

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
```

```
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

ただし、SSOトークン設定のすべてのシナリオで `sso_account_id` と `sso_role_name` が必要なわけではありません。アプリケーションでベアラー認証をサポートする AWS のサービスのみを使用する場合、従来の AWS 認証情報は必要ありません。ベアラー認証は、ベアラートークンと呼ばれるセキュリティトークンを使用する HTTP 認証スキームです。このシナリオでは、`sso_account_id` と `sso_role_name` は必須ではありません。AWS のサービスでベアラートークン認証をサポートしているかどうかについては、各サービスのガイドを参照してください。

さらに、登録スコープは `sso-session` の一部として設定できます。スコープは、アプリケーションのユーザーアカウントへのアクセスを制限するための OAuth 2.0 のメカニズムです。アプリケーションは 1 つ以上のスコープをリクエストでき、アプリケーションに発行されたアクセストークンは付与されたスコープに限定されます。これらのスコープは、登録された OIDC クライアントおよびクライアントによって取得されたアクセストークンに対して承認されるようにリクエストされたアクセス許可を定義します。次の例では、アカウント/ロールを一覧表示するアクセスを許可するように `sso_registration_scopes` を設定しています。

```
[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

認証トークンは、セッション名に基づいたファイル名を使用して、`~/.aws/sso/cache` ディレクトリの下のディスクにキャッシュされます。

設定タイプの詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

セクションタイプ: **services**

`services` セクションは、AWS のサービス リクエストのカスタムエンドポイントを設定する設定のグループです。その後、プロファイルは `services` セクションにリンクされます。

```
[profile dev]
```

```
services = my-services
```

services セクションは <SERVICE> = 行ごとにサブセクションに分割されます。ここで、<SERVICE> は AWS のサービス 識別子キーです。AWS のサービス 識別子は、すべてのスペースをアンダースコアに置き換え、すべての文字を小文字serviceIdに置き換えることでAPI、モデルのに基づいています。services セクションで使用するすべてのサービス識別子キーのリストについては、「[でのエンドポイントの使用 AWS CLI](#)」を参照してください。サービス識別子キーの後には、ネストされた設定 (それぞれが 1 行にあり、2 つのスペースでインデントされている) が続きます。

次の例では、Amazon DynamoDB のサービスに対して行われたリクエストに使用するエンドポイントを設定します。*my-services* で使用される セクション *dev* プロファイル。直後のインデントされた行はすべてそのサブセクションに含まれ、そのサービスに適用されます。

```
[profile dev]  
services = my-services  
  
[services my-services]  
dynamodb =  
    endpoint_url = http://localhost:8000
```

サービス固有のエンドポイントの詳細については、「[でのエンドポイントの使用 AWS CLI](#)」を参照してください。

プロファイルにロールを引きIAM受ける機能のために source_profileパラメータを介してロールベースの認証情報が設定されている場合、は指定されたプロファイルのサービス設定SDKのみを使用します。ロールチェーンされたプロファイルは使用されません。例えば、次の共有 config ファイルを使用します。

```
[profile A]  
credential_source = Ec2InstanceMetadata  
endpoint_url = https://profile-a-endpoint.aws/  
  
[profile B]  
source_profile = A  
role_arn = arn:aws:iam::123456789012:role/roleB  
services = profileB  
  
[services profileB]  
ec2 =
```

```
endpoint_url = https://profile-b-ec2-endpoint.aws
```

プロファイルを使用してコードで Amazon を呼び出す B と EC2、エンドポイントは として解決されます `https://profile-b-ec2-endpoint.aws`。コードが他のサービスにリクエストを送信した場合、エンドポイントの解決はカスタムロジックには従いません。エンドポイントはプロファイル A で定義されたグローバルエンドポイントには解決されません。グローバルエンドポイントを B プロファイルに対して有効にするには、プロファイル B 内で直接 `endpoint_url` を設定する必要があります。

構成設定はどこに保存されていますか？

は、指定した機密認証情報を `credentials` という名前 `aws configure` のローカルファイル、ホームディレクトリの という名前のフォルダ `.aws` に AWS CLI 保存します。 `aws configure` で指定された機密性の低い設定オプションは、 `config` という名前のローカルファイルに保存されるだけでなく、ホームディレクトリの `.aws` フォルダにも保存されます。

認証情報を設定ファイルに保存する

はファイルから認証情報を読み取ることができるため AWS CLI、すべてのプロファイル設定を 1 つの `config` ファイルに保持できます。同じ名前を共有するプロファイルの両方のファイルに認証情報がある場合、認証情報ファイルのキーが優先されます。認証情報は `credentials` ファイルに保存することをお勧めします。これらのファイルは、さまざまな言語ソフトウェア開発キット () でも使用されます SDKs。SDKs に加えてのいずれかを使用する場合は AWS CLI、認証情報を独自のファイルに保存する必要があるかどうかを確認します。

ホームディレクトリの場所はオペレーティングシステムによって異なりますが、環境変数 `%UserProfile%` (Windows の場合) および `$HOME` または `チルド ~` (Unix ベースのシステムの場合) を使用して参照されます。 `AWS_CONFIG_FILE` と `AWS_SHARED_CREDENTIALS_FILE` の環境変数を別のローカルパスに設定することで、ファイルの場所をデフォルト以外の場所に指定できます。詳細については、「[の環境変数の設定 AWS CLI](#)」を参照してください。

AWS Identity and Access Management (IAM) ロールを指定する共有プロファイルを使用する場合、は オペレーションを AWS CLI `AWS STS AssumeRole` 呼び出して一時的な認証情報を取得します。その後、これらの認証情報は `~/ .aws/cli/cache` に保存されます。後続の AWS CLI コマンドは、キャッシュされた一時的な認証情報を有効期限が切れるまで使用し、その時点では AWS CLI 認証情報を自動的に更新します。

名前を指定されたプロファイルを使用する

明示的に定義されたプロファイルがない場合は、default プロファイルが使用されます。

名前付きプロファイルを使用するには、コマンドに `--profile profile-name` オプションを追加します。次の例では、user1プロファイルで定義された認証情報と設定を使用して、すべての Amazon EC2 インスタンスを一覧表示します。

```
$ aws ec2 describe-instances --profile user1
```

複数のコマンドで名前付きプロファイルを使用する場合は、コマンドごとにプロファイルを指定しなくても済むように、AWS_PROFILE 環境変数をデフォルトプロファイルとして設定できます。この設定は、`--profile` パラメータを使用して上書きできます。

Linux or macOS

```
$ export AWS_PROFILE=user1
```

Windows

```
C:\> setx AWS_PROFILE user1
```

[set](#) を使用して環境変数を設定すると、現在のコマンドプロンプトセッションの終了時まで、または変数を別の値に設定するまで、使用する値が変更されます。

[setx](#) を使用して環境変数を設定すると、コマンドの実行後に作成するすべてのコマンドシェルの値が変更されます。コマンドの実行時に既に実行されているコマンドシェルには影響しません。変更の影響を確認するには、コマンドシェルを閉じて再起動します。

環境変数を設定すると、シェルセッションの終了時まで、または変数に別の値を設定するまで、デフォルトのプロファイルが変更されます。環境変数をシェルのスタートアップスクリプトに配置すると、今後のセッションでこの環境変数を永続的にできます。詳細については、「[の環境変数の設定 AWS CLI](#)」を参照してください。

コマンドを使用した構成設定の設定と表示

コマンドを使用して構成設定を表示および設定するには、複数の方法があります。

[aws configure](#)

このコマンドを実行すると、認証情報、リージョン、出力形式をすばやく設定および表示できます。次の例は、サンプル値を示しています。

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

[aws configure set](#)

`aws configure set` を使用して、任意の認証情報または構成設定を設定できます。--profile 設定で表示または変更するプロファイルを指定します。

例えば、次のコマンドは `region` という名前のプロファイル内の `integ` を設定します。

```
$ aws configure set region us-west-2 --profile integ
```

設定を削除するには、テキストエディタの `config` および `credentials` ファイルの設定を手動で削除します。

[aws configure get](#)

`aws configure get` を使用して設定した認証情報または構成設定を取得できます。--profile 設定で表示または変更するプロファイルを指定します。

例えば、次のコマンドは `region` という名前のプロファイル内の `integ` 設定を取得します。

```
$ aws configure get region --profile integ
us-west-2
```

出力が空の場合にはその設定は明示的に構成されず、デフォルトの値が使用されます。

[aws configure import](#)

IAM ウェブコンソールから生成された CSV 認証情報をインポートします。これは IAM Identity Center から生成された認証情報用ではありません。IAM Identity Center を使用するお客様は、`aws configure sso` を使用する必要があります。CSV ファイルは、ユーザー名に一致するプロファイル名でインポートされます。CSV ファイルには次のヘッダーが含まれている必要があります。

- ユーザー名
- アクセスキー ID
- シークレットアクセスキー

Note

最初のキーペアの作成時に、[Download .csv file] (.csv ファイルをダウンロードする) ダイアログボックスを閉じると、ダイアログボックスを閉じた後に、シークレットアクセスキーにアクセスできません。.csv ファイルが必要な場合は、必要なヘッダーと保存したキーペア情報を使用して自分で作成する必要があります。キーペア情報にアクセスできない場合は、新しいキーペアを作成する必要があります。

```
$ aws configure import --csv file://credentials.csv
```

aws configure list

設定データを一覧表示するには、aws configure list コマンドを使用します。このコマンドは、指定したプロファイルに使用されるプロファイル、アクセスキー、シークレットキー、およびリージョンの設定情報を一覧表示します。各設定項目について、値、設定値の取得元、および設定の変数名が表示されます。

例えば、環境変数 AWS リージョン に を指定すると、このコマンドは、設定したリージョンの名前、この値が環境変数から取得されたこと、および環境変数の名前を表示します。

ロールや IAM Identity Center などの一時的な認証情報メソッドの場合、このコマンドは一時的にキャッシュされたアクセスキーを表示し、シークレットアクセスキーが表示されます。

```
$ aws configure list
      Name                Value                Type    Location
      ----                -
      profile              <not set>           None    None
      access_key           *****ABCD        shared-credentials-file
      secret_key           *****ABCD        shared-credentials-file
      region                us-west-2           env     AWS_DEFAULT_REGION
```

aws configure list-profiles

すべてのプロファイル名を一覧表示するには、aws configure list-profiles コマンドを使用します。

```
$ aws configure list-profiles
default
test
```

aws configure sso

このコマンドを実行して、AWS IAM Identity Center 認証情報、リージョン、出力形式をすばやく設定して表示します。次の例は、サンプル値を示しています。

```
$ aws configure sso
SSO session name (Recommended): my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

aws configure sso-session

このコマンドを実行して、および config ファイルのセッションセクションで AWS IAM Identity Center 認証情報、リージョン、出力形式をすばやく設定 credentials して表示します。次の例は、サンプル値を示しています。

```
$ aws configure sso-session
SSO session name: my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

新しい設定と認証情報を設定するコマンドの例

次の例は、さまざまな認証方法で指定された認証情報、リージョン、出力を使用してデフォルトプロファイルを設定する方法を示しています。

IAM Identity Center (SSO)

この例では、aws configure sso ウィザード AWS IAM Identity Center を使用します。詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

```
$ aws configure sso
SSO session name (Recommended): my-sso
```

```
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

Attempting to automatically open the SSO authorization page in your default browser.

There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (111122223333)
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
  FullAccess

Using the role name "ReadOnly"

CLI default client Region [None]: us-west-2
CLI default output format [None]: json
CLI profile name [123456789011_ReadOnly]: user1
```

IAM Identity Center (Legacy SSO)

この例では、aws configure ssoウィザード AWS IAM Identity Center を使用する従来の方法を示しています。レガシーを使用するにはSSO、セッション名を空白のままにします。詳細については、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

```
$ aws configure sso
SSO session name (Recommended):
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1

SSO authorization page has automatically been opened in your default browser.
Follow the instructions in the browser to complete this authorization request.

There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (111122223333)
  ProductionAccount, production-account-admin@example.com (444455556666)

Using the account ID 111122223333

There are 2 roles available to you.
> ReadOnly
```


マンドを使用します。詳細については、「[the section called “Amazon EC2インスタンスメタデータを の認証情報として使用する AWS CLI”](#)」を参照してください。

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole
$ aws configure set credential_source Ec2InstanceMetadata
$ aws configure set region us-west-2
$ aws configure set output json
```

Long-term credentials

Warning

セキュリティ上のリスクを回避するため、専用ソフトウェアの開発時や実際のデータの使用時に、認証にIAMユーザーを使用しないでください。代わりに、[AWS IAM Identity Center](#) などの ID プロバイダーとのフェデレーションを使用してください。

この例は、AWS Identity and Access Managementの長期の認証情報用です。詳細については、「[the section called “IAM ユーザー”](#)」を参照してください。

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

サポートされる config ファイル設定

トピック

- [\[Global settings \(グローバル設定\)\]](#)
- [S3 カスタマーコマンド設定](#)

config ファイルでは、以下の設定がサポートされます。同名の環境変数があること、または同名のコマンドラインオプションによって上書きされる場合を除き、指定された (またはデフォルトの) プロファイルにリストされる値が使用されます。優先する順序設定の詳細については、「[の設定 AWS CLI](#)」を参照してください。

[Global settings (グローバル設定)]

aws_access_key_id

コマンドリクエストを認証するための認証情報の一部として使用される AWS アクセスキーを指定します。これを config ファイルに保存することもできますが、credentials ファイルに保存することが推奨されます。

AWS_ACCESS_KEY_ID 環境変数で上書きすることができます。アクセスキー ID をコマンドラインオプションとして指定することはできません。

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

aws_secret_access_key

コマンドリクエストを認証するための認証情報の一部として使用される AWS シークレットキーを指定します。これを config ファイルに保存することもできますが、credentials ファイルに保存することが推奨されます。

AWS_SECRET_ACCESS_KEY 環境変数で上書きすることができます。シークレットアクセスキーをコマンドラインオプションとして指定することはできません。

```
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

aws_session_token

AWS セッショントークンを指定します。セッショントークンは、一時的なセキュリティ認証情報を手動で指定する場合にのみ必要です。これを config ファイルに保存することもできますが、credentials ファイルに保存することが推奨されます。

AWS_SESSION_TOKEN 環境変数で上書きすることができます。セッショントークンをコマンドラインオプションとして指定することはできません。

```
aws_session_token = AqoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT  
+FvwnKwRc0IfrrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/  
IvU1dYUg2RVAJBanLiHb4IgrmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

ca_bundle

証明書の検証に使用される CA SSL証明書バンドル (.pem 拡張子を持つファイル) を指定します。

[AWS_CA_BUNDLE](#) 環境変数あるいは [--ca-bundle](#) コマンドラインオプションで上書きできます。

```
ca_bundle = dev/apps/ca-certs/cabundle-2019mar05.pem
```

cli_auto_prompt

AWS CLI バージョン 2 の自動プロンプトを有効にします。使用できる設定は 2 つあります。

- on は、ユーザーが aws コマンドを実行しようとするたびに、完全な自動プロンプトモードを使用します。これには、完全なコマンドまたは不完全なコマンドの両方の ENTER 後に を押すことが含まれます。

```
cli_auto_prompt = on
```

- on-partial は、部分的な自動プロンプトモードを使用します。コマンドが不完全、またはクライアント側の検証エラーが原因でコマンドを実行できない場合は、自動プロンプトが使用されます。このモードは、既存のスクリプトまたはランブックがある場合、あるいはすべてのコマンドに対してプロンプトを表示するのではなく、不慣れなコマンドにのみ自動プロンプトを表示したい場合に特に便利です。

```
cli_auto_prompt = on-partial
```

この設定は、[aws_cli_auto_prompt](#) 環境変数を使用するか、[--cli-auto-prompt](#) および [--no-cli-auto-prompt](#) コマンドラインパラメータを使用して上書きできます。

AWS CLI バージョン 2 自動プロンプト機能の詳細については、「」を参照してください [でのコマンドプロンプトの有効化と使用 AWS CLI](#)。

cli_binary_format

AWS CLI バージョン 2 がバイナリ入力パラメータをどのように解釈するかを指定します。これには、次のいずれかの値を指定できます。

- base64 - これはデフォルト値です。バイナリラージオブジェクト (BLOB) として入力される入力パラメータは、base64 でエンコードされた文字列を受け入れます。真のバイナリコンテンツ

ツを渡すには、コンテンツをファイルに置き、ファイルのパスと名前をパラメータの値として `fileb://` プレフィックスを付けて指定します。ファイルに含まれる base64 エンコードされたテキストを渡すには、ファイルのパスと名前をパラメータの値として `file://` プレフィックスを付けて指定します。

- `raw-in-base64-out` – AWS CLI バージョン 1 のデフォルト。設定の値が `raw-in-base64-out` の場合、`file://` プレフィックスを使用して参照されるファイルはテキストとして読み取られます。AWS CLI は、これをバイナリにエンコードしようとします。

このエントリには、同等の環境変数はありません。 `--cli-binary-format raw-in-base64-out` パラメータを使用すると、1 つのコマンドで値を指定できます。

```
cli_binary_format = raw-in-base64-out
```

`fileb://` プレフィックス表記を使用してファイル内のバイナリ値を参照する場合、はファイルに raw バイナリコンテンツが含まれていることを AWS CLI 常に想定し、値を変換しようとしません。

`file://` プレフィックス表記を使用してファイル内のバイナリ値を参照する場合、は現在の `cli_binary_format` 設定に従ってファイルを AWS CLI 処理します。この設定の値が base64 (明示的に設定されていない場合はデフォルト) の場合、は、ファイルに base64 でエンコードされたテキストが含まれていることを AWS CLI 期待します。この設定の値が の場合 `raw-in-base64-out`、はファイルに raw バイナリコンテンツが含まれていることを AWS CLI 期待します。

cli_history

デフォルトでは無効になっています。この設定により、AWS CLI のコマンド履歴が有効になります。この設定を有効にすると、は `aws` コマンドの履歴 AWS CLI を記録します。

```
cli_history = enabled
```

`aws history list` コマンドを使用すると履歴を一覧表示でき、そこから得られた `command_ids` を `aws history show` コマンドに使用すると詳細を確認できます。詳細については、AWS CLI リファレンスガイドの「[aws history](#)」を参照してください。

cli_pager

出力に使用されるページャープログラムを指定します。デフォルトでは、AWS CLI バージョン 2 はオペレーティングシステムのデフォルトのページャープログラムを通じてすべての出力を返します。

AWS_PAGER 環境変数によって上書きできます。

```
cli_pager=less
```

cli_timestamp_format

出力に含まれるタイムスタンプの形式を指定します。次の値のいずれかを指定できます。

- iso8601 – AWS CLI バージョン 2 のデフォルト値。指定した場合、は [ISO8601](#) に従ってすべてのタイムスタンプ AWS CLI を再フォーマットします。

ISO 8601 形式のタイムスタンプは次の例のようになります。最初の例は、Z 後続の を含めることで [協定世界時 \(UTC\)](#) の時間を示しています。日付と時刻は T で区切られます。

```
2019-10-31T22:21:41Z
```

別のタイムゾーンを指定するには、 の代わりに Z+または を指定し UTC、目的のタイムゾーンが の前または後ろの時間数を 2 桁の値として指定します。次の例は、前の例と同じ時刻ですが、 から 8 時間遅れている太平洋標準時に調整されています UTC。

```
2019-10-31T14:21:41-08
```

- wire – AWS CLI バージョン 1 のデフォルト値。指定した場合、は HTTP、クエリレスポンスで受信したとおりにすべてのタイムスタンプ値 AWS CLI を表示します。

このエントリには同等の環境変数あるいはコマンドラインオプションがありません。

```
cli_timestamp_format = iso8601
```

credential_process

がこのコマンドに使用する認証情報を生成または取得するために が AWS CLI 実行する外部コマンドを指定します。このコマンドは、特定の形式で認証情報を返す必要があります。この設定を使用する方法の詳細については、「[の外部プロセスを使用して認証情報を調達する AWS CLI](#)」を参照してください。

このエントリには同等の環境変数あるいはコマンドラインオプションがありません。

```
credential_process = /opt/bin/awscreds-retriever --username susan
```

credential_source

Amazon EC2インスタンスまたはコンテナ内で使用され、が AWS CLI `role_arn`パラメータで指定したロールを引き受けるために使用する認証情報をどこで見つけることができるかを指定します。 `source_profile` と `credential_source` の両方を同じプロファイルで指定することはできません。

このパラメータには、次の 3 つの値のいずれかを指定できます。

- 環境 – AWS CLI 環境変数からソース認証情報を取得する を指定します。
- `Ec2InstanceMetadata` – AWS CLI が [EC2インスタンスプロファイル](#) にアタッチされたIAMロールを使用してソース認証情報を取得するように指定します。
- `EcsContainer` – AWS CLI コンテナにアタッチされたIAMロールをソース認証情報ECSとして使用することを に指定します。

```
credential_source = Ec2InstanceMetadata
```

duration_seconds

ロールセッションの最大期間を秒単位で指定します。この値は 900 秒 (15 分) からロールの最大セッション期間設定 (上限は 43200) までの範囲を指定できます。これはオプションのパラメータであり、デフォルトでは 3600 秒に設定されています。

endpoint_url

すべてのサービスリクエストに使用されるエンドポイントを指定します。この設定が `config` ファイルの [services](#) セクションで使用されている場合、エンドポイントは指定されたサービスにのみ使用されます。

次の例では、Amazon S3 のグローバルエンドポイント `http://localhost:1234` とサービス固有のエンドポイント `http://localhost:4567` を使用しています。

```
[profile dev]
endpoint_url = http://localhost:1234
services = s3-specific

[services s3-specific]
s3 =
    endpoint_url = http://localhost:4567
```

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。
4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

ignore_configure_endpoint_urls

有効にすると、は config ファイルで指定されたすべてのカスタムエンドポイント設定 AWS CLI を無視します。有効な値は、**true** および **false** です。

```
ignore_configure_endpoint_urls = true
```

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。

4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

[external_id](#)

お客様のアカウントでサードパーティーがロールを引き受けるために使用される独自の識別子を指定します。これは ExternalId オペレーションの AssumeRole パラメータにマップします。このパラメータは、ロールの信頼ポリシーで ExternalId の値が指定されている場合にのみ必要です。詳細については、IAM「[ユーザーガイド](#)」の AWS「[リソースへのアクセスを第三者に付与するとき外部 ID を使用する方法](#)」を参照してください。

[max_attempts](#)

AWS CLI 再試行ハンドラーが使用する最大再試行回数の値を指定します。最初の呼び出しは、指定した max_attempts 値にカウントされます。

環境変数 AWS_MAX_ATTEMPTS を使用して、この値を上書きできます。

```
max_attempts = 3
```

[mfa_serial](#)

ロールを引き受けるときに使用する MFA デバイスの識別番号。これは、引き受けるロールの信頼ポリシーに MFA 認証を必要とする条件が含まれている場合にのみ必須です。値は、ハードウェアデバイスのシリアル番号 (など GAHT12345678) または仮想 MFA デバイス (など) の Amazon リソースネーム (ARN) のいずれかです `arn:aws:iam::123456789012:mfa/user`。

output

このプロファイルを使用してリクエストするコマンドのデフォルトの出力形式を指定します。次の値のいずれかを指定できます。

- [json](#) – 出力は [JSON](#) 文字列としてフォーマットされます。

- [yaml](#) – 出力は [YAML](#) 文字列としてフォーマットされます。
- [yaml-stream](#) – 出力はストリーミングされ、[YAML](#) 文字列としてフォーマットされます。ストリーミングにより、大きなデータタイプの処理を高速化できます。
- [text](#) - 出力は、複数行のタブ区切りの文字列値としてフォーマットされます。これは、grep、sed、または awk などのテキストプロセッサに出力を渡すのに役立ちます。
- [table](#) - 出力は、テーブルとしてフォーマットされ、文字の「+|-」を使用してセルの境界を形成します。通常、情報は他の形式よりも読みやすい「わかりやすい」形式で表示されますが、プログラムとしては役立ちません。

AWS_DEFAULT_OUTPUT 環境変数あるいは --output コマンドラインオプションで上書きできません。

```
output = table
```

parameter_validation

AWS CLI クライアントが AWS サービスエンドポイントに送信する前にパラメータを検証しようとするかどうかを指定します。

- true - これはデフォルト値です。指定した場合、はコマンドラインパラメータのローカル検証 AWS CLI を実行します。
- false – 指定した場合、AWS CLI は AWS サービスエンドポイントに送信する前にコマンドラインパラメータを検証しません。

このエントリーには同等の環境変数あるいはコマンドラインオプションがありません。

```
parameter_validation = false
```

region

このプロファイルを使用してリクエスト AWS リージョン されたコマンドのリクエストを に送信する を指定します。

- 「Amazon Web Services 全般のリファレンス」の「[AWS のリージョンとエンドポイント](#)」にリストされるように、選択されたサービスに使用できるリージョンコードのいずれかを指定できます。
- aws_global では、AWS Security Token Service (AWS STS) や Amazon Simple Storage Service (Amazon S3) などのリージョンエンドポイントに加えて、グローバルエンドポイントをサポートするサービスのグローバルエンドポイントを指定できます。

AWS_REGION 環境変数 AWS_DEFAULT_REGION 環境変数または `--region` コマンドラインオプションを使用して、この値を上書きできます。

```
region = us-west-2
```

retry_mode

使用する再試行モードを指定します AWS CLI。再試行モードには、レガシー (デフォルト)、標準、アダプティブの 3 つがあります。再試行の詳細については、「[AWS CLI での再試行 AWS CLI](#)」を参照してください。

環境変数 AWS_RETRY_MODE を使用して、この値を上書きできます。

```
retry_mode = standard
```

role_arn

AWS CLI コマンドの実行に使用する IAM ロールの Amazon リソースネーム (ARN) を指定します。また、以下のいずれかのパラメータを指定して、このロールを引き受けるアクセス権限を持つ認証情報を特定する必要があります。

- `source_profile`
- `credential_source`

```
role_arn = arn:aws:iam::123456789012:role/role-name
```

環境変数 [AWS_ROLE_ARN](#) により、この設定が上書きされます。

ウェブ ID の使用の詳細については、「[the section called “ウェブ ID を使用したロールの継承”](#)」を参照してください。

role_session_name

ロールセッションにアタッチする名前を指定します。この値は、AWS CLI が AssumeRole オペレーションを呼び出すときに RoleSessionName パラメータに提供され、引き受けたロールユーザーの一部になりますARN `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`。このパラメータはオプションです。この値を指定しない場合、セッション名は自動的に生成されます。この名前は、このセッションに関連付けられたエントリの AWS CloudTrail ログに表示されます。


```
role_session_name = maria_garcia_role
```

環境変数 [AWS_ROLE_SESSION_NAME](#) により、この設定が上書きされます。

ウェブ ID の使用の詳細については、「[the section called “ウェブ ID を使用したロールの継承”](#)」を参照してください。

[services](#)

プロファイルに使用するサービス設定を指定します。

```
[profile dev-s3-specific-and-global]  
endpoint_url = http://localhost:1234  
services = s3-specific  
  
[services s3-specific]  
s3 =  
    endpoint_url = http://localhost:4567
```

詳細については、services セクションの「[the section called “services”](#)」を参照してください。

環境変数 [AWS_ROLE_SESSION_NAME](#) により、この設定が上書きされます。

ウェブ ID の使用の詳細については、「[the section called “ウェブ ID を使用したロールの継承”](#)」を参照してください。

[sdk_ua_app_id](#)

単一の を複数のカスタマーアプリケーションで使用して、 を呼び出し AWS アカウント または AWS のサービス。アプリケーション ID は、 を使用して一連の呼び出しを行ったソースアプリケーションを識別します AWS のサービス。AWS SDKs および サービスは、この値を 以外の方法で使用または解釈して、カスタマーコミュニケーションに返します。例えば、この値を運用 E メールに含めることで、通知に関連付けられているアプリケーションを一意に識別できます。

アプリケーション ID は、最大長が 50 文字の文字列です。文字、数字、および次の特殊文字を使用できます。! \$ % & * + - . , ^ _ ` | ~ デフォルトでは、値は割り当てられません。

```
sdk_ua_app_id = prod1
```

この設定は、[AWS_SDK_UA_APP_ID](#) 環境変数を使用して上書きできます。この値をコマンドラインパラメータとして設定することはできません。

source_profile

AWS CLI パラメータで指定したロールを引き受けるために `role_arn` が使用できる、長期的な認証情報を持つ名前付きプロファイルを指定します。 `source_profile` と `credential_source` の両方を同じプロファイルで指定することはできません。

```
source_profile = production-profile
```

sso_account_id

関連付けられた IAM Identity Center ユーザーに割り当てるアクセス許可を持つ IAM ロールを含む AWS アカウント ID を指定します。

この設定には、環境変数またはコマンドラインオプションはありません。

```
sso_account_id = 123456789012
```

sso_region

AWS アクセスポータルホストを含む AWS リージョンを指定します。これはとは別個であり、デフォルトの `CLIRegion` パラメータとは異なるリージョンにすることができます。

この設定には、環境変数またはコマンドラインオプションはありません。

```
sso_region = us-west-2
```

sso_registration_scopes

`sso-session` に許可するスコープのカンマ区切りのリストです。スコープは、IAM Identity Center ベアトークンの許可されたエンドポイントへのアクセスを許可します。有効なスコープは、`sso:account:access` などの文字列です。この設定は、更新できない従来の設定には適用されません。

```
sso_registration_scopes = sso:account:access
```

sso_role_name

このプロファイルを使用する際にユーザーのアクセス許可を定義する IAM ロールのわかりやすい名前を指定します。

この設定には、環境変数またはコマンドラインオプションはありません。

```
sso_role_name = ReadAccess
```

sso_start_url

組織の AWS アクセスポータルURLを指す `sso_start_url` を指定します。AWS CLI はURL、これを使用して IAM Identity Center サービスとセッションを確立し、ユーザーを認証します。AWS アクセスポータルを検索するにはURL、次のいずれかを使用します。

- 招待メールを開くと、AWS アクセスポータルURLが表示されます。
- で AWS IAM Identity Center コンソールを開きます <https://console.aws.amazon.com/singlesignon/>。AWS アクセスポータルURLは、設定に表示されます。

この設定には、環境変数またはコマンドラインオプションはありません。

```
sso_start_url = https://my-sso-portal.awsapps.com/start
```

use_dualstack_endpoint

デュアルスタックエンドポイントを使用して AWS リクエストを送信できるようにします。IPv4 とトラフィックの両方をサポートするデュアルスタックエンドポイントの詳細については、[Amazon S3 デュアルスタックエンドポイントの使用IPv6](#)」を参照してください。デュアルスタックのエンドポイントは、一部のリージョンでは一部のサービスで利用できます。サービスまたはリージョンにデュアルスタックエンドポイントが存在しない場合 AWS リージョン、リクエストは失敗します。これはデフォルトでは無効になっています。

これは `use_accelerate_endpoint` 設定と相互に排他的となります。

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

- `--endpoint-url` コマンドラインオプション。
- 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
- サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。

4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

use_fips_endpoint

一部の AWS サービスでは、の一部で[連邦情報処理標準 \(FIPS\) 140-2](#) をサポートするエンドポイントを提供しています AWS リージョン。AWS サービスが をサポートする場合 FIPS、この設定 AWS CLI は を使用する FIPS エンドポイントを指定します。標準 AWS エンドポイントとは異なり、FIPS エンドポイントは 140-2 FIPS に準拠した TLS ソフトウェアライブラリを使用します。このエンドポイントは、米国政府とやり取りをする企業で必要とされる場合があります。

この設定が有効になっていても、のサービスに FIPS エンドポイントが存在しない場合 AWS リージョン、AWS コマンドは失敗する可能性があります。この場合は、[--endpoint-url](#) オプションを使用してコマンドで使用するエンドポイントを手動で指定するか、[サービス固有のエンドポイント](#)を使用します。

による FIPS エンドポイントの指定の詳細については AWS リージョン、[FIPS 「サービスによるエンドポイント」](#)を参照してください。

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。

4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービスが最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

[web_identity_token_file](#)

ID プロバイダーによって提供される OAuth 2.0 アクセストークンまたは OpenID Connect ID トークンを含むファイルへのパスを指定します。AWS CLI はこのファイルの内容をロードし、WebIdentityToken 引数として AssumeRoleWithWebIdentity オペレーションに渡します。

環境変数 [AWS_WEB_IDENTITY_TOKEN_FILE](#) により、この設定が上書きされます。

ウェブ ID の使用の詳細については、「[the section called “ウェブ ID を使用したロールの継承”](#)」を参照してください。

tcp_keepalive

AWS CLI クライアントが TCP keep-alive パケットを使用するかどうかを指定します。

このエントリには同等の環境変数あるいはコマンドラインオプションがありません。

```
tcp_keepalive = false
```

S3 カスタマーコマンド設定

Amazon S3 は、が Amazon S3 オペレーション AWS CLI を実行する方法を設定するいくつかの設定をサポートしています。一部は、s3api および s3 名前空間の両方のすべての S3 コマンドに適用されます。その他は、一般的なオペレーションを抽象化し、API オペレーションへのマッピング以上の one-to-one ことを行う S3 「カスタム」コマンド専用です。aws s3 は cp、sync、mv を転送し、rm には S3 転送に使用できる追加の設定があります。

これらすべてのオプションは、config ファイル内のネストされた s3 設定を指定することで構成できます。各設定は、それぞれの行にインデントされます。

Note

これらの設定はすべてオプションです。これらの設定を一切構成せずに、aws s3 転送コマンドを正しく使用できることが必要です。これらの設定は、パフォーマンスを調整したり、上述の aws s3 コマンドを実行する特定の環境に対応したりできるように用意されています。

次の s3 プロファイルの例に示すように、これらの設定はすべて、config ファイルの上位レベル development キーで設定します。

```
[profile development]
s3 =
  max_concurrent_requests = 20
  max_queue_size = 10000
  multipart_threshold = 64MB
  multipart_chunksize = 16MB
  max_bandwidth = 50MB/s
  use_accelerate_endpoint = true
  addressing_style = path
```

次の設定は、s3 あるいは s3api 名前空間内の任意の S3 コマンドに適用されます。

addressing_style

使用するアドレス形式を指定します。これにより、バケット名がホスト名内にあるか、の一部であるかが制御されますURL。有効な値は、path、virtual、auto です。デフォルト値は auto です。

Amazon S3 エンドポイントを構成するには 2 つのスタイルがあります。1 つ目は virtual と呼ばれ、ホスト名の一部としてバケット名が含まれています。例: `https://bucketname.s3.amazonaws.com`。または、path スタイルでは、バケット名をパスのように扱いますURI。例えば、`https://s3.amazonaws.com/bucketname`。のデフォルト値CLIは を使用することです。このはauto、可能な限りvirtualスタイルを使用しようとしませんが、必要に応じてpathスタイルに戻ります。例えば、バケット名がDNS互換性がない場合、バケット名をホスト名の一部にすることはできません。また、パス内に存在する必要が

あります。を使用すると `auto`、CLIはこの条件を検出し、自動的に `path` スタイルに切り替えます。アドレス指定スタイルを `path` に設定する場合は `path`、AWS CLI で設定した AWS リージョンがバケットのリージョンと一致することを確認する必要があります。

`payload_signing_enabled`

`sigv4` ペイロードSHA256に署名するかどうかを指定します。デフォルトでは、`auto` を使用する場合、ストリーミングアップロード (`UploadPart` と `PutObject`) では無効になっています。デフォルトでは、ストリーミングアップロード (`UploadPart` および `PutObject`) `HTTPS`。デフォルトでは、ストリーミングアップロード (`UploadPart` および `PutObject`) `false` に対して `ContentMD5` が存在し (デフォルトで生成されます)、エンドポイントが `HTTPS` を使用している場合にのみ設定されます。

`true` に設定すると、S3 リクエストは追加のコンテンツ検証をチェックサム形式で受け取ります。SHA256チェックサムはユーザー用に計算され、リクエスト署名に含まれます。`false` に設定すると、チェックサムは計算されません。これを無効にすると、チェックサム計算によって生じるパフォーマンスのオーバーヘッドの減少に役立ちます。

`use_accelerate_endpoint`

`s3` および `s3api` のすべてのコマンドに対して Amazon S3 アクセラレートエンドポイントを使用します。デフォルト値は `false` です。これは `use_dualstack_endpoint` 設定と相互に排他的となります。

`true` に設定すると、AWS CLI はすべての Amazon S3 リクエストを `s3-accelerate.amazonaws.com` の S3 Accelerate エンドポイントに送信します。このエンドポイントを使用するには、S3 Accelerate を使用するようにバケットを有効化する必要があります。リクエストはすべて、仮想形式のバケットのアドレス指定 (`my-bucket.s3-accelerate.amazonaws.com`) を使用して送信されます。いずれの `ListBuckets`、`CreateBucket`、`DeleteBucket` リクエストも、S3 高速化エンドポイントには送信されません。高速化エンドポイントでは、これらのオペレーションをサポートしていないためです。また、`--endpoint-url` または `https://s3-accelerate.amazonaws.com` コマンドのいずれかに `http://s3-accelerate.amazonaws.com` パラメータが `s3` あるいは `s3api` に設定されている場合には、この動作を実行できます。

以下の設定は、`s3` 名前空間コマンドセット内のコマンドのみに適用されます。

`max_bandwidth`

Amazon S3 との間でデータをアップロードおよびダウンロードするために消費できる最大帯域幅を指定します。デフォルトは無制限です。

これにより、S3 コマンドが Amazon S3 との間でデータを転送するために使用できる最大帯域幅が制限されます。この値は、アップロードとダウンロードのみに適用され、コピーや削除には適用されません。この値は 1 秒あたりのバイト数で示されます。この値は次のように指定できます。

- 整数。例えば、1048576 は 1 秒あたりの 1 メガバイトの最大帯域幅使用を設定します。
- 整数に続くレートサフィックス。KB/s、MB/s、あるいは GB/s を使用して、レートサフィックスを指定できます。例えば、300KB/s や 10MB/s などです。

一般的には、まず、`max_concurrent_requests` を低くして、低い帯域幅の消費を試してみることが推奨されます。必要なレートの帯域幅消費を十分に制限できない場合には、`max_bandwidth` 設定を使用して、さらに帯域幅消費を制限できます。これは、`max_concurrent_requests` が現在実行中のスレッド数を制御できるためです。代わりに、まず `max_bandwidth` を低くしながら `max_concurrent_requests` 設定を高くすると、スレッドが不必要に待機する結果が生じることがあります。これにより、余分なリソースの消費と接続タイムアウトにつながる可能性があります。

`max_concurrent_requests`

同時リクエストの最大数を指定します。デフォルト値は 10 です。

`aws s3` 転送コマンドはマルチスレッドです。いつでも、複数の Amazon S3 リクエストを実行できます。例えば、コマンドを使用して S3 バケットにファイル `aws s3 cp localdir s3://bucket/ --recursive` をアップロードする場合、はファイル `localdir/file1`、`localdir/file2`、および `localdir/file3` を並行してアップロード AWS CLI できます。`max_concurrent_requests` の設定は、同時に実行できる転送オペレーションの最大数を指定します。

この値を変更する必要には、いくつかの理由が考えられます。

- この値を下げる - 環境によっては、デフォルトの 10 個の同時要求がシステムに過負荷になる場合があります。これによって接続タイムアウトやシステムの応答性の低下が見られる場合があります。この値を小さくすると、S3 転送コマンドのリソース消費が減少します。その代わりに、S3 転送の完了により長くかかる可能性があります。帯域幅を制限するツールを使用する場合、この値を低めに抑える必要が生じることがあります。
- この値を増やす - シナリオによっては、必要なだけのネットワーク帯域幅を使用して、Amazon S3 転送をできるだけ早く完了させることができます。このシナリオでは、利用できるすべてのネットワーク帯域幅を使用するためにデフォルトの同時リクエスト数が十分でないことがあります。この値を大きくすると、Amazon S3 転送が完了する時間を短縮できます。

max_queue_size

タスクキューのタスクの最大数を指定します。デフォルト値は 1000 です。

は、Amazon S3 タスクをキューに入れるモデルを AWS CLI 内部的に使用します。このタスクは、によって番号が制限されているコンシューマーによって実行され、max_concurrent_requests。タスクは通常の場合、単一の Amazon S3 オペレーションをマッピングします。例えば、タスクは PutObjectTask、GetObjectTask、または UploadPartTask にできます。タスクがキューに追加される率は、コンシューマーがタスクを終了する率よりも早くなる可能性があります。無限の拡大を回避するには、タスクキューのサイズを指定するサイズに抑えます。この設定は、タスクキューの最大数値を変更します。

一般的には、この設定を変更する必要はありません。この設定 AWS CLI は、が実行する必要があることを認識しているタスクの数にも対応します。つまり、デフォルトでは、AWS CLI は 1,000 個のタスクしか表示できません。この値を増やすと、キューイングレートがタスクの完了レートよりも速いと仮定して、は必要なタスクの合計数をより迅速に知る AWS CLI ことができます。その代わりに、より大きな max_queue_size はより多くのメモリを必要とします。

multipart_chunksize

が個々のファイルのマルチパート転送 AWS CLI に使用するチャンクサイズを指定します。デフォルト値は 8 MB、最小は 5 MB です。

ファイル転送が multipart_threshold を超えると、AWS CLI はファイルをこのサイズのチャンクに分割します。multipart_threshold と同じ構文をバイト数あるいは整数として使用するか、またはサイズとサフィックスを使用して、この値を指定することができます。

multipart_threshold

が個々のファイルのマルチパート転送 AWS CLI に使用するサイズしきい値を指定します。デフォルト値は 8 MB です。

ファイルのアップロード、ダウンロードやコピーを行うとき、Amazon S3 コマンドは、ファイルがこのサイズを超える場合に、マルチパートオペレーションに切り替わります。この値は、次の 2 つのいずれかに指定します。

- ファイルサイズ (バイト単位)。例えば、1048576 と指定します。
- サイズのサフィックスを使用したファイルサイズ。KB、MB、GB、あるいは TB を使用できません。例: 10MB、1GB。

Note

S3 は、マルチパートオペレーションに使用できる有効値に制約を課すことができません。詳細については、[Amazon Simple Storage Service デベロッパーガイド](#)のS3 マルチパートアップロードのドキュメントを参照してください。

の環境変数の設定 AWS CLI

環境変数は、設定オプションと認証情報を指定する別の方法を提供し、スクリプト作成に役立ちます。

オプションの優先順位

- このトピックで示されている環境変数のいずれかを使用してオプションを指定した場合、設定ファイルのプロファイルからロードされた値は上書きされます。
- AWS CLI コマンドラインでパラメータを使用してオプションを指定すると、対応する環境変数または設定ファイルのプロファイルのいずれかの値が上書きされます。

優先順位と、が使用する認証情報 AWS CLI を決定する方法の詳細については、「」を参照してくださいの[設定 AWS CLI](#)。

トピック

- [環境変数の設定方法](#)
- [AWS CLI サポートされている環境変数](#)

環境変数の設定方法

次の例では、デフォルトのユーザーの環境変数を設定する方法を示します。

Linux or macOS

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
$ export AWS_DEFAULT_REGION=us-west-2
```

環境変数を設定すると、シェルセッションの終了時まで、または変数に別の値を設定するまで、使用する値が変更されます。シェルのスタートアップスクリプトで変数を設定することで、以降のセッションでその変数を永続的にすることができます。

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
C:\> setx AWS_DEFAULT_REGION us-west-2
```

[setx](#) を使用して環境変数を設定すると、現在のコマンドプロンプトセッションおよびコマンド実行後に作成するすべてのコマンドプロンプトセッションで使用する値が変更されます。これは、コマンド実行時にすでに実行されている他のコマンドシェルには影響を及ぼしません。設定をロードするには、ターミナルの再起動が必要になる場合があります。

現在のセッションのみに設定するには

[set](#) を使用して環境変数を設定すると、現在のコマンドプロンプトセッションの終了時まで、または変数を別の値に設定するまで、使用する値が変更されます。

```
C:\> set AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
C:\> set AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
C:\> set AWS_DEFAULT_REGION=us-west-2
```

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
PS C:\> $Env:AWS_DEFAULT_REGION="us-west-2"
```

前の例に示すように、PowerShell プロンプトに環境変数を設定すると、現在のセッションの期間のみ値が保存されます。環境変数設定をすべての PowerShell および コマンドプロンプトセッションにわたって永続的にするには、コントロールパネルのシステムアプリケーションを使用して保存します。または、PowerShell プロファイルに追加することで、今後のすべての PowerShell セッションの変数を設定できます。環境変数の保存やセッション間での保持の詳細については、[PowerShell ドキュメント](#)を参照してください。

AWS CLI サポートされている環境変数

は、以下の環境変数 AWS CLI をサポートしています。

AWS_ACCESS_KEY_ID

IAM アカウントに関連付けられた AWS アクセスキーを指定します。

定義されている場合、この環境変数はプロファイル設定 `aws_access_key_id` の値よりも優先されます。アクセスキー ID をコマンドラインオプションを使用して指定することはできません。

AWS_CA_BUNDLE

証明書の検証に使用するHTTPS証明書バンドルへのパスを指定します。

定義されている場合、この環境変数はプロファイル設定 `ca_bundle` の値よりも優先されます。この環境変数は、`--ca-bundle` コマンドラインパラメータを使用して上書きできます。

AWS_CLI_AUTO_PROMPT

AWS CLI バージョン 2 の自動プロンプトを有効にします。使用できる設定は 2 つあります。

- `on` は、ユーザーが `aws` コマンドを実行しようとするたびに、完全な自動プロンプトモードを使用します。これには、完全なコマンドまたは不完全なコマンドの両方のENTER後に を押すことが含まれます。
- `on-partial` は、部分的な自動プロンプトモードを使用します。コマンドが不完全、またはクライアント側の検証エラーが原因でコマンドを実行できない場合は、自動プロンプトが使用されます。このモードは、既存のスクリプトやランブックがある場合や、すべてのコマンドでプロンプトされるのではなく、慣れていないコマンドに対してのみ自動プロンプトを表示する場合に便利です。

定義されている場合は、この環境変数が `cli_auto_prompt` プロファイル設定の値を上書きします。この環境変数は、`--cli-auto-prompt` および `--no-cli-auto-prompt` コマンドラインパラメータを使用して上書きできます。

AWS CLI バージョン 2 自動プロンプト機能の詳細については、「」を参照してください [でのコマンドプロンプトの有効化と使用 AWS CLI](#)。

AWS_CLI_FILE_ENCODING

テキストファイルに使用するエンコードを指定します。デフォルトでは、エンコードはロケールと一致します。ロケールとは異なるエンコードを設定するには、`aws_cli_file_encoding`

環境変数を使用します。例えば、Windows をデフォルトのエンコーディング で使用する場合CP1252、 を設定すると、 を使用してテキストファイルを開くCLIのように `aws_cli_file_encoding=UTF-8`が設定されますUTF-8。

AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS

カスタム `s3 mv` コマンドを使用するときにはソースバケットと宛先バケットが同じ場合、ソースファイルまたはオブジェクトをそれ自体に移動できるため、ソースファイルまたはオブジェクトが誤って削除される可能性があります。AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS 環境変数と `--validate-same-s3-paths` オプションは、Amazon S3 ソースARNsまたは宛先 でアクセスポイントまたはアクセスポイントエイリアスを検証するかどうかを指定しますURIs。

Note

のパス検証には追加のAPI呼び出し `s3 mv` が必要です。

AWS_CONFIG_FILE

が設定プロファイルの保存 AWS CLI に使用するファイルの場所を指定します。デフォルトのパスは `~/.aws/config` です。

この値は、名前付きプロファイル設定に指定したり、コマンドラインパラメータを使用して指定したりすることはできません。

AWS_DATA_PATH

AWS CLI データのロード `~/.aws/models` 時に の組み込み検索パス外でチェックする追加のディレクトリのリスト。環境変数を設定することで、組み込まれた検索パスに戻る前に最初にチェックする追加のディレクトリを表示します。複数のエントリは `os.pathsep` 文字で区切る必要があります、Linux または macOS では `:`、Windows では `;` を使用します。

AWS_DEFAULT_OUTPUT

使用する [出力形式](#) を指定します。

定義されている場合、この環境変数はプロファイル設定 `output` の値よりも優先されます。この環境変数は、`--output` コマンドラインパラメータを使用して上書きできます。

AWS_DEFAULT_REGION

は、デフォルトでリクエストを送信するサーバーがある AWS リージョン `Default region name` を識別します。通常、お客様の最寄りのリージョンですが、どのリージョンでもかまいません

ん。例えば、「us-west-2」と入力すると、米国西部 (オレゴン) を使用できます。これは、個別のコマンドで指定されない限り、今後のすべてのリクエストが送信されるリージョンです。

Note

を使用するときは AWS CLI、明示的に、またはデフォルトの AWS リージョンを設定して、リージョンを指定する必要があります。使用可能なリージョンのリストについては、「[リージョンとエンドポイント](#)」を参照してください。で使用されるリージョン識別子 AWS CLI は、 および サービスエンドポイントに表示される AWS Management Console URLs ものと同じ名前です。

定義されている場合、この環境変数はプロファイル設定 `region` の値よりも優先されます。 `--region` コマンドラインパラメータ と互換性のある環境変数 を使用して、この `AWS_REGION` 環境変数を AWS SDK 上書きできます。

AWS_EC2_METADATA_DISABLED

Amazon EC2 インスタンスメタデータサービス () の使用を無効にします IMDS。

`true` に設定すると、ユーザー認証情報または設定 (リージョンなど) は からリクエストされません IMDS。

AWS_ENDPOINT_URL

すべてのサービスリクエストに使用されるエンドポイントを指定します。

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. `--endpoint-url` コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。
4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。

- 共有 config ファイルの `services` セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
- 共有 config ファイルの `profile` 内の [endpoint_url](#) 設定によって提供される値。
- [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
- それぞれの URL のデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

AWS_ENDPOINT_URL_<SERVICE>

特定のサービスに使用されるカスタムエンドポイントを指定します。<SERVICE> は AWS のサービス 識別子に置き換えられます。例えば、には `serviceId` の Amazon DynamoDB があります [DynamoDB](#)。このサービスの場合、エンドポイント URL 環境変数は ではなく `AWS_ENDPOINT_URL_DYNAMODB`。

サービス固有の環境変数のリストについては、「[サービス固有の識別子のリスト](#)」を参照してください。

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

- [--endpoint-url](#) コマンドラインオプション。
- 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
- サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) (`AWS_ENDPOINT_URL_DYNAMODB` など) によって提供される値。
- [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
- 共有 config ファイルの `services` セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
- 共有 config ファイルの `profile` 内の [endpoint_url](#) 設定によって提供される値。
- [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
- それぞれの URL のデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web

Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

AWS_IGNORE_CONFIGURED_ENDPOINT_URLS

有効にすると、はすべてのカスタムエンドポイント設定 AWS CLI を無視します。有効な値は、**true** および **false** です。

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントが無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。
4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

AWS_MAX_ATTEMPTS

AWS CLI 再試行ハンドラーが使用する最大再試行回数の値を指定します。最初の呼び出しは、指定した値にカウントされます。再試行の詳細については、「[AWS CLI での再試行 AWS CLI](#)」を参照してください。

定義されている場合、この環境変数はプロファイル設定 `max_attempts` の値よりも優先されません。

AWS_METADATA_SERVICE_NUM_ATTEMPTS

IAM ロールで設定された Amazon EC2 インスタンスの認証情報を取得しようとする、は停止する前にインスタンスメタデータサービスから認証情報を 1 回取得 AWS CLI しようします。コマンドが Amazon EC2 インスタンスで実行されることがわかっている場合は、この値を増やして、あきらめる前に複数回 AWS CLI 再試行できます。

AWS_METADATA_SERVICE_TIMEOUT

インスタンスメタデータサービスに接続する前にタイムアウトするまでの秒数。IAM ロールで設定された Amazon EC2 インスタンスの認証情報を取得しようとする、インスタンスメタデータサービスへの接続はデフォルトで 1 秒後にタイムアウトします。IAM ロールが設定された Amazon EC2 インスタンスで を実行していることがわかっている場合は、必要に応じてこの値を増やすことができます。

AWS_PAGER

出力に使用されるページャープログラムを指定します。デフォルトでは、AWS CLI バージョン 2 はオペレーティングシステムのデフォルトのページャープログラムを通じてすべての出力を返します。

外部ページングプログラムの使用をすべて無効にするには、変数を空の文字列に設定します。

定義されている場合、この環境変数はプロファイル設定 `cli_pager` の値よりも優先されます。

AWS_PROFILE

使用する認証情報とオプションを含む AWS CLI プロファイルの名前を指定します。これは、`credentials` ファイルまたは `config` ファイルに保存されているプロファイルの名前、または、デフォルトプロファイルを使用する場合は値 `default` となります。

この環境変数を指定した場合、設定ファイルの `[default]` という名前のプロファイルを使用する動作は上書きされます。この環境変数は、`--profile` コマンドラインパラメータを使用して上書きできます。

AWS_REGION

リクエストを送信するリージョンを指定する AWS SDK AWS 互換性のある環境変数。

定義されている場合、この環境変数は環境変数 `AWS_DEFAULT_REGION` とプロファイル設定 `region` の値を上書きします。この環境変数は、`--region` コマンドラインパラメータを使用して上書きできます。

AWS_RETRY_MODE

使用する再試行モードを指定します AWS CLI。再試行モードには、レガシー (デフォルト)、標準、アダプティブの 3 つがあります。再試行の詳細については、「[AWS CLI での再試行 AWS CLI](#)」を参照してください。

定義されている場合、この環境変数はプロファイル設定 `retry_mode` の値よりも優先されます。

AWS_ROLE_ARN

AWS CLI コマンドの実行に使用するウェブ ID プロバイダーを持つ IAM ロールの Amazon リソースネーム (ARN) を指定します。

AWS_WEB_IDENTITY_TOKEN_FILE および AWS_ROLE_SESSION_NAME 環境変数で使用されます。

定義されている場合、この環境変数はプロファイル設定 `role_arn` の値よりも優先されます。ロールセッション名をコマンドラインパラメータとして指定することはできません。

Note

この環境変数は、ウェブ ID プロバイダを使用したロール継承にのみ適用され、一般的なロール継承プロバイダ設定には適用されません。

ウェブ ID の使用の詳細については、「[the section called “ウェブ ID を使用したロールの継承”](#)」を参照してください。

AWS_ROLE_SESSION_NAME

ロールセッションにアタッチする名前を指定します。この値は、`assumeRole` オペレーションを AWS CLI 呼び出すときに `RoleSessionName` パラメータに提供され、引き受けたロールユーザーの一部になりますARN `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`。このパラメータはオプションです。この値を指定しない場合、セッション名は自動的に生成されます。この名前は、このセッションに関連付けられたエントリの AWS CloudTrail ログに表示されます。

定義されている場合、この環境変数はプロファイル設定 `role_session_name` の値よりも優先されます。

AWS_ROLE_ARN および AWS_WEB_IDENTITY_TOKEN_FILE 環境変数で使用されます。

ウェブ ID の使用の詳細については、「[the section called “ウェブ ID を使用したロールの継承”](#)」を参照してください。

Note

この環境変数は、ウェブ ID プロバイダを使用したロール継承にのみ適用され、一般的なロール継承プロバイダ設定には適用されません。

AWS_SDK_UA_APP_ID

単一の を複数のカスタマーアプリケーションで使用して、 を呼び出し AWS アカウント または AWS のサービス。アプリケーション ID は、 を使用して一連の呼び出しを行ったソースアプリケーションを識別します AWS のサービス。AWS SDKs および サービスは、この値を 以外の方法で使用または解釈して、カスタマーコミュニケーションに返します。例えば、この値を運用 E メールに含めることで、通知に関連付けられているアプリケーションを特定できます。

デフォルトでは、値はありません。

アプリケーション ID は、最大長が 50 文字の文字列です。文字、数字、および次の特殊文字を使用できます。

```
! $ % & * + - . , ^ _ ` | ~
```

定義されている場合、この環境変数はプロファイル設定 `sdk_ua_app_id` の値よりも優先されます。コマンドラインオプションとしてアプリケーション ID を指定することはできません。

AWS_SECRET_ACCESS_KEY

アクセスキーに関連付けられるシークレットキーを指定します。これは、基本的にアクセスキーの「パスワード」です。

定義されている場合、この環境変数はプロファイル設定 `aws_secret_access_key` の値よりも優先されます。シークレットアクセスキー ID をコマンドラインオプションとして指定することはできません。

AWS_SESSION_TOKEN

AWS STS オペレーションから直接取得した一時的なセキュリティ認証情報を使用している場合に必要なセッショントークン値を指定します。詳細については、AWS CLI コマンドリファレンスにある [assume-role コマンドの「出力」](#) セクションを参照してください。

定義されている場合、この環境変数はプロファイル設定 `aws_session_token` の値よりも優先されます。

AWS_SHARED_CREDENTIALS_FILE

がアクセスキーの保存 AWS CLI に使用するファイルの場所を指定します。デフォルトのパスは `~/.aws/credentials` です。

この値は、名前付きプロファイル設定に指定したり、コマンドラインパラメータを使用して指定したりすることはできません。

AWS_USE_DUALSTACK_ENDPOINT

デュアルスタックエンドポイントを使用して AWS リクエストを送信できるようにします。IPv4 とトラフィックの両方をサポートするデュアルスタックエンドポイントの詳細については、[Amazon S3 デュアルスタックエンドポイントの使用IPv6](#)」を参照してください。デュアルスタックのエンドポイントは、一部のリージョンでは一部のサービスで利用できます。サービスまたはにデュアルスタックエンドポイントが存在しない場合 AWS リージョン、リクエストは失敗します。これはデフォルトでは無効になっています。

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。
4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの `services` セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの `profile` 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web

Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

AWS_USE_FIPS_ENDPOINT

一部の AWS サービスでは、の一部で[連邦情報処理標準 \(FIPS\) 140-2](#) をサポートするエンドポイントを提供しています AWS リージョン。AWS サービスが をサポートする場合FIPS、この設定は AWS CLI を使用するFIPSエンドポイントを指定します。標準 AWS エンドポイントとは異なり、FIPSエンドポイントは 140-2 FIPS に準拠したTLSソフトウェアライブラリを使用します。このエンドポイントは、米国政府とやり取りをする企業で必要とされる場合があります。

この設定が有効になっていても、のサービスにFIPSエンドポイントが存在しない場合 AWS リージョン、AWS コマンドは失敗する可能性があります。この場合は、[--endpoint-url](#) オプションを使用してコマンドで使用するエンドポイントを手動で指定するか、[サービス固有のエンドポイント](#)を使用します。

によるFIPSエンドポイントの指定の詳細については AWS リージョン、[FIPS 「サービスによるエンドポイント](#)」を参照してください。

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。
4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URLのデフォルトエンドポイント AWS のサービス が最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web

Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

[AWS_WEB_IDENTITY_TOKEN_FILE](#)

ID プロバイダーが提供する OAuth 2.0 アクセストークンまたは OpenID Connect ID トークンを含むファイルへのパスを指定します。AWS CLI はこのファイルの内容をロードし、WebIdentityToken 引数として AssumeRoleWithWebIdentity オペレーションに渡します。

AWS_ROLE_ARN および AWS_ROLE_SESSION_NAME 環境変数で使用されます。

定義されている場合、この環境変数はプロファイル設定 web_identity_token_file の値よりも優先されます。

ウェブ ID の使用の詳細については、「[the section called “ウェブ ID を使用したロールの継承”](#)」を参照してください。

Note

この環境変数は、ウェブ ID プロバイダを使用したロール継承にのみ適用され、一般的なロール継承プロバイダ設定には適用されません。

のコマンドラインオプション AWS CLI

では AWS CLI、コマンドラインオプションは、その単一のコマンドのデフォルト設定、対応するプロファイル設定、または環境変数設定を上書きするために使用できるグローバルパラメータです。使用するプロファイルを指定することはできますが、コマンドラインオプションを使用して認証情報を直接指定することはできません。

トピック

- [コマンドラインオプションの使用方法](#)
- [AWS CLI サポートされているグローバルコマンドラインオプション](#)
- [コマンドラインオプションの一般的な用途](#)

コマンドラインオプションの使用方法

ほとんどのコマンドラインオプションは、次の例のプロファイル名 `profile1` などの単純な文字列です。

```
$ aws s3 ls --profile profile1
amzn-s3-demo-bucket1
amzn-s3-demo-bucket2
...
```

引数を使用する各オプションには、オプション名から引数を分離するスペースまたは等号 (=) が必要です。引数値にスペースが含まれている文字列がある場合は、引数を引用符で囲む必要があります。引数の型とパラメータの書式設定の詳細については、「[でのパラメータ値の指定 AWS CLI](#)」を参照してください。

AWS CLI サポートされているグローバルコマンドラインオプション

では、次のコマンドラインオプションを使用して、その1つのコマンドのデフォルト設定、対応するプロファイル設定、または環境変数設定を上書き AWS CLI できます。

`--ca-bundle <string>`

証明書を検証するときに使用する認証局 (CA) SSL証明書バンドルを指定します。

定義されている場合、このオプションはプロファイル設定 `ca_bundle` の値および `AWS_CA_BUNDLE` 環境変数よりも優先されます。

`--cli-auto-prompt`

単一のコマンドに対して自動プロンプトモードを有効にします。以下の例にあるように、これは任意の場所で指定できます。

```
$ aws --cli-auto-prompt
$ aws dynamodb --cli-auto-prompt
$ aws dynamodb describe-table --cli-auto-prompt
```

このオプションは、`aws_cli_auto_prompt` 環境変数と `cli_auto_prompt` プロファイル設定を上書きします。

AWS CLI バージョン 2 自動プロンプト機能の詳細については、「」を参照してください [でのコマンドプロンプトの有効化と使用 AWS CLI](#)。

--cli-binary-format

AWS CLI バージョン 2 がバイナリ入力パラメータをどのように解釈するかを指定します。これには、次のいずれかの値を指定できます。

- `base64` - これはデフォルト値です。バイナリラージオブジェクト (BLOB) として入力される入力パラメータは、`base64` でエンコードされた文字列を受け入れます。真のバイナリコンテンツを渡すには、コンテンツをファイルに置き、ファイルのパスと名前をパラメータの値として `fileb://` プレフィックスを付けて指定します。ファイルに含まれる `base64` エンコードされたテキストを渡すには、ファイルのパスと名前をパラメータの値として `file://` プレフィックスを付けて指定します。
- `raw-in-base64-out` - AWS CLI バージョン 1 のデフォルト。設定の値が `raw-in-base64-out`、`file://` プレフィックスを使用して参照されるファイルはテキストとして読み取られ、`base64` はそれをバイナリにエンコードしようとします。

これは `cli_binary_format` ファイル設定を上書きします。

```
$ aws lambda invoke \  
  --cli-binary-format raw-in-base64-out \  
  --function-name my-function \  
  --invocation-type Event \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

`fileb://` プレフィックス表記を使用してファイル内のバイナリ値を参照する場合、`raw` はファイルに `raw` バイナリコンテンツが含まれていることを AWS CLI 常に想定し、値を変換しようとしません。

`file://` プレフィックス表記を使用してファイル内のバイナリ値を参照する場合、`base64` は現在の `cli_binary_format` 設定に従ってファイルを AWS CLI 処理します。この設定の値が `base64` (明示的に設定されていない場合はデフォルト) の場合、`base64` はファイルに `base64` でエンコードされたテキストが含まれていることを AWS CLI 期待します。設定の値が `raw-in-base64-out` の場合、AWS CLI はファイルに `raw` バイナリコンテンツが含まれていることを期待します。

--cli-connect-timeout *<integer>*

ソケットの最大接続時間を秒単位で指定します。この値をゼロ (0) に設定すると、ソケットの接続は無期限に待機し (ブロックされ)、タイムアウトになりません。

--cli-read-timeout <integer>

ソケットの最大読み込み時間を秒単位で指定します。この値をゼロ (0) に設定すると、ソケットの読み込みは無期限に待機し (ブロックされ)、タイムアウトになりません。

--color <string>

色出力のサポートを指定します。有効な値は、on、off、auto です。デフォルト値は auto です。

--debug

デバッグログ記録を有効にするブールスイッチ。AWS CLI デフォルトでは、は、コマンド出力のコマンド結果に関する成功または失敗に関するクリーンアップ情報を提供します。--debug このオプションでは、完全な Python ログが提供されます。これには、コマンドが予期しない結果を提供する原因をトラブルシューティングするときを使用できるコマンドのオペレーションに関する追加の stderr 診断情報が含まれています。デバッグログを簡単に表示するには、ログをファイルに送信して、情報をより簡単に検索することをお勧めします。これを行うには、次のいずれかを使用します。

stderr の診断情報のみを送信するには、2> debug.txt を追加します。ここで、debug.txt はデバッグファイルに使用する名前です。

```
$ aws servicename commandname options --debug 2> debug.txt
```

出力情報と stderr 診断情報の両方を送信するには、&> debug.txt を追加します。ここで、debug.txt はデバッグファイルに使用する名前です。

```
$ aws servicename commandname options --debug &> debug.txt
```

--endpoint-url <string>

リクエストURLを送信する を指定します。ほとんどのコマンドでは、は選択したサービスと指定された AWS リージョンURLに基づいて AWS CLI を自動的に決定します。ただし、一部のコマンドでは、アカウント固有の を指定する必要がありますURL。 [プライベート 内でエンドポイントを直接ホストVPC](#) するように一部の AWS サービスを設定することもできます。この場合は、指定が必要になる場合があります。

次のコマンド例では、カスタム Amazon S3 エンドポイント を使用しますURL。

```
$ aws s3 ls --endpoint-url http://localhost:4567
```


エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。
4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービスが最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

`--no-cli-auto-prompt`

単一のコマンドに対して自動プロンプトモードを無効にします。

```
$ aws dynamodb describe-table --table-name Table1 --no-cli-auto-prompt
```

このオプションは、[aws_cli_auto_prompt](#) 環境変数と [cli_auto_prompt](#) プロファイル設定を上書きします。

AWS CLI バージョン 2 自動プロンプト機能の詳細については、「」を参照してください [でのコマンドプロンプトの有効化と使用 AWS CLI](#)。

`--no-cli-pager`

コマンドの出力に対するページャーの使用を無効にするブール型スイッチ。

--no-paginate

出力のページ分割を作成するすべてのコマンド結果を受信するために自動的に AWS CLI 行う複数の呼び出しを無効にするブール型スイッチ。つまり、出力の最初のページのみが表示されません。

--no-sign-request

AWS サービスエンドポイントへの HTTP リクエストの署名を無効にするブールスイッチ。これにより、認証情報がロードされることを防ぎます。

--no-verify-ssl

デフォルトでは、は AWS サービスと通信 SSL するとき AWS CLI を使用します。SSL 接続と呼び出しごとに、は SSL 証明書 AWS CLI を検証します。このオプションを使用すると、SSL 証明書を検証するデフォルトの動作が上書きされます。

Warning

このオプションは、ベストプラクティスではありません。を使用する場合 `--no-verify-ssl`、クライアントと AWS サービス間のトラフィックは保護されなくなります。つまり、トラフィックはセキュリティリスクであり、man-in-the-middle エクスプロイトに対して脆弱です。証明書に問題がある場合は、代わりに問題を解決するのが最善です。証明書のトラブルシューティング手順については、「[the section called “SSL 証明書エラー”](#)」を参照してください。

--output *<string>*

このコマンドに使用する出力形式を指定します。次の値のいずれかを指定できます。

- **[json](#)** – 出力は [JSON](#) 文字列としてフォーマットされます。
- **[yaml](#)** – 出力は [YAML](#) 文字列としてフォーマットされます。
- **[yaml-stream](#)** – 出力はストリーミングされ、[YAML](#) 文字列としてフォーマットされます。ストリーミングにより、大きなデータタイプの処理を高速化できます。
- **[text](#)** – 出力は、複数行のタブ区切りの文字列値としてフォーマットされます。これは、grep、sed、または awk などのテキストプロセッサに出力を渡すのに役立ちます。
- **[table](#)** – 出力は、テーブルとしてフォーマットされ、文字の「+|-」を使用してセルの境界を形成します。通常、情報は他の形式よりも読みやすい「わかりやすい」形式で表示されますが、プログラムとしては役立ちません。

--profile <string>

このコマンドに使用する [named profile](#) を指定します。追加の名前付きプロファイルを設定するには、aws configure オプションを指定して --profile コマンドを使用します。

```
$ aws configure --profile <profilename>
```

--クエリ <string>

レスポンスデータのフィルタリングに使用する [JMESPathクエリ](#) を指定します。詳細については、「[での出力のフィルタリング AWS CLI](#)」を参照してください。

--region <string>

このコマンドの AWS リクエストを送信する AWS リージョンを指定します。指定できるすべてのリージョンのリストについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

--version

実行中の AWS CLI プログラムの現在のバージョンを表示するブールスイッチ。

コマンドラインオプションの一般的な用途

コマンドラインオプションの一般的な使用方法には、複数の AWS リージョンでのリソースの確認、および、スクリプティングでの読みやすさや使いやすさのための出力形式の変更が含まれます。次の例では、インスタスがどのリージョンにあるかを見つけるまで、各リージョンに対して describe-instances コマンドを実行します。

```
$ aws ec2 describe-instances --output table --region us-west-1
```

```
-----
|DescribeInstances|
+-----+
```

```
$ aws ec2 describe-instances --output table --region us-west-2
```

```
-----
|                               DescribeInstances                               |
+-----+
||                               Reservations                               ||
|+-----+-----+-----+-----+|
||  OwnerId                       |  012345678901                       ||
||  ReservationId                  |  r-abcdefgh                       ||
|+-----+-----+-----+-----+|
```

```

|||                               Instances                               ||| |
||+-----+-----+-----+-----+-----+-----+-----+-----+||
|||  AmiLaunchIndex             |  0                               |||
|||  Architecture               |  x86_64                       |||
...

```

でのコマンド完了の設定 AWS CLI

AWS Command Line Interface (AWS CLI) には、タブキーを使用して部分的に入力されたコマンドを完了できる bash 互換のコマンド完了機能が含まれています。ほとんどのシステムでは、これを手動で設定する必要があります。

代わりに AWS CLI バージョン 2 自動プロンプト機能の詳細については、「」を参照してください [でのコマンドプロンプトの有効化と使用 AWS CLI](#)。

トピック

- [仕組み](#)
- [Linux または macOS でのコマンド補完の設定。](#)
- [Windows でのコマンド補完の設定。](#)

仕組み

コマンド、パラメータ、またはオプションの一部を入力すると、コマンド補完機能によってコマンドが自動的に入力されるか、コマンドの候補リストが表示されます。コマンドの完了を促すには、コマンドに部分的に入力し、完了キーを押します。通常、**Tab** ほとんどのシェルで

次の例は、コマンド補完を使用するさまざまな方法を示しています。

- コマンドを部分的に入力し、**Tab** コマンドの推奨リストを表示します。

```

$ aws dynamodb dTAB
delete-backup                describe-global-table
delete-item                  describe-global-table-settings
delete-table                 describe-limits
describe-backup              describe-table
describe-continuous-backups  describe-table-replica-auto-scaling
describe-contributor-insights describe-time-to-live
describe-endpoints

```

- パラメータを部分的に入力し、**Tab** パラメータの推奨リストを表示します。

```
$ aws dynamodb delete-table --TAB
--ca-bundle           --endpoint-url           --profile
--cli-connect-timeout --generate-cli-skeleton --query
--cli-input-json      --no-paginate            --region
--cli-read-timeout   --no-sign-request       --table-name
--color               --no-verify-ssl         --version
--debug               --output
```

- パラメータを入力し、**Tab** を押します。**Tab** リソース値の推奨リストを表示します。この機能はバージョン 2 AWS CLI でのみ使用できます。

```
$ aws dynamodb db delete-table --table-name TAB
Table 1           Table 2           Table 3
```

Linux または macOS でのコマンド補完の設定。

Linux または macOS でコマンド補完を設定するには、使用しているシェルの名前と `aws_completer` スクリプトの場所を知っている必要があります。

Note

コマンドの完了は、Amazon Linux を実行する Amazon EC2 インスタンスで自動的に設定され、デフォルトで有効になっています。

トピック

- [コンプリータがフォルダがパスにあることを確認します](#)
- [コマンド補完を有効にする](#)
- [コマンド補完の確認](#)

コンプリータがフォルダがパスにあることを確認します

AWS コンプリータが正常に機能するには、シェルのパスに `aws_completer` が必要です。「`which`」コマンドで、コンプリータがパスにあるかどうかをチェックできます。

```
$ which aws_completer
```

```
/usr/local/bin/aws_completer
```

「which」コマンドで補完するものが見つからない場合は、次の手順を使用してコンプリータのフォルダをパスに追加します。

ステップ 1: AWS コンプリーターを見つける

AWS コンプリータの場所は、使用するインストール方法によって異なります。

- Package Manager - pip、yum、などのプログラムでbrew、apt-get通常、AWS コンプリータ (またはそのシンボリックリンク) を標準のパスロケーションにインストールします。
 - pip を「--user」パラメータを指定せずに使用した場合、デフォルトのパスは「/usr/local/bin/aws_completer」です。
 - pip を「--user」パラメータと共に使用した場合デフォルトのパスは「/home/*username*/.local/bin/aws_completer」です。
- バンドルインストーラ - バンドルされたインストーラを使用した場合、デフォルトのパスは「/usr/local/bin/aws_completer」です。

それ以外のすべてが失敗した場合、find コマンドを使用してファイルシステムを検索して AWS、コンプリータを検索できます。

```
$ find / -name aws_completer
/usr/local/bin/aws_completer
```

ステップ 2: シェルを識別する

使用しているシェルを特定するには、次のコマンドのいずれかを使用します。

- echo \$SHELL – シェルのプログラムファイル名を表示します。これは通常、ログイン後に別のシェルを起動しない限り、使用中のシェルの名前と一致します。

```
$ echo $SHELL
/bin/bash
```

- ps - 現在のユーザーに対して実行中のプロセスを表示します。それらの 1 つはシェルです。

```
$ ps
  PID TTY          TIME CMD
 2148 pts/1    00:00:00 bash
```

```
8756 pts/1    00:00:00 ps
```

ステップ 3: コンプリータをパスに追加する

1. ユーザーフォルダーでシェルのプロファイルスクリプトを見つけます。

```
$ ls -a ~/
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – .bash_profile、.profile、または .bash_login。
 - Zsh – .zshrc
 - Tcsh – .tcshrc、.cshrc、.login
2. 次の例のように、プロファイルスクリプトの末尾にエクスポートコマンドを追加します。「`/usr/local/bin/`」を、前のセクションで検出したフォルダに置き換えます。

```
export PATH=/usr/local/bin/:$PATH
```

3. 変更が有効になるように、プロファイルを現在のセッションに再ロードします。`.bash_profile` を最初のセッションで見つけたシェルスクリプトの名前で置き換えます。

```
$ source ~/bash_profile
```

コマンド補完を有効にする

コンプリータがパスにあることを確認したら、使用しているシェル上で適切なコマンドを実行して、コマンド補完を有効にします。コマンドをシェルのプロファイルに追加して、新しいシェルを開くたびに実行できます。各コマンドで、`/usr/local/bin/` のシステムで見つかったパス。 [コンプリータがフォルダがパスにあることを確認します](#)

- **bash** - 組み込みコマンド「complete」を使用します。

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

前のコマンドを `~/bashrc` に追加して、新しいシェルを開くたびに実行します。`~/bash_profile` はソースとして `~/bashrc` を使用して、コマンドがログインシェルでも実行されるようにできます。

- **zsh** - コマンド補完を実行するには、`bashcompinit` プロファイルスクリプトの最後に次の自動ロード行を追加して「`~/.zshrc`」を実行する必要があります。

```
$ autoload bashcompinit && bashcompinit
$ autoload -Uz compinit && compinit
```

コマンド補完を有効にするには、組み込みコマンド `complete` を使用します。

```
$ complete -C '/usr/local/bin/aws_completer' aws
```

前のコマンドを `~/.zshrc` に追加して、新しいシェルを開くたびに実行します。

- **tcsh** - 「`tcsh`」の補完は、補完の振る舞いを定義するためのワードタイプとパターンを取りません。

```
> complete aws 'p/*/'`aws_completer`/'
```

前のコマンドを `~/.tschrc` に追加して、新しいシェルを開くたびに実行します。

コマンド補完を有効にした後は、「[コマンド補完の確認](#)」は動作しています。

コマンド補完の確認

コマンド補完を有効にしたら、シェルをリロードし、コマンドの一部を入力し、`Tab` を押して使用可能なコマンドを表示します。

```
$ aws sTAB
s3          ses          sqs          sts          swf
s3api       sns          storagegateway support
```

Windows でのコマンド補完の設定。

Note

がさまざまな完了キーを含む完了 PowerShell を処理する方法については、Microsoft PowerShell Docs の「[about_Tab_Expansion](#)」を参照してください。

Windows PowerShell でのコマンド完了を有効にするには、 の次の手順を実行します PowerShell。

1. 次のコマンドで「\$PROFILE」を開きます。

```
PS C:\> Notepad $PROFILE
```

「\$PROFILE」がない場合は、次のコマンドを使用してユーザープロファイルを作成します。

```
PS C:\> if (!(Test-Path -Path $PROFILE ))  
{ New-Item -Type File -Path $PROFILE -Force }
```

PowerShell プロファイルの詳細については、Microsoft Docs ウェブサイトの [Windows でプロファイルを使用する方法 PowerShell ISE](#)を参照してください。

2. コマンド補完を有効にするには、次のコードブロックをプロファイルに追加し、保存し、ファイルを閉じます。

```
Register-ArgumentCompleter -Native -CommandName aws -ScriptBlock {  
    param($commandName, $wordToComplete, $cursorPosition)  
    $env:COMP_LINE=$wordToComplete  
    if ($env:COMP_LINE.Length -lt $cursorPosition){  
        $env:COMP_LINE=$env:COMP_LINE + " "  
    }  
    $env:COMP_POINT=$cursorPosition  
    aws_completer.exe | ForEach-Object {  
        [System.Management.Automation.CompletionResult]::new($_, $_,  
'ParameterValue', $_)  
    }  
    Remove-Item Env:\COMP_LINE  
    Remove-Item Env:\COMP_POINT  
}
```

3. コマンド補完を有効にしたらシェルをリロードして、コマンドの一部を入力し、[Tab] (タブ) を押して使用可能なコマンドを順次表示させます。

```
$ aws sTab
```

```
$ aws s3
```

補完で使用できるコマンドを表示するには、コマンドの一部を入力して、Ctrl + [Space] (スペース) を押します。

```
$ aws sCtrl + Space
s3          ses          sqs          sts          swf
s3api      sns          storagegateway support
```

AWS CLI での再試行 AWS CLI

このトピックでは AWS CLI、予期しない問題によりが AWS サービスへの呼び出しが失敗する様子について説明します。これらの問題は、サーバー側で発生するか、呼び出しを実行しようとしている AWS のサービスからのレート制限が原因で発生する可能性があります。通常、これらの種類のエラーは特別な処理を必要とせず、多くの場合、短い待機期間の後に自動的に再度呼び出しが実行されます。AWS CLI には、このようなエラーや例外が発生した場合に、AWS サービスへのクライアントト呼び出しを再試行するのに役立つ多くの機能が用意されています。

トピック

- [使用可能な再試行モード](#)
- [再試行モードの設定](#)
- [再試行ログの表示](#)

使用可能な再試行モード

AWS CLI には、バージョンに応じて選択できる複数のモードがあります。

- [レガシー再試行モード](#)
- [標準再試行モード](#)
- [アダプティブ再試行モード](#)

レガシー再試行モード

レガシーモードでは、以下を含む、機能が制限された古い再試行ハンドラーが使用されます。

- 最大再試行回数のデフォルト値は 4 で、呼び出しの試行回数は合計 5 回です。この値は、`max_attempts` 設定パラメータを使用して上書きできます。
- DynamoDB では、最大再試行回数のデフォルト値は 9 で、呼び出しの試行回数は合計 10 回です。この値は、`max_attempts` 設定パラメータを使用して上書きできます。
- 次の限られた数のエラー/例外を再試行します。

- 一般的なソケット/接続エラー:
 - `ConnectionError`
 - `ConnectionClosedError`
 - `ReadTimeoutError`
 - `EndpointConnectionError`
- サービス側のスロットリング/制限エラーと例外:
 - `Throttling`
 - `ThrottlingException`
 - `ThrottledException`
 - `RequestThrottledException`
 - `ProvisionedThroughputExceededException`
- 429、500、502、503、504、509 など、いくつかのHTTPステータスコードで再試行します。
- 再試行には、基本係数 2 によるエクスポネンシャルバックオフが含まれます。

標準再試行モード

標準モードは、レガシーよりも多くの機能を備えた AWS SDKs 全体の再試行ルールの標準セットです。このモードは、AWS CLI バージョン 2 のデフォルトです。標準モードは AWS CLI バージョン 2 用に作成され、AWS CLI バージョン 1 にバックポートされます。標準モードの機能は次のとおりです。

- 最大再試行回数のデフォルト値は 2 で、合計で 3 回呼び出しが試みられます。この値は、`max_attempts` 設定パラメータを使用して上書きできます。
- 次の拡張されたエラー/例外リストを再試行します。
 - 一時的なエラー/例外
 - `RequestTimeout`
 - `RequestTimeoutException`
 - `PriorRequestNotComplete`
 - `ConnectionError`
 - `HTTPClientError`
 - サービス側のスロットリング/制限エラーと例外:

- ThrottlingException
 - ThrottledException
 - RequestThrottledException
 - TooManyRequestsException
 - ProvisionedThroughputExceededException
 - TransactionInProgressException
 - RequestLimitExceeded
 - BandwidthLimitExceeded
 - LimitExceededException
 - RequestThrottled
 - SlowDown
 - EC2ThrottledException
- わかりにくい一時的なエラーコードに対して再試行します。具体的には、これらのHTTPステータスコード: 500、502、503、504。
 - 再試行には、最大バックオフ時間が 20 秒の場合、基本係数 2 によるエクスポネンシャルバックオフが含まれます。

アダプティブ再試行モード

Warning

アダプティブモードは実験的なモードであり、機能と動作の両方について変更される可能性があります。

アダプティブ再試行モードは、標準モードのすべての機能を含む実験的な再試行モードです。アダプティブモードでは、標準モード機能に加えて、トークンバケットおよびレート制限変数の使用を通じて、クライアント側のレート制限も導入されます。これらの変数は、再試行するたびに動的に更新されます。このモードは、AWS サービスからのエラー/例外状態のレスポンスに適応するクライアント側の再試行に柔軟性を提供します。

新しい再試行のたびに、アダプティブモードは、AWS サービスからのレスポンスに表示されるエラー、例外、またはHTTPステータスコードに基づいてレート制限変数を変更します。その後、クラ

リアントの新しい呼び出しレートを計算するためにこれらのレート制限変数が使用されず。サービスからの AWS 各例外/エラーまたは非成功 HTTP レスポンス (上記のリストに記載) は、成功に達するか、トークンバケットが枯渇するか、設定された最大試行値に達するまで再試行が発生すると、レート制限変数を更新します。

再試行モードの設定

AWS CLI には、クライアントオブジェクトの作成時に考慮すべきさまざまな再試行設定と設定方法が含まれています。

使用可能な設定方法

では AWS CLI、ユーザーは次の方法で再試行を設定できます。

- 環境変数
- AWS CLI 設定ファイル

次の再試行オプションをカスタマイズできます。

- 再試行モード - が AWS CLI 使用する再試行モードを指定します。前述のように、レガシー、標準、アダプティブの 3 つの再試行モードを使用できます。AWS CLI バージョン 2 が標準です。
- 最大試行回数 - AWS CLI 再試行ハンドラーが使用する最大再試行回数の値を指定します。最初の呼び出しは、指定した値にカウントされます。デフォルト値は 5 です。

環境変数での再試行設定の定義

の再試行設定を定義するには AWS CLI、オペレーティングシステムの環境変数を更新します。

再試行環境変数は次のとおりです。

- AWS_RETRY_MODE
- AWS_MAX_ATTEMPTS

環境変数の詳細については、「[の環境変数の設定 AWS CLI](#)」を参照してください。

設定ファイルでの再試行 AWS 設定の定義

再試行設定を変更するには、グローバル AWS 設定ファイルを更新します。AWS 設定ファイルのデフォルトの場所は `~/.aws/config` です。

AWS 設定ファイルの例を次に示します。

```
[default]
retry_mode = standard
max_attempts = 6
```

設定ファイルの詳細については、「[の設定と認証情報ファイルの設定 AWS CLI](#)」を参照してください。

再試行ログの表示

は Boto3 の再試行方法とログ記録 AWS CLI を使用します。任意のコマンドで `--debug` オプションを使用して、デバッグログを受け取ることができます。`--debug` オプションの使用の詳細については、「[のコマンドラインオプション AWS CLI](#)」を参照してください。

デバッグログで「retry」を検索すると、必要な再試行情報が表示されます。再試行のクライアントログエントリは、有効にした再試行モードによって異なります。

レガシーモード:

再試行メッセージは、`botocore.retryhandler` によって生成されます。次の 3 つのメッセージのいずれかが表示されます。

- No retry needed
- Retry needed, action of: *<action_name>*
- Reached the maximum number of retry attempts: *<attempt_number>*

標準モードまたはアダプティブモード:

再試行メッセージは、`botocore.retries.standard` によって生成されます。次の 3 つのメッセージのいずれかが表示されます。

- No retrying request
- Retry needed, retrying request after delay of: *<delay_value>*
- Retry needed but retry quota reached, not retrying request

botocore 再試行の完全な定義ファイルについては、botocore GitHub リポジトリの [_retry.json](#) を参照してください。

のHTTPプロキシの使用 AWS CLI

プロキシサーバー AWS を介して にアクセスするには、プロキシサーバーが使用するDNSドメイン名または IP アドレスとポート番号を使用してHTTP_PROXY、 および HTTPS_PROXY環境変数を設定できます。

トピック

- [例の使用](#)
- [プロキシを認証する](#)
- [Amazon EC2インスタンスでのプロキシの使用](#)
- [トラブルシューティング](#)

例の使用

Note

次の例は、環境変数名をすべて大文字で示しています。ただし、大文字と小文字を区別して変数を 2 回指定した場合、小文字が優先されます。システムでの混同や予期しない動作を回避するため、各変数は 1 回だけ定義することをお勧めします。

次の例は、プロキシの明示的な IP アドレスまたはプロキシの IP アドレスに解決されるDNS名前を使用する方法を示しています。どちらの場合でも、コロンおよびクエリが送信されるべきポート番号を追加することができます。

Linux or macOS

```
$ export HTTP_PROXY=http://10.15.20.25:1234
$ export HTTP_PROXY=http://proxy.example.com:1234
$ export HTTPS_PROXY=http://10.15.20.25:5678
$ export HTTPS_PROXY=http://proxy.example.com:5678
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx HTTP_PROXY http://10.15.20.25:1234
C:\> setx HTTP_PROXY http://proxy.example.com:1234
```

```
C:\> setx HTTPS_PROXY http://10.15.20.25:5678
C:\> setx HTTPS_PROXY http://proxy.example.com:5678
```

[setx](#) を使用して環境変数を設定すると、現在のコマンドプロンプトセッションおよびコマンド実行後に作成するすべてのコマンドプロンプトセッションで使用する値が変更されます。これは、コマンド実行時にすでに実行されている他のコマンドシェルには影響を及ぼしません。

現在のセッションのみに設定するには

[set](#) を使用して環境変数を設定すると、現在のコマンドプロンプトセッションの終了時まで、または変数を別の値に設定するまで、使用する値が変更されます。

```
C:\> set HTTP_PROXY=http://10.15.20.25:1234
C:\> set HTTP_PROXY=http://proxy.example.com:1234
C:\> set HTTPS_PROXY=http://10.15.20.25:5678
C:\> set HTTPS_PROXY=http://proxy.example.com:5678
```

プロキシを認証する

Note

AWS CLI はNTLMプロキシをサポートしていません。NTLM または Kerberos プロトコルプロキシを使用している場合は、[Cntlm](#) などの認証プロキシを介して接続できる場合があります。

はHTTP基本認証 AWS CLI をサポートしています。次のようにURL、プロキシでユーザー名とパスワードを指定します。

Linux or macOS

```
$ export HTTP_PROXY=http://username:password@proxy.example.com:1234
$ export HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx HTTP_PROXY http://username:password@proxy.example.com:1234
```



```
C:\> setx HTTPS_PROXY http://username:password@proxy.example.com:5678
```

現在のセッションのみに設定するには

```
C:\> set HTTP_PROXY=http://username:password@proxy.example.com:1234  
C:\> set HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Amazon EC2インスタンスでのプロキシの使用

IAM ロールがアタッチされた状態で起動された Amazon EC2 インスタンスでプロキシを設定する場合は、[インスタンスメタデータ](#) へのアクセスに使用されるアドレスを免除してください。これを行うには、NO_PROXY 環境変数をインスタンスメタデータサービスの IP アドレス「169.254.169.254」に設定します。このアドレスは変化しません。

Linux or macOS

```
$ export NO_PROXY=169.254.169.254
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx NO_PROXY 169.254.169.254
```

現在のセッションのみに設定するには

```
C:\> set NO_PROXY=169.254.169.254
```

トラブルシューティング

で問題が発生した場合は AWS CLI、トラブルシューティングの手順[エラーのトラブルシューティング](#)については、「」を参照してください。最も関連性の高いトラブルシューティングステップについては、「[the section called “SSL 証明書エラー”](#)」を参照してください。

でのエンドポイントの使用 AWS CLI

プログラムでに接続するには AWS のサービス、エンドポイントを使用します。エンドポイントは、AWS ウェブサービスのエンリポイント URL のです。AWS Command Line Interface

(AWS CLI) は、内の各サービスのデフォルトエンドポイントを自動的に使用しますが AWS リージョン、APIリクエストに代替エンドポイントを指定できます。

エンドポイントのトピック

- [1つのコマンドのエンドポイントを設定する](#)
- [すべてののグローバルエンドポイントを設定する AWS のサービス](#)
- [すべてのにFIPsエンドポイントを使用するようにを設定する AWS のサービス](#)
- [すべての AWS のサービスでデュアルスタックのエンドポイントを使用するように設定](#)
- [サービス固有のエンドポイントを設定します。](#)
 - [サービス固有のエンドポイント: 環境変数](#)
 - [サービス固有のエンドポイント: 共有 config ファイル](#)
 - [サービス固有のエンドポイント: サービス固有の識別子のリスト](#)
- [エンドポイント設定と設定の優先順位](#)

1つのコマンドのエンドポイントを設定する

1つのコマンドのエンドポイント設定や環境変数を上書きするには、[--endpoint-url](#) コマンドラインオプションを使用します。次のコマンド例では、カスタム Amazon S3 エンドポイントを使用しますURL。

```
$ aws s3 ls --endpoint-url http://localhost:4567
```

すべてののグローバルエンドポイントを設定する AWS のサービス

すべてのサービスのリクエストをカスタムエンドポイントにルーティングするにはURL、次のいずれかの設定を使用します。

- 環境変数:
 - [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) - 設定されたエンドポイント を無視しますURLs。

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

現在のセッションのみに設定するには

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS_ENDPOINT_URL](#) - グローバルエンドポイント を設定しますURL。

Linux or macOS

```
$ export AWS_ENDPOINT_URL=http://localhost:4567
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx AWS_ENDPOINT_URL http://localhost:4567
```

現在のセッションのみに設定するには

```
C:\> set AWS_ENDPOINT_URL=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL="http://localhost:4567"
```

- config ファイル:
- [ignore_configure_endpoint_urls](#) - 設定されたエンドポイント を無視しますURLs。

```
ignore_configure_endpoint_urls = true
```

- [endpoint_url](#) - グローバルエンドポイント を設定しますURL。

```
endpoint_url = http://localhost:4567
```

サービス固有のエンドポイントと `--endpoint-url` コマンドラインオプションはすべてのグローバルエンドポイントよりも優先されます。

すべての `fips` エンドポイントを使用するように を設定する AWS のサービス

エンドポイントを使用するすべてのサービスのリクエストをルーティングするにはFIPs、次のいずれかを使用します。

- [AWS_USE_FIPS_ENDPOINT](#) 環境変数

Linux or macOS

```
$ export AWS_USE_FIPS_ENDPOINT=true
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx AWS_USE_FIPS_ENDPOINT true
```

現在のセッションのみに設定するには

```
C:\> set AWS_USE_FIPS_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_FIPS_ENDPOINT="true"
```

- [use_fips_endpoint](#) ファイルの設定。

```
use_fips_endpoint = true
```

一部の AWS サービスでは、[連邦情報処理標準 \(FIPS\) 140-2](#) をサポートするエンドポイントを提供しています AWS リージョン。AWS サービスが をサポートする場合 FIPS、この設定 AWS CLI は を使用する FIPS エンドポイントを指定します。標準 AWS エンドポイントとは異なり、FIPS エンドポイントは 140-2 FIPS に準拠した TLS ソフトウェア ライブラリを使用します。このエンドポイントは、米国政府とやり取りをする企業で必要とされる場合があります。

この設定が有効になっていても、 のサービスに FIPS エンドポイントが存在しない場合 AWS リージョン、AWS コマンドは失敗する可能性があります。この場合は、[--endpoint-url](#) オプションを使用してコマンドで使用するエンドポイントを手動で指定するか、[サービス固有のエンドポイント](#)を使用します。

による FIPS エンドポイントの指定の詳細については AWS リージョン、[FIPS 「サービスによるエンドポイント」](#)を参照してください。

すべての AWS のサービスでデュアルスタックのエンドポイントを使用するように設定

利用可能な場合に、すべてのサービスでデュアルスタックエンドポイントを使用するようにリクエストをルーティングするには、以下の設定のいずれかを使用します。

- [AWS_USE_DUALSTACK_ENDPOINT](#) 環境変数

Linux or macOS

```
$ export AWS_USE_DUALSTACK_ENDPOINT=true
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx AWS_USE_DUALSTACK_ENDPOINT true
```

現在のセッションのみに設定するには

```
C:\> set AWS_USE_DUALSTACK_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_DUALSTACK_ENDPOINT="true"
```

- [use_dualstack_endpoint](#) ファイルの設定。

```
use_dualstack_endpoint = true
```

デュアルスタックエンドポイントを使用して AWS リクエストを送信できるようにします。IPv4 とトラフィックの両方をサポートするデュアルスタックエンドポイントの詳細については、[Amazon S3 デュアルスタックエンドポイントの使用IPv6](#)」を参照してください。デュアルスタックのエンドポイントは、一部のリージョンでは一部のサービスで利用できます。サービスまたはにデュアルスタックエンドポイントが存在しない場合 AWS リージョン、リクエストは失敗します。これはデフォルトでは無効になっています。

サービス固有のエンドポイントを設定します。

サービス固有のエンドポイント設定では、AWS CLI リクエストに選択した永続エンドポイントを使用するオプションが提供されます。これらの設定により、ローカルエンドポイント、VPCエンドポイント、およびサードパーティーのローカル AWS 開発環境を柔軟にサポートできます。テスト環境と本番環境には異なるエンドポイントを使用できます。個々の URLのエンドポイントを指定できず AWS のサービス。

サービス固有のエンドポイントは以下の方法で指定できます。

- 1つのコマンドラインオプション [--endpoint-url](#)。
- 環境変数:
 - [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) - コマンドラインで指定されていない限り URLs、設定されたすべてのエンドポイント を無視します。
 - [AWS_ENDPOINT_URL_<SERVICE>](#) - 特定のサービスに使用されるカスタムエンドポイントを指定します。<SERVICE> は AWS のサービス 識別子に置き換えられます。すべてのサービス固有の変数については、「[the section called “サービス固有の識別子のリスト”](#)」を参照してください。
- config ファイル:
 - [ignore_configure_endpoint_urls](#) - 環境変数を使用して、またはコマンドラインで指定されていない限り URLs、設定されたすべてのエンドポイント を無視します。
 - [endpoint_url](#) ファイル設定と組み合わされた config ファイルの [services](#) セクション。

サービス固有のエンドポイントに関するトピック:

- [サービス固有のエンドポイント: 環境変数](#)
- [サービス固有のエンドポイント: 共有 config ファイル](#)
- [サービス固有のエンドポイント: サービス固有の識別子のリスト](#)

サービス固有のエンドポイント: 環境変数

環境変数は Config ファイルの設定を上書きしますが、コマンドラインで指定されたオプションは上書きしません。すべてのプロファイルにデバイス上の同じエンドポイントを使用させたい場合は、環境変数を使用してください。

サービス固有の環境変数は次のとおりです。

- [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) - コマンドラインで指定されていない限りURLs、設定されたすべてのエンドポイント を無視します。

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

現在のセッションのみに設定するには

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS_ENDPOINT_URL_<SERVICE>](#) - 特定のサービスに使用されるカスタムエンドポイントを指定します。<SERVICE> は AWS のサービス 識別子に置き換えられます。すべてのサービス固有の変数については、「[the section called “サービス固有の識別子のリスト”](#)」を参照してください。

次の環境変数の例では、AWS Elastic Beanstalkのエンドポイントを設定します。

Linux or macOS

```
$ export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

Windows Command Prompt

すべてのセッションに設定するには

```
C:\> setx AWS_ENDPOINT_URL_ELASTIC_BEANSTALK http://localhost:4567
```

現在のセッションのみに設定するには

```
C:\> set AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL_ELASTIC_BEANSTALK="http://localhost:4567"
```

環境変数の設定については、「[the section called “環境可変”](#)」を参照してください。

サービス固有のエンドポイント: 共有 **config** ファイル

共有 config ファイルでは、`endpoint_url` は複数のセクションで使用されます。サービス固有のエンドポイントを設定するには、`services` セクション内のサービス ID キーの下にネストされた `endpoint_url` 設定を使用します。共有 config ファイル内の `services` セクションの定義について詳しくは、「[the section called “services”](#)」を参照してください。

次の例では、`services` セクションを使用して、URL Amazon S3 のサービス固有のエンドポイントと、他のすべてのサービスに使用されるカスタムグローバルエンドポイントを設定します。

```
[profile dev1]  
endpoint_url = http://localhost:1234  
services = s3-specific  
  
[services testing-s3]  
s3 =  
  endpoint_url = http://localhost:4567
```


1つのプロファイルで複数のサービスのエンドポイントを設定できます。次の例では、Amazon S3とURLsのサービス固有のエンドポイントを同じプロファイル AWS Elastic Beanstalk に設定します。

services セクションで使用するすべてのサービス識別子キーのリストについては、「[サービス固有の識別子のリスト](#)」を参照してください。

```
[profile dev1]
services = testing-s3-and-eb

[services testing-s3-and-eb]
s3 =
  endpoint_url = http://localhost:4567
elastic_beanstalk =
  endpoint_url = http://localhost:8000
```

サービス設定セクションは複数のプロファイルで使用できます。次の例では、2つのプロファイルが同じ services 定義を使用しています。

```
[profile dev1]
output = json
services = testing-s3

[profile dev2]
output = text
services = testing-s3

[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567
```

サービス固有のエンドポイント: サービス固有の識別子のリスト

AWS のサービス 識別子は、すべてのスペースをアンダースコアに置き換え、すべての文字を小文字 serviceId に置き換えることでAPI、モデルの に基づいています。

次のサービス識別子の例では を使用しています AWS Elastic Beanstalk。AWS Elastic Beanstalk は serviceId の を持つため [Elastic Beanstalk](#)、サービス識別子キーは です elastic_beanstalk。

次の表は、サービス固有の識別子、config ファイルキー、環境変数をすべて一覧にしたものです。

エンドポイント設定と設定の優先順位

エンドポイント設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。AWS CLI エンドポイント構成設定は、次の順序で優先されます。

1. [--endpoint-url](#) コマンドラインオプション。
2. 有効にすると、[AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) グローバルエンドポイント環境変数またはプロファイル設定 [ignore_configure_endpoint_urls](#) はカスタムエンドポイントを無視します。
3. サービス固有の環境変数 [AWS_ENDPOINT_URL_<SERVICE>](#) ([AWS_ENDPOINT_URL_DYNAMODB](#) など) によって提供される値。
4. [AWS_USE_DUALSTACK_ENDPOINT](#)、[AWS_USE_FIPS_ENDPOINT](#)、および [AWS_ENDPOINT_URL](#) 環境変数によって提供される値。
5. 共有 config ファイルの services セクション内の [endpoint_url](#) 設定によって提供されるサービス固有のエンドポイント値。
6. 共有 config ファイルの profile 内の [endpoint_url](#) 設定によって提供される値。
7. [use_dualstack_endpoint](#)、[use_fips_endpoint](#)、および [endpoint_url](#) の設定。
8. それぞれの URL のデフォルトエンドポイント AWS のサービスが最後に使用されます。各リージョンで使用できる標準的なサービスエンドポイントについては、「Amazon Web Services 全般のリファレンス」の「[AWS リージョンとエンドポイント](#)」を参照してください。

の認証認証情報とアクセス認証情報 AWS CLI

サービスを使用して を開発する AWS ときに、 が で AWS CLI AWS 認証する方法を確立する必要があります。 のプログラムによるアクセス用の認証情報を設定するには AWS CLI、次のいずれかのオプションを選択します。オプションは推奨順になっています。

プログラムによるアクセスを必要とするユーザー	目的	手順
ワークフォース ID (AWS IAM Identity Center ユーザー)	(推奨) 短期の認証情報を使用します。	the section called “IAM Identity Center 認証”
IAM	短期の認証情報を使用します。	the section called “短期の認証情報”
IAM またはワークフォース ID (AWS IAM Identity Center ユーザー)	認証情報には Amazon EC2 インスタンスメタデータを使用します。	the section called “Amazon EC2 インスタンスメタデータを の認証情報として使用する AWS CLI”
IAM またはワークフォース ID (AWS IAM Identity Center ユーザー)	別の認証情報メソッドと組み合わせて、アクセス許可のロールを継承します。	the section called “IAM ロール”
IAM	(非推奨) 長期の認証情報を使用します。	the section called “IAM ユーザー”
IAM またはワークフォース ID (AWS IAM Identity Center ユーザー)	(非推奨) 別の認証情報メソッドと組み合わせます。ただし、AWS CLI以外の場所に保存されている認証情報値を使用します。	the section called “外部認証情報”

設定と認証情報の優先順位

認証情報と設定は、システム環境変数やユーザー環境変数、ローカル AWS 設定ファイルなど複数の場所に配置され、コマンドラインでパラメータとして明示的に宣言されます。特定の認証は、他の認証よりも優先されます。AWS CLI 認証設定は、次の順番で優先されます。

1. [コマンドラインオプション](#) - `--region`、`--output`、`--profile` パラメータなど、他の任意の場所にある設定を上書きします。
2. [環境変数](#) - システムの環境変数に値を保存できます。
3. [ロールを引き受ける](#) - 設定または `aws sts assume-role` コマンドを使用して IAM ロールのアクセス許可を引き受けます。
4. [ウェブ ID でロールを引き受ける](#) - 設定または `aws sts assume-role` コマンドを使用して、ウェブ ID を使用して IAM ロールのアクセス許可を引き受けます。
5. [AWS IAM Identity Center](#) - `config` ファイルに保存されている IAM Identity Center 設定は、`aws configure sso` コマンドを実行すると更新されます。その後、`aws sso login` コマンドを実行すると認証情報が認証されます。「`config`」ファイルは、Linux または macOS では「`~/.aws/config`」、Windows では「`C:\Users\USERNAME\.aws\config`」にあります。
6. [認証情報ファイル](#) - コマンド `aws configure` を実行すると、`credentials` ファイルと `config` ファイルが更新されます。「`credentials`」ファイルは、Linux または macOS では「`~/.aws/credentials`」、Windows では「`C:\Users\USERNAME\.aws\credentials`」にあります。
7. [カスタムプロセス](#) - 外部ソースから認証情報を取得します。
8. [設定ファイル](#) - コマンド `aws configure` を実行すると、`credentials` ファイルと `config` ファイルが更新されます。「`config`」ファイルは、Linux または macOS では「`~/.aws/config`」、Windows では「`C:\Users\USERNAME\.aws\config`」にあります。
9. [コンテナ認証情報](#) - IAM ロールを各 Amazon Elastic Container Service (Amazon ECS) タスク定義に関連付けることができます。関連付けられると、そのロールの一時認証情報は、そのタスクのコンテナで使用できるようになります。詳細については、[IAM 「Amazon Elastic Container Service デベロッパーガイド」の「タスクのロール」](#)を参照してください。
10. [Amazon EC2 インスタンスプロファイル認証情報](#) - IAM ロールを各 Amazon Elastic Compute Cloud (Amazon EC2) インスタンスに関連付けることができます。関連付けられると、そのロールの一時認証情報は、インスタンスで実行中のコードで使用できるようになります。認証情報は Amazon EC2 メタデータサービスを介して配信されます。詳細については、[IAM 「Amazon ユーザーガイド」の「Amazon のロール EC2」](#)と IAM 「ユーザーガイド」の「[インスタンスプロファイルの使用](#)」を参照してください。 EC2

このセクションのその他のトピック

- [the section called “IAM Identity Center 認証”](#)
- [the section called “短期の認証情報”](#)
- [the section called “IAM ロール”](#)
- [the section called “IAM ユーザー”](#)
- [the section called “Amazon EC2インスタンスメタデータを の認証情報として使用する AWS CLI”](#)
- [the section called “外部認証情報”](#)

を使用した IAM Identity Center 認証の設定 AWS CLI

このトピックでは、AWS IAM Identity Center (IAM Identity Center) AWS CLI を使用して を設定して、AWS CLI コマンドを実行するための認証情報を取得する方法について説明します。IAM Identity Center でユーザーを認証して、config ファイルを介して AWS CLI コマンドを実行するための認証情報を取得する方法は主に 2 つあります。

- (推奨) SSO トークンプロバイダーの設定。
- 更新不可のレガシー設定。

アカウント ID とロールを使用しないベアラ認証の使用については、「Amazon CodeCatalyst ユーザーガイド」の [AWS CLI 「で を使用するように設定 CodeCatalyst する」](#) を参照してください。

Note

AWS CLI コマンドで IAM Identity Center を使用するガイド付きプロセスについては、「」を参照してください [the section called “チュートリアル: AWS IAM Identity Center および Amazon S3”](#)。

トピック

- [the section called “前提条件”](#)
- [the section called “aws configure sso ウィザードでプロファイルを設定する”](#)
- [the section called “aws configure sso-session ウィザードで sso-session セクションのみを設定する”](#)

- [the section called “config ファイルを使用した手動設定”](#)
- [the section called “IAM Identity Center セッションにサインインする”](#)
- [the section called “IAM Identity Center プロファイルでコマンドを実行する”](#)
- [the section called “IAM Identity Center セッションからサインアウトする”](#)
- [the section called “トラブルシューティング”](#)
- [the section called “関連リソース”](#)

前提条件

- をインストールします AWS CLI。詳細については、「[the section called “インストール/更新”](#)」を参照してください。
- 最初に IAM Identity Center 内のSSO認証にアクセスする必要があります。AWS 認証情報にアクセスするには、次のいずれかの方法を選択します。

IAM Identity Center 経由でアクセスを確立していない

「AWS IAM Identity Center ユーザーガイド」の「[開始方法](#)」の手順に従います。このプロセスでは、IAM Identity Center をアクティブ化し、管理ユーザーを作成し、適切な最小権限のアクセス許可セットを追加します。

Note

最小権限のアクセス許可を適用するアクセス許可セットを作成します。雇用主がこの目的のためにカスタムアクセス許可セットを作成していない限り、定義済みの PowerUserAccess アクセス許可セットを使用することをお勧めします。

ポータルを終了し、再度サインインして、AWS アカウント、プログラムによるアクセスの詳細、Administrator または のオプションを確認します PowerUserAccess。を操作する PowerUserAccess ときに を選択します SDK。

雇用主が管理するフェデレーテッド ID プロバイダー (Azure AD や Okta など) AWS から に既にアクセスできる

ID プロバイダーのポータル AWS から にサインインします。Cloud Administrator がユーザー PowerUserAccess (開発者) にアクセス許可を付与している場合は、AWS アカウント アクセスで

きるとアクセス許可セットが表示されます。アクセス許可セットの名前の横に、そのアクセス許可セットを使用してアカウントに手動またはプログラムでアクセスするオプションが表示されます。

カスタム実装では、アクセス許可セット名が異なるなど、エクスペリエンスが異なる場合があります。どのアクセス許可セットを使用すればよいかわからない場合は、IT チームにお問い合わせください。

雇用主が管理する AWS アクセスポータル AWS から に既にアクセスできる

AWS アクセスポータル AWS から にサインインします。Cloud Administrator がユーザー PowerUserAccess (開発者) にアクセス許可を付与している場合は、AWS アカウント アクセスできるとアクセス許可セットが表示されます。アクセス許可セットの名前の横に、そのアクセス許可セットを使用してアカウントに手動またはプログラムでアクセスするオプションが表示されます。

雇用主が管理するフェデレーティッドカスタム ID プロバイダー AWS から に既にアクセスできるサポートについては、IT チームにお問い合わせください。

IAM Identity Center へのアクセスを取得したら、以下を実行して IAM Identity Center 情報を収集します。

1. AWS アクセスポータルで、開発に使用するアクセス許可セットを選択し、アクセスキーリンクを選択します。
2. 認証情報の取得ダイアログボックスで、オペレーティングシステムに一致するタブを選択します。
3. IAM Identity Center の認証情報方法を選択して、 の実行に必要な SSO Start URL と SSO Region 値を取得します `aws configure sso`。登録するスコープ値の詳細については、IAM 「Identity Center ユーザーガイド [OAuth](#)」の「[2.0 アクセススコープ](#)」を参照してください。

aws configure sso ウィザードでプロファイルを設定する

の IAM Identity Center プロファイルを設定するには AWS CLI :

1. 任意のターミナルで、`aws configure sso` コマンドを実行します。

IAM Identity Center

セッション名を作成し、IAM Identity Center の開始 URL、IAM Identity Center ディレクトリをホスト AWS リージョン する、および登録範囲を指定します。


```
$ aws configure sso
SSO session name (Recommended): my-sso
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]: us-east-1
SSO registration scopes [None]: sso:account:access
```

Legacy IAM Identity Center

セッション名をスキップし、IAMアイデンティティセンターの開始URLと、アイデンティティセンターディレクトリをホストする AWS リージョンを指定します。

```
$ aws configure sso
SSO session name (Recommended):
SSO start URL [None]: https://my-sso-portal.awsapps.com/start
SSO region [None]:us-east-1
```

2. は、IAMデフォルトのブラウザを開いて Identity Center アカウントのサインインプロセス AWS CLI を実行しようとします。このプロセスでは、データ AWS CLI へのアクセスを許可するように求められる場合があります。AWS CLI は SDK for Python 上に構築されているため、アクセス許可メッセージにはbotocore名前のバリエーションが含まれている場合があります。
 - がブラウザを開く AWS CLI ことができない場合、サインインプロセスを手動で開始する手順が表示されます。

```
If the browser does not open or you wish to use a
different device to
authorize this request, open the following URL:
https://device.sso.us-west-2.amazonaws.com/
Then enter the code:
QCFK-N451
```

3. 表示されたリストから使用する AWS アカウントを選択します。1つのアカウントのみを使用する権限がある場合、はそのアカウント AWS CLI を自動的に選択し、プロンプトをスキップします。

```
There are 2 AWS accounts available to you.
> DeveloperAccount, developer-account-admin@example.com (123456789011)
  ProductionAccount, production-account-admin@example.com (123456789022)
```

4. 表示されたリストから使用するIAMロールを選択します。使用できるロールが1つだけの場合、はそのロール AWS CLI を自動的に選択し、プロンプトをスキップします。


```
Using the account ID 123456789011
There are 2 roles available to you.
> ReadOnly
  FullAccess
```

5. [デフォルトの出力形式](#)、コマンドを送信する[デフォルト AWS リージョン](#)、および[プロファイルの名前](#)を指定します。プロファイル名defaultとしてを指定すると、このプロファイルが使用されるデフォルトプロファイルになります。次の例では、デフォルトのリージョン、デフォルトの出力形式、プロファイルの名前を入力します。

```
CLI default client Region [None]: us-west-2<ENTER>
CLI default output format [None]: json<ENTER>
CLI profile name [123456789011_ReadOnly]: my-dev-profile<ENTER>
```

6. 最後のメッセージは、完了したプロファイル設定について説明しています。このプロファイルを使用して認証情報をリクエストできるようになりました。aws sso login コマンドを使用して、コマンドを実行するために必要な認証情報をリクエストして取得します。手順については、[IAM Identity Center セッションにサインインする](#)を参照してください。

生成された設定ファイル

これらのステップにより、次のような sso-sessionセクションと名前付きプロファイルが config ファイル内に作成されます。

IAM Identity Center

```
[profile my-dev-profile]
sso_session = my-sso
sso_account_id = 123456789011
sso_role_name = readOnly
region = us-west-2
output = json

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Legacy IAM Identity Center

```
[profile my-dev-profile]  
sso_start_url = https://my-sso-portal.awsapps.com/start  
sso_region = us-east-1  
sso_account_id = 123456789011  
sso_role_name = readOnly  
region = us-west-2  
output = json
```

aws configure sso-session ウィザードで sso-session セクションのみを設定する

Note

この設定は、レガシー IAM Identity Center と互換性がありません。

aws configure sso-session コマンドは、`~/.aws/config` ファイル内の sso-session セクションを更新します。aws configure sso-session コマンドを実行して、IAM アイデンティティセンターの開始 URL と IAM、アイデンティティセンターディレクトリをホストする AWS リージョンを指定します。

```
$ aws configure sso-session  
SSO session name: my-sso  
SSO start URL [None]: https://my-sso-portal.awsapps.com/start  
SSO region [None]: us-east-1  
SSO registration scopes [None]: sso:account:access
```

config ファイルを使用した手動設定

IAM Identity Center の設定情報は config ファイルに保存され、テキストエディタを使用して編集できます。Identity IAM Center のサポートを名前付きプロファイルに手動で追加するには、キーと値を config ファイルに追加する必要があります。

IAM Identity Center 設定ファイル

config ファイルの `sso-session` セクションは、SSO アクセストークンを取得するための設定変数をグループ化するために使用されます。これにより、AWS 認証情報を取得できます。次の設定を使用します。

- (必須) [sso_start_url](#)
- (必須) [sso_region](#)
- [sso_account_id](#)
- [sso_role_name](#)
- [sso_registration_scopes](#)

`sso-session` セクションを定義し、プロファイルに関連付けます。 `sso_region` と `sso_start_url` の設定は、 `sso-session` セクション内で設定する必要があります。通常、 `sso_account_id` と `sso_role_name` を設定する必要があります。

次の例では、SSO 認証情報をリクエスト SDK するようにを設定し、トークンの自動更新をサポートしています。

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

これにより、`sso-session` 設定を複数のプロファイルでも再利用できるようになります。

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
sso_account_id = 111122223333
```

```
sso_role_name = SampleRole2

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

ただし、SSOトークン設定のすべてのシナリオで `sso_account_id` と `sso_role_name` が必要なのわけではありません。アプリケーションがベアラ認証をサポートする AWS サービスのみを使用している場合、従来の AWS 認証情報は必要ありません。ベアラ認証は、ベアラトークンと呼ばれるセキュリティトークンを使用する HTTP 認証スキームです。このシナリオでは、`sso_account_id` と `sso_role_name` は必須ではありません。AWS のサービスでベアラトークン認可をサポートしているかどうかについては、各サービスのガイドを参照してください。

さらに、登録スコープは `sso-session` の一部として設定できます。スコープは、アプリケーションのユーザーアカウントへのアクセスを制限するための OAuth 2.0 のメカニズムです。アプリケーションは 1 つ以上のスコープをリクエストでき、アプリケーションに発行されたアクセストークンは付与されたスコープに限定されます。これらのスコープは、登録された OIDC クライアントおよびクライアントによって取得されたアクセストークンに対して承認されるようにリクエストされたアクセス許可を定義します。次の例では、アカウント/ロールを一覧表示するアクセスを許可するように `sso_registration_scopes` を設定しています。

```
[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

認証トークンは、セッション名に基づいたファイル名を使用して、`sso/cache` ディレクトリの下にディスクにキャッシュされます。

レガシー IAM Identity Center 設定ファイル

Note

トークンの自動更新は、更新不可のレガシー設定ではサポートされていません。SSO トークン設定を使用することをお勧めします。

名前付きプロファイルに IAM Identity Center サポートを手動で追加するには、`config` ファイル内のプロファイル定義に次のキーと値を追加する必要があります。

- [sso_start_url](#)
- [sso_region](#)
- [sso_account_id](#)
- [sso_role_name](#)

ファイルには、有効な他のキーと値を含めることができます。aws/config。次の例は、IAM Identity Center プロファイルです。

```
[profile my-sso-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-west-2
sso_account_id = 111122223333
sso_role_name = SSOReadOnlyRole
region = us-west-2
output = json
```

コマンドを実行するには、まず一時的な認証情報 [the section called “IAM Identity Center セッションにサインインする”](#) をリクエストして取得する必要があります。

config ファイルと credentials ファイルの詳細については、「[the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)」を参照してください。

IAM Identity Center セッションにサインインする

Note

サインインプロセスでは、データ AWS CLI へのアクセスを許可するように求められる場合があります。AWS CLI は SDK for Python 上に構築されているため、アクセス許可メッセージには botocore 名前のバリエーションが含まれている場合があります。

Identity Center 認証情報のセットを取得してキャッシュするには、に対して次のコマンドを実行してデフォルトのブラウザ AWS CLI を開き、IAM Identity Center のログインを確認します。

```
$ aws sso login --profile my-dev-profile
SSO authorization page has automatically been opened in your default browser.
Follow the instructions in the browser to complete this authorization request.
Successfully logged into Start URL: https://my-sso-portal.awsapps.com/start
```

IAM Identity Center セッション認証情報はキャッシュされ、はそれら AWS CLI を使用して、プロフィールで指定されたIAMロールの AWS 認証情報を安全に取得します。

AWS CLI がブラウザを開くことができない場合

AWS CLI がブラウザを開くことができない場合、ブラウザを自分で開き、指定されたコードを入力するように求められます。

```
$ aws sso login --profile my-dev-profile
Using a browser, open the following URL:
https://device.sso.us-west-2.amazonaws.com/
and enter the following code:
QCFK-N451
```

aws sso login コマンドの --sso-session パラメータを使用してログインするときに、どの sso-session プロファイルを使用するかも指定できます。このオプションは sso-session、レガシー IAM Identity Center では使用できません。

```
$ aws sso login --sso-session my-dev-session
```

認証トークンは、に基づくファイル名で ~/.aws/sso/cache ディレクトリの下のディスクにキャッシュされます sso_start_url。

IAM Identity Center プロファイルでコマンドを実行する

ログインしたら、認証情報を使用して、関連付けられた名前付きプロフィールで AWS CLI コマンドを呼び出すことができます。次の例は、プロフィールを使用したコマンドを示しています。

```
$ aws sts get-caller-identity --profile my-dev-profile
```

IAM Identity Center にサインインし、キャッシュされた認証情報の有効期限が切れていない限り、は必要に応じて期限切れの AWS 認証情報 AWS CLI を自動的に更新します。ただし、IAM Identity Center 認証情報の有効期限が切れた場合は、IAM Identity Center アカウントに再度ログインして明示的に更新する必要があります。

IAM Identity Center セッションからサインアウトする

IAM Identity Center プロファイルの使用が完了したら、認証情報の有効期限を切れるようにするか、次のコマンドを実行してキャッシュされた認証情報を削除できます。

```
$ aws sso logout
Successfully signed out of all SSO profiles.
```

トラブルシューティング

を使用して問題が発生した場合は AWS CLI、トラブルシューティングの手順 [エラーのトラブルシューティング](#) については、「」を参照してください。

関連リソース

その他のリソースは次のとおりです。

- [the section called “IAM Identity Center の概念”](#)
- [the section called “チュートリアル: AWS IAM Identity Center および Amazon S3”](#)
- [the section called “インストール/更新”](#)
- [the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)
- [aws configure sso](#) AWS CLI バージョン 2 リファレンスの
- [aws configure sso-session](#) AWS CLI バージョン 2 リファレンスの
- [aws sso login](#) AWS CLI バージョン 2 リファレンスの
- [aws sso logout](#) AWS CLI バージョン 2 リファレンスの
- Amazon CodeCatalyst ユーザーガイドの [AWS CLI でを使用するようにを設定する CodeCatalyst](#)
- [OAuth 2.0 Identity Center ユーザーガイドのアクセススコープ IAM](#)
- IAM Identity Center [ユーザーガイドの入門チュートリアル](#)

AWS IAM Identity Center の概念 AWS CLI

このトピックでは、AWS IAM Identity Center (IAM Identity Center) の主要な概念について説明します。IAM Identity Center は、既存の ID プロバイダー (IdP) と統合することで、複数の AWS アカウント、アプリケーション SDKs、および ツールでのユーザーアクセス管理を簡素化するクラウドベースの IAM サービスです。これにより、一元化されたユーザーポータルを通じて安全なシングルサインオン、アクセス許可管理、監査が可能になり、組織のアイデンティティとアクセスガバナンスが合理化されます。

トピック

- [IAM Identity Center とは](#)

- [用語](#)
- [IAM Identity Center の仕組み](#)
- [追加リソース](#)

IAM Identity Center とは

IAM Identity Center は、クラウドベースのアイデンティティとアクセス管理 (IAM) サービスであり、複数の AWS アカウント およびビジネスアプリケーションへのアクセスを一元管理できます。

権限のあるユーザーが、既存の企業認証情報を使用して、アクセス許可を付与された AWS アカウント およびアプリケーションにアクセスできるユーザーポータルを提供します。これにより、組織は一貫したセキュリティポリシーを適用し、ユーザーアクセス管理を合理化できます。

使用する IdP に関係なく、IAM Identity Center はこれらの区別を抽象化します。例えば、ブログ記事 [IAM 「アイデンティティセンターの次の進化」](#) で説明されているように、Microsoft Azure AD を接続できます。

Note

アカウント ID とロールを使用しないベアラ認証の使用については、「Amazon CodeCatalyst ユーザーガイド」の [AWS CLI 「でを使用するように設定 CodeCatalystする」](#) を参照してください。

用語

IAM Identity Center を使用する際の一般的な用語は次のとおりです。

ID プロバイダー (IdP)

Identity Center、Microsoft Azure AD、Okta、または独自の企業ディレクトリサービスなどの IAM ID 管理システム。

AWS IAM Identity Center

IAM Identity Center は、AWS 所有の IdP サービスです。以前は AWS シングルサインオンと呼ばれていました SDKs が、ツールは下位互換性のために ssoAPI 名前空間を保持します。詳細については、「ユーザーガイド」の [IAM 「アイデンティティセンターの名前変更」](#) を参照してください。AWS IAM Identity Center

AWS アクセスポータル

承認された AWS アカウント、サービス、リソースにアクセスURLするための一意の IAM Identity Center。

フェデレーション

IAM Identity Center と ID プロバイダー間の信頼を確立してシングルサインオンを有効にするプロセス (SSO)。

AWS アカウント

を通じてユーザーにアクセス権を付与 AWS アカウント する AWS IAM Identity Center。

アクセス許可セット、AWS 認証情報、認証情報、sigv4 認証情報

ユーザーまたはグループに割り当てて、へのアクセスを許可できるアクセス許可の事前定義されたコレクション AWS のサービス。

登録スコープ、アクセススコープ、スコープ

スコープは、アプリケーションのユーザーアカウントへのアクセスを制限するための OAuth 2.0 のメカニズムです。アプリケーションは 1 つ以上のスコープをリクエストでき、アプリケーションに発行されたアクセストークンは付与されたスコープに限定されます。スコープの詳細については、IAM 「Identity Center ユーザーガイド」の [OAuth 「2.0 アクセススコープ」](#) を参照してください。

トークン、更新トークン、アクセストークン

トークンは、認証時に発行される一時的なセキュリティ認証情報です。これらのトークンには、ID と付与されたアクセス許可に関する情報が含まれています。

IAM Identity Center ポータルから AWS リソースまたはアプリケーションにアクセスすると、トークンは認証と認可 AWS のためにに提示されます。これにより AWS、はアイデンティティを検証し、リクエストされたアクションを実行するために必要なアクセス許可があることを確認します。

認証トークンは、セッション名に基づく JSON ファイル名で ~/.aws/sso/cache ディレクトリの下にディスクにキャッシュされます。

セッション

IAM Identity Center セッションとは、ユーザーが認証され、AWS リソースまたはアプリケーションへのアクセスが許可されている期間を指します。ユーザーが IAM Identity Center ポータ

ルにサインインすると、セッションが確立され、ユーザーのトークンは指定された期間有効です。セッション期間の設定の詳細については、AWS IAM Identity Center 「ユーザーガイド」の「[セッション期間の設定](#)」を参照してください。

セッション中、セッションがアクティブである限り、再認証することなく、異なる AWS アカウントとアプリケーション間を移動できます。セッションの有効期限が切れたら、再度サインインしてアクセスを更新します。

IAM Identity Center セッションは、ユーザーアクセス認証情報の有効性を制限することで、シームレスなユーザーエクスペリエンスを提供すると同時に、セキュリティのベストプラクティスを適用するのに役立ちます。

IAM Identity Center の仕組み

IAM Identity Center は、IAM Identity Center、Microsoft Azure AD、Okta などの組織の ID プロバイダーと統合されます。ユーザーはこの ID プロバイダーに対して認証され、IAM Identity Center はこれらの ID を環境内の適切なアクセス許可とアクセスにマッピングします AWS。

次の IAM Identity Center ワークフローは、IAM Identity Center を使用する AWS CLI ように を既に設定していることを前提としています。

1. 任意のターミナルで、`aws sso login` コマンドを実行します。
2. にサインイン AWS アクセスポータルして、新しいセッションを開始します。
 - 新しいセッションを開始すると、キャッシュされる更新トークンとアクセストークンを受け取ります。
 - 既にアクティブなセッションがある場合、既存のセッションは再利用され、既存のセッションの有効期限が切れると期限切れになります。
3. `config` ファイルで設定したプロファイルに基づいて、IAM Identity Center は適切なアクセス許可セットを引き受け、関連する AWS アカウント および アプリケーションへのアクセスを許可します。
4. AWS CLI、SDKs、および Tools は、引き受けた IAM ロールを使用して、セッションの有効期限が切れるまで Amazon S3 バケットを作成する AWS のサービスなど、を呼び出します。
5. IAM Identity Center のアクセストークンは 1 時間ごとにチェックされ、更新トークンを使用して自動的に更新されます。
 - アクセストークンの有効期限が切れている場合、SDK または ツールは更新トークンを使用して新しいアクセストークンを取得します。次に、これらのトークンのセッション期間を比較し、

更新トークンの有効期限が切れていない場合は、IAM Identity Center が新しいアクセストークンを提供します。

- 更新トークンの有効期限が切れている場合、新しいアクセストークンは提供されず、セッションは終了しました。
6. セッションは、更新トークンの有効期限が切れるか、`aws sso logout` コマンドを使用して手動でログアウトすると終了します。キャッシュされた認証情報は削除されます。IAM Identity Center を使用してサービスへのアクセスを継続するには、`aws sso login` コマンドを使用して新しいセッションを開始する必要があります。

追加リソース

その他のリソースは次のとおりです。

- [the section called “IAM Identity Center 認証”](#)
- [the section called “チュートリアル: AWS IAM Identity Center および Amazon S3”](#)
- [the section called “インストール/更新”](#)
- [the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)
- [aws configure sso](#) AWS CLI バージョン 2 リファレンスの
- [aws configure sso-session](#) AWS CLI バージョン 2 リファレンスの
- [aws sso login](#) AWS CLI バージョン 2 リファレンスの
- [aws sso logout](#) AWS CLI バージョン 2 リファレンスの
- Amazon CodeCatalyst ユーザーガイドの [AWS CLI でを使用するようにを設定する CodeCatalyst](#)
- AWS IAM Identity Center ユーザーガイドの [IAM Identity Center の名前変更](#)
- [OAuth 2.0 Identity Center ユーザーガイドのアクセススコープ IAM](#)
- AWS IAM Identity Center ユーザーガイドで [セッション期間を設定する](#)
- IAM Identity Center [ユーザーガイドの入門チュートリアル](#)

チュートリアル: IAM Identity Center を使用して Amazon S3 コマンドを実行する AWS CLI

このトピックでは、AWS IAM Identity Center (IAMAmazon S3) の AWS Command Line Interface () コマンドを実行する認証情報を取得するために、現在の Amazon Simple Storage Service (Identity Center AWS CLI) でユーザーを AWS CLI 認証するようにを設定する方法について説明します。

トピック

- [ステップ 1: IAM Identity Center での認証](#)
- [ステップ 2: IAM Identity Center 情報を収集する](#)
- [ステップ 3: Amazon S3 バケットを作成する](#)
- [ステップ 4: をインストールする AWS CLI](#)
- [ステップ 6: プロファイルを設定する AWS CLI](#)
- [ステップ 7: IAM Identity Center にログインする](#)
- [ステップ 8: Amazon S3 コマンドを実行する](#)
- [ステップ 9: IAM Identity Center からログアウトする](#)
- [ステップ 10: リソースをクリーンアップする](#)
- [トラブルシューティング](#)
- [追加リソース](#)

ステップ 1: IAM Identity Center での認証

IAM Identity Center 内のSSO認証にアクセスします。AWS 認証情報にアクセスするには、次のいずれかの方法を選択します。

IAM Identity Center 経由でアクセスを確立していない

「AWS IAM Identity Center ユーザーガイド」の「[開始方法](#)」の手順に従います。このプロセスでは、IAM Identity Center をアクティブ化し、管理ユーザーを作成し、適切な最小権限のアクセス許可セットを追加します。

Note

最小権限のアクセス許可を適用するアクセス許可セットを作成します。雇用主がこの目的のためにカスタムアクセス許可セットを作成していない限り、定義済みの PowerUserAccess アクセス許可セットを使用することをお勧めします。

ポータルを終了し、再度サインインして、AWS アカウント、プログラムによるアクセスの詳細、Administrator または のオプションを確認します PowerUserAccess。 を操作する PowerUserAccess ときに を選択します SDK。

雇用主が管理するフェデレーテッド ID プロバイダー (Azure AD や Okta など) AWS からに既にアクセスできる

ID プロバイダーのポータル AWS からにサインインします。Cloud Administrator がユーザー PowerUserAccess (開発者) にアクセス許可を付与している場合は、AWS アカウント アクセスできるとアクセス許可セットが表示されます。アクセス許可セットの名前の横に、そのアクセス許可セットを使用してアカウントに手動またはプログラムでアクセスするオプションが表示されます。

カスタム実装では、アクセス許可セット名が異なるなど、エクスペリエンスが異なる場合があります。どのアクセス許可セットを使用すればよいかわからない場合は、IT チームにお問い合わせください。

雇用主が管理する AWS アクセスポータル AWS からに既にアクセスできる

AWS アクセスポータル AWS からにサインインします。Cloud Administrator がユーザーに PowerUserAccess (開発者) アクセス許可を付与している場合は、AWS アカウント アクセスできるとアクセス許可セットが表示されます。アクセス許可セットの名前の横に、そのアクセス許可セットを使用してアカウントに手動またはプログラムでアクセスするオプションが表示されます。

雇用主が管理するフェデレーテッドカスタム ID プロバイダー AWS からに既にアクセスできるサポートについては、IT チームにお問い合わせください。

ステップ 2: IAM Identity Center 情報を収集する

へのアクセスを取得したら AWS、以下を実行して IAM Identity Center 情報を収集します。

1. AWS アクセスポータルで、開発に使用するアクセス許可セットを選択し、アクセスキーリンクを選択します。
2. 認証情報の取得ダイアログボックスで、オペレーティングシステムに一致するタブを選択します。
3. IAM Identity Center の認証情報方法を選択して、の実行に必要な SSO Start URL と SSO Region 値を取得します `aws configure sso`。登録するスコープ値の詳細については、IAM 「Identity Center ユーザーガイド」の [OAuth 「2.0 アクセススコープ」](#) を参照してください。

ステップ 3: Amazon S3 バケットを作成する

にサインイン AWS Management Console し、で Amazon S3 コンソールを開きます <https://console.aws.amazon.com/s3/>。

このチュートリアルでは、後でリストで取得するいくつかのバケットを作成します。

ステップ 4: をインストールする AWS CLI

オペレーティングシステムの AWS CLI 以下の手順をインストールします。詳細については、「[the section called “インストール/更新”](#)」を参照してください。

インストールしたら、希望するターミナルを開き、次のコマンドを実行してインストールを確認できます。これにより、 のインストール済みバージョンが表示されます AWS CLI。

```
$ aws --version
```

ステップ 6: プロファイルを設定する AWS CLI

次のいずれかの方法を使用してプロファイルを設定する

aws configure sso ウィザードでプロファイルを設定する

config ファイルの sso-session セクションは、SSO アクセストークンを取得するための設定変数をグループ化するために使用されます。これにより、AWS 認証情報を取得できます。次の設定を使用します。

- (必須) [sso_start_url](#)
- (必須) [sso_region](#)
- [sso_account_id](#)
- [sso_role_name](#)
- [sso_registration_scopes](#)

sso-session セクションを定義し、プロファイルに関連付けます。sso_region および sso_start_url 設定は、sso-session セクション内で設定する必要があります。通常、 が SSO 認証情報を SDK リクエストできるように、profile セクションで sso_account_id と を設定 sso_role_name する必要があります。

次の例では、SSO 認証情報をリクエスト SDK するように を設定し、トークンの自動更新をサポートしています。

```
[profile dev]
sso_session = my-sso
```

```
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

config ファイルを使用した手動設定

config ファイルの sso-session セクションは、SSO アクセストークンを取得するための設定変数をグループ化するために使用されます。これにより、AWS 認証情報を取得できます。次の設定を使用します。

- (必須) [sso_start_url](#)
- (必須) [sso_region](#)
- [sso_account_id](#)
- [sso_role_name](#)
- [sso_registration_scopes](#)

sso-session セクションを定義してプロファイルに関連付けます。sso_region と sso_start_url は sso-session セクション内に設定する必要があります。通常、が SSO 認証情報を SDK リクエストできるように、profile セクションで sso_account_id とを設定 sso_role_name する必要があります。

次の例では、SSO 認証情報をリクエスト SDK するようにを設定し、トークンの自動更新をサポートしています。

```
[profile my-dev-profile]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

認証トークンは、セッション名に基づいたファイル名を使用して、~/.aws/sso/cache ディレクトリの下のディスクにキャッシュされます。

ステップ 7: IAM Identity Center にログインする

Note

サインインプロセスでは、データ AWS CLI へのアクセスを許可するように求められる場合があります。AWS CLI は SDK for Python 上に構築されているため、アクセス許可メッセージには `botocore` 名前のバリエーションが含まれている場合があります。

IAM Identity Center の認証情報を取得してキャッシュするには、AWS CLI の次のコマンドを実行してデフォルトのブラウザを開き、IAM Identity Center のログインを確認します。

```
$ aws sso login --profile my-dev-profile
```

ステップ 8: Amazon S3 コマンドを実行する

前に作成したバケットを一覧表示するには、[aws s3 ls](#) コマンドを使用します。次の例では、すべての Amazon S3 バケットを一覧表示します。

```
$ aws s3 ls
2018-12-11 17:08:50 my-bucket
2018-12-14 14:55:44 my-bucket2
```

ステップ 9: IAM Identity Center からログアウトする

IAM Identity Center プロファイルの使用が完了したら、次のコマンドを実行してキャッシュされた認証情報を削除します。

```
$ aws sso logout
Successfully signed out of all SSO profiles.
```

ステップ 10: リソースをクリーンアップする

このチュートリアルが完了したら、Amazon S3 バケットなど、このチュートリアル中に作成した不要になったリソースをクリーンアップします。

トラブルシューティング

を使用して問題が発生した場合は AWS CLI、一般的なトラブルシューティング手順 [エラーのトラブルシューティング](#) については、「」を参照してください。

追加リソース

その他のリソースは次のとおりです。

- [the section called “IAM Identity Center の概念”](#)
- [the section called “IAM Identity Center 認証”](#)
- [the section called “インストール/更新”](#)
- [the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)
- [aws configure sso](#) AWS CLI バージョン 2 リファレンスの
- [aws configure sso-session](#) AWS CLI バージョン 2 リファレンスの
- [aws sso login](#) AWS CLI バージョン 2 リファレンスの
- [aws sso logout](#) AWS CLI バージョン 2 リファレンスの
- Amazon CodeCatalyst ユーザーガイドの [AWS CLI でを使用するようにを設定する CodeCatalyst](#)
- [OAuth 2.0 Identity Center ユーザーガイドのアクセススコープ IAM](#)
- IAM Identity Center ユーザーガイドの [入門チュートリアル](#)

の短期認証情報による認証 AWS CLI

セッション期間の延長オプションで [IAM Identity Center 認証](#)を使用するように SDKまたは ツールを設定することをお勧めします。ただし、AWS アクセスポータルで利用可能な一時的な認証情報をコピーして使用できます。有効期限が切れたら、新しい認証情報をコピーする必要があります。一時的な認証情報は、プロファイルで使用することも、システムプロパティや環境変数の値として使用することもできます。

1. [AWS アクセスポータルにサインインします。](#)
2. アクセス AWS ポータルからIAMロール認証情報をコピーするには、[次の手順に従います。](#)
 1. リンクされた手順のステップ 2 では、開発ニーズに応じたアクセスを許可する AWS アカウントとIAMロール名を選択します。このロールには通常、PowerUserAccessまたは Developer などの名前があります。
 2. ステップ 4 で、[AWS 認証情報ファイルにプロファイルを追加] オプションを選択し、内容をコピーします。
3. 共有 credentials ファイルを作成するか、開きます。このファイルは、`~/.aws/credentialsLinux`およびmacOSシステム、および`%USERPROFILE%\aws`

- [クロスアカウントロールと外部 ID](#)
- [監査を容易にするためのロールセッション名の指定](#)
- [ウェブ ID を使用したロールの継承](#)
- [キャッシュされた認証情報のクリア](#)

前提条件

これらの iam コマンドを使用するには、AWS CLIをインストールして設定する必要があります。詳細については、「[the section called “インストール/更新”](#)」を参照してください。

IAM ロールの使用の概要

AWS Command Line Interface (AWS CLI) は、`~/.aws/config` ファイル内のIAMロールのプロファイルを定義することで、ロールを使用するように設定できます。

次の例は marketingadmin という名前のロールプロファイルを示しています。コマンドを `awscli` で実行する `--profile marketingadmin` (または [AWS_PROFILE環境変数](#) で指定する) 場合、は別のプロファイルで定義された認証情報 AWS CLI を使用して user1、Amazon リソースネーム (ARN) を持つロールを引き受け `arn:aws:iam::123456789012:role/marketingadminrole`。このロールに割り当てられたアクセス権限で許可される任意のオペレーションを実行することができます。

```
[profile marketingadmin]  
role_arn = arn:aws:iam::123456789012:role/marketingadminrole  
source_profile = user1
```

その後、このロールを使用するアクセス許可があるユーザー認証情報を含む、別の名前付きプロファイルを示す `source_profile` を指定できます。前の例では、marketingadmin プロファイルは user1 プロファイル内の認証情報を使用しています。AWS CLI コマンドがプロファイル を使用するように指定すると marketingadmin、はリンクされた user1 プロファイルの認証情報 AWS CLI を自動的に検索し、それらを使用して指定された IAM ロールの一時的な認証情報をリクエストします。CLI は、バックグラウンドで [sts:AssumeRole](#) オペレーションを使用してこれを実行します。これらの一時的な認証情報は次に、リクエストされた AWS CLI コマンドを実行するために使用されます。指定されたロールには、リクエストされた AWS CLI コマンドの実行を許可する IAM アクセス許可ポリシーがアタッチされている必要があります。

Amazon Elastic Compute Cloud (Amazon EC2) インスタンスまたは Amazon Elastic Container Service (Amazon ECS) コンテナ内から AWS CLI コマンドを実行するには、インスタンスプロ

ファイルまたはコンテナにアタッチされたIAMロールを使用できます。プロファイルを指定しない、または環境変数を設定しない場合、そのロールが直接使用されます。これにより、存続期間の長いアクセスキーをインスタンスで保存することを回避できます。これらのインスタンスまたはコンテナのロールは、別のロールの認証情報を取得するためにのみ使用できます。これを行うには、`credential_source` (`source_profile` の代わりに) を使用して、認証情報を検索する方法を指定します。`credential_source` 属性では、以下の値がサポートされます。

- `Environment` - 環境変数からソース認証情報の取得。
- `Ec2InstanceMetadata` - Amazon EC2インスタンスプロファイルにアタッチされたIAMロールを使用します。
- `EcsContainer` - Amazon ECSコンテナにアタッチされたIAMロールを使用します。

次の例は、Amazon EC2インスタンスプロファイルを参照する際に使用するのと同じ`marketingadminrole`ロールを示しています。

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
```

ロールを呼び出すとき、多要素認証や外部 ID (サードパーティー企業がクライアントのリソースにアクセスするために使用する) の使用などを必須とする追加オプションがあります。また、AWS CloudTrail ログでより簡単に監査できる一意のロールセッション名を指定することもできます。

ロールの設定と使用

IAM ロールを指定するプロファイルを使用してコマンドを実行すると、AWS CLI はソースプロファイルの認証情報を使用して AWS Security Token Service (AWS STS) を呼び出し、指定されたロールの一時的な認証情報をリクエストします。ソースプロファイルのユーザーは、指定されたプロファイルのロール用の `sts:assume-role` を呼び出すアクセス許可を持っている必要があります。ロールには、ソースプロファイルのユーザーがこのロールを使用できる信頼関係が必要です。ロールの一時的な認証情報を取得して使用するプロセスを、一般にロールを引き受けると呼びます。

AWS Identity and Access Management 「ユーザーガイド」の「[アクセス許可をIAMユーザーに委任するロールの作成](#)」の手順に従って、[ユーザーに引き受けさせるアクセス許可IAMを持つロール](#)を作成できます。ロールとソースプロファイルのユーザーが同じアカウントに存在する場合、ロールの信頼関係を設定するときに、独自のアカウント ID を入力することができます。

ロールを作成した後、ユーザーが引き受けられることを許可するように信頼関係を変更します。

次の例では、ロールにアタッチできる信頼ポリシーを示します。このポリシーは、アカウント「123456789012」の任意のユーザーがロールを引き受けることを許可します (そのアカウントの管理者が明示的にユーザーに「sts:AssumeRole」のアクセス許可を付与した場合)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

信頼ポリシーは、実際にはアクセス許可を付与しません。アカウントの管理者は、適切なアクセス許可を持つポリシーをアタッチすることによって、ロールを引き受けるアクセス許可を個々のユーザーに委任する必要があります。次の例では、ユーザーに付与を行い、ユーザーが marketingadminrole ロールのみを引き受けることを許可するポリシーを示しています。ロールを引き受けるためのユーザーアクセス許可の付与の詳細については、IAM ユーザーガイドの [「ロールを切り替えるためのユーザーアクセス許可の付与」](#) を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

ユーザーは、ロールプロファイルを使用して AWS CLI コマンドを実行するための追加のアクセス許可を持つ必要はありません。代わりに、コマンドを実行するためのアクセス権限は、ロールにアタッチされたアクセス権限によって提供されます。アクセス許可ポリシーをロールにアタッチして、どのアクションをどの AWS リソースに対して実行できるかを指定します。ロールへのアクセス許可の

タッチ (ユーザーと同じ動作) の詳細については、IAM 「ユーザーガイド」の [IAM 「ユーザーのアクセス許可の変更」](#) を参照してください。

ロールプロファイル、ロールのアクセス許可、ロールの信頼関係およびユーザーアクセス許可が正しく設定されたので、コマンドラインで `--profile` オプションを呼び出してロールを使用できます。例えば、以下の例では、このトピックの冒頭で定義された `ls` ロールにアタッチされたアクセス権限を使用して Amazon S3 `marketingadmin` コマンドを呼び出します。

```
$ aws s3 ls --profile marketingadmin
```

いくつかの呼び出しにロールを使用するには、コマンドラインから、現在のセッションに対して `AWS_PROFILE` 環境変数を設定することができます。この環境変数が定義されている場合、各コマンドで `--profile` オプションを指定する必要はありません。

Linux または macOS

```
$ export AWS_PROFILE=marketingadmin
```

Windows

```
C:\> setx AWS_PROFILE marketingadmin
```

ユーザーとロールの設定の詳細については、IAM 「ユーザーガイドIAM」の [「アイデンティティ \(ユーザー、ユーザーグループ、ロール\) とIAMロール」](#) を参照してください。

多要素認証を使用する

セキュリティを強化するために、ロールプロファイルを使用して通話を行おうとするときに、多要素認証 (MFA) デバイス、U2F デバイス、またはモバイルアプリから生成されたワンタイムキーをユーザーに指定するよう要求できます。

まず、を必要とするIAMロールの信頼関係を変更することができますMFA。これにより、を使用して最初に認証することなく、誰もロールを使用できなくなりますMFA。例として、次の例の `Condition` 行を参照してください。このポリシーでは、という名前のユーザーがポリシーがアタッチされているロールを引き受けanikaることを許可します。ただし、を使用して認証する場合に限りますMFA。

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Sid": "",
  "Effect": "Allow",
  "Principal": { "AWS": "arn:aws:iam::123456789012:user/anika" },
  "Action": "sts:AssumeRole",
  "Condition": { "Bool": { "aws:multifactorAuthPresent": true } }
}
]
```

次に、ユーザーのMFAデバイスの を指定する行ARNをロールプロファイルに追加します。次のサンプル config ファイルエントリでは、2つのロールプロファイルを示しています。どちらもユーザー anika のアクセスキーを使用してロール cli-role の一時的な認証情報をリクエストします。ユーザー anika には、ロールの信頼ポリシーによって付与されたロールを引き受けるためのアクセス権限があります。

```
[profile role-without-mfa]
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile=cli-user

[profile role-with-mfa]
region = us-west-2
role_arn= arn:aws:iam::128716708097:role/cli-role
source_profile = cli-user
mfa_serial = arn:aws:iam::128716708097:mfa/cli-user

[profile cli-user]
region = us-west-2
output = json
```

mfa_serial 設定ではARN、図に示すように、またはハードウェアMFAトークンのシリアル番号を取得できます。

最初のプロファイル には role-without-mfaは必要ありませんMFA。ただし、ロールにアタッチされた前の信頼ポリシーの例には が必要なためMFA、このプロファイルでコマンドを実行しようとすると失敗します。

```
$ aws iam list-users --profile role-without-mfa
```

```
An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```


2 番目のプロファイルエントリは `role-with-mfa`、使用する MFA デバイスを識別します。ユーザーがこのプロファイルで AWS CLI コマンドを実行しようとする、は、MFA デバイスが提供するワンタイムパスワード (OTP) を入力するようにユーザーに AWS CLI 促します。MFA 認証が成功すると、コマンドはリクエストされたオペレーションを実行します。OTP は画面に表示されません。

```
$ aws iam list-users --profile role-with-mfa
Enter MFA code for arn:aws:iam::123456789012:mfa/cli-user:
{
  "Users": [
    {
      ...
    }
  ]
}
```

クロスアカウントロールと外部 ID

クロスアカウントロールとしてロールを設定することにより、ユーザーが別のアカウントに属しているロールを使用できるようにすることができます。ロールの作成中に、ロールタイプを別の AWS アカウントに設定します。詳細については、「[ロールを作成して IAM ユーザーにアクセス許可を委任する](#)」を参照してください。必要に応じて、を要求する MFA を選択します。で説明されているように、信頼関係で適切な条件を設定する必要があります [MFA 多要素認証を使用する](#)。

[外部 ID](#) を使用して、複数のアカウント間でロールを使用できるユーザーをさらに制御する場合は、ロールプロファイルにも `external_id` パラメータを追加する必要があります。これは通常、もう一方のアカウントが社外または組織外のユーザーによって制御される場合にのみ使用します。

```
[profile crossaccountrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
mfa_serial = arn:aws:iam::123456789012:mfa/saanvi
external_id = 123456
```

監査を容易にするためのロールセッション名の指定

ロールが多くの個人間で共有されると、監査はより難しくなります。そこで、呼び出された各オペレーションを、アクションを呼び出した個人に関連付けることが必要になります。ただし、個人がロールを使用する場合、個人によるロールの引き受けはオペレーションの呼び出しとは別のアクションであるため、この 2 つを手動で相互に関連付ける必要があります。

ユーザーがロールを引き受けるときに一意的なロールセッション名を指定すれば、この手順を簡素化できます。これを行うには、ロールを指定する `role_session_name` ファイルの各名前付きプロフ

イルに `config` パラメータを追加します。 `role_session_name` 値は `AssumeRole` オペレーションに渡され、ARN ロールセッションの一部になります。また、ログに記録されたすべてのオペレーションの AWS CloudTrail ログにも含まれます。

例えば、次のようにロールベースのプロファイルを作成できます。

```
[profile namedsessionrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
role_session_name = Session_Maria_Garcia
```

これにより、ロールセッションには次の `ARN` があります。

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

また、すべての AWS CloudTrail ログには、各オペレーションでキャプチャされた情報にロールセッション名が含まれます。

ウェブ ID を使用したロールの継承

[ウェブ ID フェデレーションと Open ID Connect \(OIDC\)](#) を使用して、 `role_arn` を引き受け AWS CLI する必要がありますを示すプロファイルを設定できます。プロファイルでこれを指定すると、AWS CLI は自動的に対応する AWS STS `AssumeRoleWithWebIdentity` 呼び出しを行います。

Note

IAM ロールを使用するプロファイルを指定すると、は一時的な認証情報を取得するために適切な呼び出し AWS CLI を行います。これらの認証情報は `~/.aws/cli/cache` に保存されます。同じプロファイルを指定する後続の AWS CLI コマンドは、キャッシュされた一時的な認証情報を有効期限まで使用します。その時点で、は一時的な認証情報 AWS CLI を自動的に更新します。

ウェブ ID フェデレーションを使用して一時的な認証情報を取得、使用するには、共有プロファイルで以下の設定値を指定します。

`role_arn`

引き受けるロールARNの `role_arn` を指定します。

web_identity_token_file

ID プロバイダーによって提供される OAuth 2.0 アクセストークンまたは OpenID Connect ID トークンを含むファイルへのパスを指定します。AWS CLI はこのファイルをロードし、その内容を `WebIdentityToken` 引数として `AssumeRoleWithWebIdentity` オペレーションに渡します。

[role_session_name](#)

このロール継承セッションに適用されるオプションの名前を指定します。

ウェブ ID を使用したロールの継承プロファイルの設定に必要な最小限の設定例を次に示します。

```
# In ~/.aws/config

[profile web-identity]
role_arn=arn:aws:iam:123456789012:role/RoLeNameToAssume
web_identity_token_file=/path/to/a/token
```

この設定は、[環境変数](#)を使用して提供することもできます。

AWS_ROLE_ARN

引き受けるロールARNの。

AWS_WEB_IDENTITY_TOKEN_FILE

ウェブ ID トークンファイルへのパス。

AWS_ROLE_SESSION_NAME

このロール継承セッションで適用される名前です。

Note

これらの環境変数は、現在、ウェブ ID プロバイダーのロールを継承する場合にのみ適用されます。これらは、`AssumeRole` プロバイダーの設定には適用されません。

キャッシュされた認証情報のクリア

ロールを使用すると、は一時的な認証情報を有効期限が切れるまでローカルに AWS CLI キャッシュします。次回使用しようとする、はユーザーに代わって更新 AWS CLI を試みます。

ロールの一時的な認証情報が[取り消された場合](#)、それらは自動的に更新されず、使用しようとする、と失敗します。ただし、キャッシュを削除して、に新しい認証情報の取得 AWS CLI を強制できます。

Linux または macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\aws\cli\cache
```

のIAMユーザー認証情報を使用した認証 AWS CLI

Warning

セキュリティリスクを回避するため、専用ソフトウェアの開発時や実際のデータの使用时に、認証にIAMユーザーを使用しないでください。代わりに、[AWS IAM Identity Center](#) などの ID プロバイダーとのフェデレーションを使用してください。

このセクションでは、IAMユーザーで基本設定を設定する方法について説明します。これらには、config ファイルと credentials ファイルを使用したセキュリティ認証情報が含まれます。代わりに AWS IAM Identity Center の設定手順を参照するには、「[the section called “IAM Identity Center 認証”](#)」を参照してください。

トピック

- [ステップ 1: IAM ユーザーを作成する](#)
- [ステップ 2: アクセスキーを取得する](#)
- [の設定 AWS CLI](#)
 - [aws configure の使用](#)
 - [.CSV ファイルを介したアクセスキーのインポート](#)

- [config ファイルと credentials ファイルの直接編集](#)

ステップ 1: IAM ユーザーを作成する

IAM ユーザーガイドのIAM [IAM「ユーザーの作成 \(コンソール\)」](#) 手順に従ってユーザーを作成します。

- [アクセス許可オプション] で、このユーザーにアクセス許可を割り当てる方法として [ポリシーを直接アタッチする] を選択します。
- ほとんどの「Getting Started SDK」チュートリアルでは、Amazon S3 サービスを例として使用しています。アプリケーションに Amazon S3 へのフルアクセスを提供するには、このユーザーにアタッチする AmazonS3FullAccess ポリシーを選択します。

ステップ 2: アクセスキーを取得する

1. にサインイン AWS Management Console し、でIAMコンソールを開きます <https://console.aws.amazon.com/iam/>。
2. IAM コンソールのナビゲーションペインで、ユーザーを選択し、以前に作成した **User name** ユーザーの を選択します。
3. ユーザーのページで、[セキュリティ認証情報] ページを選択します。次に、[アクセスキー] で [アクセスキーの作成] を選択します。
4. アクセスキーの作成ステップ 1 で、コマンドラインインターフェイス (CLI) を選択します。
5. [アクセスキーの作成ステップ 2] で、オプションのタグを入力して [次へ] を選択します。
6. アクセスキーの作成ステップ 3 では、.csv ファイルのダウンロードを選択して、IAMユーザーのアクセスキーとシークレットアクセスキーを含む.csvファイルを保存します。この情報は後で必要になります。
7. [完了] を選択します。

の設定 AWS CLI

一般的な使用のために、には以下の情報 AWS CLI が必要です。

- アクセスキー ID
- シークレットアクセスキー

- AWS リージョン
- 出力形式

は、この情報を credentials ファイル内の という名前のプロファイル (設定のコレクション) default に AWS CLI 保存します。デフォルトでは、このプロファイルの情報は、使用するプロファイルを明示的に指定しない AWS CLI コマンドを実行するときに使用されます。credentials ファイルの詳細については、「[の設定と認証情報ファイルの設定 AWS CLI](#)」を参照してください。

を設定するには AWS CLI、次のいずれかの手順を使用します。

トピック

- [aws configure の使用](#)
- [.CSV ファイルを介したアクセスキーのインポート](#)
- [config ファイルと credentials ファイルの直接編集](#)

aws configure の使用

一般的な用途では、aws configure コマンドは AWS CLI インストールをセットアップする最も速い方法です。この設定ウィザードでは、開始するために必要な各情報を入力するよう求められます。--profile オプションを使用して特に指定しない限り、はこの情報を default プロファイルに AWS CLI 保存します。

次の例では、サンプル値を使用して default プロファイルを設定します。次のセクションで説明するように、これらの値を独自の値に置き換えます。

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

次の例では、サンプル値を使用して userprod という名前のプロファイルを設定します。次のセクションで説明するように、これらの値を独自の値に置き換えます。

```
$ aws configure --profile userprod
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
```

```
Default output format [None]: json
```

.CSV ファイルを介したアクセスキーのインポート

`aws configure` を使用してアクセスキーを入力する代わりに、アクセスキーを作成した後にダウンロードしたプレーンテキスト `.csv` ファイルをインポートできます。

`.csv` ファイルには、次のヘッダーが含まれている必要があります。

- ユーザー名 - この列は `.csv` に追加する必要があります。これは、インポート時に `config` および `credentials` ファイルで使用されるプロファイル名を作成するために使用されます。
- アクセスキー ID
- シークレットアクセスキー

Note

最初のアクセスキーの作成時に、`[csv ファイルをダウンロード]` ダイアログボックスを閉じると、その後はシークレットアクセスキーにアクセスできなくなります。`.csv` ファイルが必要な場合は、必要なヘッダーと保存したキーペア情報を使用して自分でファイルを作成する必要があります。アクセスキー情報にアクセスできない場合は、新しいアクセスキーを作成する必要があります。

`.csv` ファイルをインポートするには、次のような `--csv` オプションで `aws configure import` コマンドを使用します。

```
$ aws configure import --csv file://credentials.csv
```

詳細については、「[aws_configure_import](#)」を参照してください。

config ファイルと credentials ファイルの直接編集

`config` ファイルと `credentials` ファイルを直接編集するには、次の操作を行います。

1. 共有 AWS `credentials` ファイルを作成するか、開きます。このファイルは、`~/.aws/credentialsLinux` および `macOS` システム、および `%USERPROFILE%\aws\credentialsWindows` 上にあります。詳細については、「[the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)」を参照してください。

- 共有 credentials ファイルに次のテキストを追加します。以前にダウンロードした .csv ファイル内のサンプル値を置き換え、ファイルを保存します。

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Amazon EC2 インスタンスメタデータを の認証情報として使用する AWS CLI

Amazon Elastic Compute Cloud (Amazon EC2) インスタンス内で AWS CLI から を実行すると、コマンドへの認証情報の提供を簡素化できます。各 Amazon EC2 インスタンスには、AWS CLI が一時的な認証情報を直接クエリできるメタデータが含まれています。IAM ロールがインスタンスにアタッチされると、 はインスタンスメタデータから認証情報 AWS CLI を自動的に安全に取得します。

このサービスを無効にするには、[AWS_EC2_METADATA_DISABLED](#) 環境変数を使用します。

トピック

- [前提条件](#)
- [Amazon EC2メタデータのプロファイルの設定](#)

前提条件

で Amazon EC2 認証情報を使用するには AWS CLI、以下を完了する必要があります。

- AWS CLI をインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- 設定ファイルおよび名前付きプロファイルについて理解します。詳細については、「[の設定と認証情報ファイルの設定 AWS CLI](#)」を参照してください。
- 必要なリソースにアクセスできる AWS Identity and Access Management (IAM) ロールを作成し、起動時にそのロールを Amazon EC2 インスタンスにアタッチしました。詳細については、IAM 「[Amazon ユーザーガイド EC2](#)」の「Amazon ポリシー」と IAM 「[ユーザーガイド](#)」の「[Amazon EC2 インスタンスで実行される AWS リソースへのアクセス許可アプリケーションを付与する](#)」を参照してください。 EC2

Amazon EC2メタデータのプロファイルの設定

ホストする Amazon EC2 インスタンスプロファイルで使用できる認証情報を使用するように指定するには、設定ファイルの名前付きプロファイルで次の構文を使用します。詳細な手順については、以下のステップを参照してください。

```
[profile profilename]  
role_arn = arn:aws:iam::123456789012:role/rolename  
credential_source = Ec2InstanceMetadata  
region = region
```

1. 設定ファイルにプロファイルを作成します。

```
[profile profilename]
```

2. 必要なリソースにアクセスできる IAM arn ロールを追加します。

```
role_arn = arn:aws:iam::123456789012:role/rolename
```

3. 認証情報のソースとして Ec2InstanceMetadata を指定します。

```
credential_source = Ec2InstanceMetadata
```

4. リージョンを設定します。

```
region = region
```

例

次の例では、を前提としています。##### ロールとは、という名前の Amazon EC2 インスタンスプロファイルで *us-west-2* リージョンを使用します *marketingadmin*。

```
[profile marketingadmin]  
role_arn = arn:aws:iam::123456789012:role/marketingadminrole  
credential_source = Ec2InstanceMetadata  
region = us-west-2
```


の外部プロセスを使用して認証情報を調達する AWS CLI

⚠ Warning

このトピックでは、外部プロセスからの認証情報の調達について説明します。認証情報を生成するコマンドが非承認のプロセスやユーザーからアクセスできることは、セキュリティ上のリスクにつながります。認証情報を侵害するリスクを軽減 AWS するために、AWS CLI および が提供する、サポートされている安全な代替手段を使用することをお勧めします。漏洩を防ぐために、config ファイルおよびサポートされたファイルとツールの安全を確保します。

SDKs とはそのような情報をキャプチャしてログに記録し AWS CLI、許可されていないユーザーに公開する可能性がある StdErr ため、カスタム認証情報ツールが機密情報を書き込まないようにしてください。

によって直接サポートされていない認証情報を生成または検索する方法がある場合は AWS CLI、config ファイルで credential_process 設定を設定することで、それを使用する AWS CLI ように を設定できます。

例えば、config ファイルに次のようなエントリを含めることができます。

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

構文

任意のオペレーティングシステムと互換性のある方法でこの文字列を作成するには、以下のルールに従います。

- パスまたはファイル名にスペースが含まれている場合は、完全なパスとファイル名を二重引用符 (" ") で囲みます。パスとファイル名は A ~ Z、a ~ z、0 ~ 9、- _ . の文字、スペースのみで構成できます。
- パラメータ名またはパラメータ値にスペースが含まれている場合は、その要素を二重引用符 (" ") で囲みます。囲むのは、名前または値のみであり、そのペアではありません。
- 文字列に環境変数を含めないでください。例えば、\$HOME または %USERPROFILE% を含めることはできません。
- ホームフォルダを ~ として指定しないでください。完全パスを指定する必要があります。

Windows の例

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter with spaces"
```

Linux または macOS の例

```
credential_process = "/Users/Dave/path/to/credentials.sh" parameterWithoutSpaces "parameter with spaces"
```

認証情報プログラムからの想定される出力

は、プロファイルで指定されたコマンド AWS CLI を実行し、 からデータを読み込みますSTDOUT。指定するコマンドはSTDOUT、次の構文に一致するJSON出力を生成する必要があります。

```
{
  "Version": 1,
  "AccessKeyId": "an AWS access key",
  "SecretAccessKey": "your AWS secret access key",
  "SessionToken": "the AWS session token for temporary credentials",
  "Expiration": "ISO8601 timestamp when the credentials expire"
}
```

Note

本文書の執筆時点では、Version キーは 1 に設定する必要があります。構造が進化するため、時間の経過と共に増えていく可能性があります。

Expiration キーは [ISO8601](#) 形式のタイムスタンプです。Expiration キーがツールの出力に存在しない場合、は、認証情報が更新しない長期的な認証情報であることをCLI前提としています。それ以外の認証情報は一時的な認証情報と見なされ、有効期限が切れる前に credential_process を再実行して自動的に更新されます。

Note

AWS CLI は、外部プロセス認証情報を、引き受けロール認証情報のようにキャッシュしません。キャッシュが必要な場合は、外部プロセス内で実装する必要があります。

外部プロセスはゼロ以外のリターンコードを返して、認証情報の取得時にエラーが発生したことを示すことができます。

の使用 AWS CLI

このセクションでは、設定セクションで説明されている詳細を超えて、AWS Command Line Interface (AWS CLI) で利用可能な一般的な使用、一般的な機能、およびオプションの概要を説明します [the section called “エンドポイント”](#)。

このガイドでは、基本的な構造、フォーマット、フィルタリング機能など、AWS CLI コマンドの記述に関する基本的な側面について詳しく説明します。これらのコア要素を理解することで、複雑なウェブベースのコンソールを操作することなく、必要なリソースとアクションを正確にターゲットとするコマンドを構築できます。

さらに、で利用できるヘルプコンテンツとドキュメントが強調表示されます AWS CLI。組み込みのコマンドラインヘルプから包括的な [AWS CLI バージョン 2 リファレンスガイド](#) まで、の機能を調べるのに役立つ情報にアクセスできます AWS CLI。

AWS のサービス 具体的な例とユースケースについては、 [コードの例](#) または [AWS CLI バージョン 2 リファレンスガイド](#) を参照してください。これらはコマンド固有の情報を提供し、AWS CLI さまざまな に活用する方法の例を示します AWS のサービス。

Note

デフォルトでは、はTCPポート 443 HTTPSで AWS のサービス を使用して にリクエスト AWS CLI を送信します。を正常に使用するには AWS CLI、このポートでアウトバウンド接続を実行できる必要があります。

このガイドのトピック

- [のヘルプとリソースへのアクセス AWS CLI](#)
- [のコマンド構造 AWS CLI](#)
- [でのパラメータ値の指定 AWS CLI](#)
- [でのコマンドプロンプトの有効化と使用 AWS CLI](#)
- [でのコマンド出力の制御 AWS CLI](#)
- [のコマンドラインリターンコード AWS CLI](#)
- [カスタムウィザードを使用して でインタラクティブコマンドを実行する AWS CLI](#)
- [でのエイリアスの作成と使用 AWS CLI](#)

のヘルプとリソースへのアクセス AWS CLI

このトピックでは、AWS Command Line Interface () のヘルプコンテンツにアクセスする方法について説明しますAWS CLI。

トピック

- [組み込みの AWS CLI help コマンド](#)
- [AWS CLI リファレンスガイド](#)
- [API ドキュメント](#)
- [エラーのトラブルシューティング](#)
- [その他のヘルプ](#)

組み込みの AWS CLI help コマンド

AWS Command Line Interface () を使用する場合、どのコマンドでもヘルプを受けることができます AWS CLI。そのためには、コマンド名の末尾に help と入力するだけです。

例えば、次のコマンドは、一般的な AWS CLI オプションと使用可能な最上位コマンドのヘルプを表示します。

```
$ aws help
```

次のコマンドは、使用可能な Amazon Elastic Compute Cloud (Amazon EC2) 固有のコマンドを表示します。

```
$ aws ec2 help
```

次の例では、Amazon EC2DescribeInstancesオペレーションの詳細なヘルプを表示します。ヘルプには、入力パラメータ、使用可能なフィルター、および出力に含まれるものについての説明があります。コマンドの一般的なバリエーションを入力する方法を示す例も含まれています。

```
$ aws ec2 describe-instances help
```

各コマンドのヘルプは 6 つのセクションに分かれています。

名前

コマンドの名前。

NAME

```
describe-instances -
```

説明

コマンドが呼び出すAPIオペレーションの説明。

DESCRIPTION

Describes one or more of your instances.

If you specify one or more instance IDs, Amazon EC2 returns information for those instances. If you do not specify instance IDs, Amazon EC2 returns information for all relevant instances. If you specify an instance ID that is not valid, an error is returned. If you specify an instance that you do not own, it is not included in the returned results.

...

概要

コマンドとそのオプションを使用するための基本的な構文。オプションが角括弧で示されている場合は、そのオプションが任意である、デフォルト値がある、または使用できる代替オプションがあることを意味しています。

SYNOPSIS

```
describe-instances
[--dry-run | --no-dry-run]
[--instance-ids <value>]
[--filters <value>]
[--cli-input-json <value>]
[--starting-token <value>]
[--page-size <value>]
[--max-items <value>]
[--generate-cli-skeleton]
```

例えば、`describe-instances`には、現在のアカウントとAWSリージョン内のすべてのインスタンスを記述するデフォルトの動作があります。必要に応じて `instance-ids` のリストを指定して、1つ以上のインスタンスを定義することもできます。`dry-run` は値を取らないオプションのブールフラグです。ブールフラグを使用するには、表示される値のいずれかを指定します。この場合は `--dry-run` または `--no-dry-run` です。同様に、`--generate-cli-skeleton`

も値を取りません。オプションの使用に条件がある場合は、OPTIONS セクションで説明されるか、例に示されます。

オプション

Synopsis に示される各オプションの説明。

OPTIONS

`--dry-run | --no-dry-run (boolean)`

Checks whether you have the required permissions for the action, without actually making the request, and provides an error response. If you have the required permissions, the error response is `DryRunOperation`. Otherwise, it is `UnauthorizedOperation`.

`--instance-ids (list)`

One or more instance IDs.

Default: Describes all your instances.

...

例

コマンドとそのオプションの使用法を示す例。必要なコマンドまたはユースケースの例がない場合は、このページのフィードバックリンク、または AWS CLI コマンドのヘルプページのコマンドリファレンスを使用してリクエストします。

EXAMPLES

To describe an Amazon EC2 instance

Command:

```
aws ec2 describe-instances --instance-ids i-5203422c
```

To describe all instances with the instance type m1.small

Command:

```
aws ec2 describe-instances --filters "Name=instance-type,Values=m1.small"
```

To describe all instances with an Owner tag

Command:

```
aws ec2 describe-instances --filters "Name=tag-key,Values=Owner"
```

```
...
```

出力

からの応答に含まれる各フィールドとデータタイプの説明 AWS

describe-instances の場合は、出力は予約オブジェクトのリストであり、それぞれのオブジェクトに、関連付けられたインスタンスに関する情報を含む複数のフィールドとオブジェクトがあります。この情報は、Amazon で使用される[API予約データタイプのドキュメント](#)から取得されますEC2。

OUTPUT

Reservations -> (list)

One or more reservations.

(structure)

Describes a reservation.

ReservationId -> (string)

The ID of the reservation.

OwnerId -> (string)

The ID of the AWS account that owns the reservation.

RequesterId -> (string)

The ID of the requester that launched the instances on your behalf (for example, AWS Management Console or Auto Scaling).

Groups -> (list)

One or more security groups.

(structure)

Describes a security group.

GroupName -> (string)

The name of the security group.

GroupId -> (string)

The ID of the security group.

Instances -> (list)

One or more instances.


```
(structure)
  Describes an instance.

  InstanceId -> (string)
    The ID of the instance.

  ImageId -> (string)
    The ID of the AMI used to launch the instance.

  State -> (structure)
    The current state of the instance.

  Code -> (integer)
    The low byte represents the state. The high byte
    is an opaque internal value and should be ignored.

...
```

が出力をに AWS CLI レンダリングするとJSON、次の例のように予約オブジェクトの配列になります。

```
{
  "Reservations": [
    {
      "OwnerId": "012345678901",
      "ReservationId": "r-4c58f8a0",
      "Groups": [],
      "RequesterId": "012345678901",
      "Instances": [
        {
          "Monitoring": {
            "State": "disabled"
          },
          "PublicDnsName": "ec2-52-74-16-12.us-
west-2.compute.amazonaws.com",
          "State": {
            "Code": 16,
            "Name": "running"
          },
        },
      ],
    },
  ],
}
```

各予約オブジェクトには、予約およびインスタンスオブジェクトの配列を説明するフィールドがあり、それぞれにそれを説明する独自のフィールド (例: `PublicDnsName`) とオブジェクト (例: `State`) があります。

Windows ユーザー

ヘルプコマンドの出力を `more` コマンドにパイプ (`|`) して、ヘルプファイルを 1 ページずつ表示することができます。スペースバー または `PgDn` を押してドキュメントをさらに表示し、`q` 終了します。

```
C:\> aws ec2 describe-instances help | more
```

AWS CLI リファレンスガイド

ヘルプファイルには、コマンドラインからは表示や移動ができないリンクが含まれています。これらのリンクを表示して操作するには、オンライン [AWS CLI バージョン 2 リファレンスガイド](#) を使用します。リファレンスには、すべての AWS CLI コマンドのヘルプコンテンツも含まれています。説明は、モバイル、タブレット、またはデスクトップ画面で移動や表示がしやすいように表示されます。

API ドキュメント

のすべてのコマンドは、AWS サービスのパブリックに対するリクエスト AWS CLI に対応します API。パブリックを持つ各サービス API には、[AWS ドキュメントウェブサイト](#) のサービスのホームページにある API リファレンスがあります。API リファレンスのコンテンツ API は、の構築方法と使用するプロトコルによって異なります。通常、API リファレンスには、でサポートされているオペレーション API、サービスとの間で送受信されるデータ、およびサービスがレポートできるエラー条件に関する詳細情報が含まれています。

API ドキュメントセクション

- **アクション** - 各オペレーションとそのパラメータに関する詳細情報 (長さまたは内容に関する制約、デフォルト値を含む)。このオペレーションで発生する可能性のあるエラーが一覧表示されます。各オペレーションは、のサブコマンドに対応します AWS CLI。
- **データタイプ** - コマンドが必要なパラメータそして、リクエストに 응답して返す構造体に関する詳細情報。
- **Common Parameters** - サービスのすべてのアクションに共通のパラメータに関する詳細情報。
- **Common Errors** - サービスの操作によって返される可能性のあるエラーに関する詳細情報。

各セクションの名前と有無は、サービスによって異なる場合があります。

サービス固有 CLIs

一部のサービスには、すべてのサービスを操作するために 1 つの AWS CLI が作成される前とは別の CLI があります。これらのサービス固有のドキュメントには CLIs、サービスのドキュメントページにリンクされた個別のドキュメントがあります。サービス固有のドキュメント CLIs は、には適用されません AWS CLI。

エラーのトラブルシューティング

AWS CLI エラーの診断と修正については、「」を参照してください [エラーのトラブルシューティング](#)。

その他のヘルプ

AWS CLI 問題に関するその他のヘルプについては、の [AWS CLI コミュニティ](#) を参照してください GitHub。

のコマンド構造 AWS CLI

このトピックでは、AWS Command Line Interface (AWS CLI) コマンドの構造と、待機コマンドの使用方法について説明します。

トピック

- [コマンド構造](#)
- [Wait コマンド](#)

コマンド構造

は、コマンドラインでマルチパート構造 AWS CLI を使用します。この構造は、次の順序で指定する必要があります。

1. aws プログラムのベースコール。
2. 最上位のコマンド。通常、がサポートする AWS サービスに対応します AWS CLI。
3. 実行する操作を指定するサブコマンド。

4. オペレーションに必要な一般的な AWS CLI オプションまたはパラメータ。これらは、最初の 3 つの部分に続く限り、任意の順序で指定することができます。排他的パラメータが複数回指定された場合は、最後の値のみ適用されます。

```
$ aws <command> <subcommand> [options and parameters]
```

パラメータは、数値、文字列、リスト、マップ、JSON構造など、さまざまなタイプの入力値を取ることができます。サポートされる内容は、指定したコマンドおよびサブコマンドによって異なります。

例

Amazon S3

次の例では、すべての Amazon S3 バケットを一覧表示します。

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

Amazon S3 コマンドの詳細については、AWS CLI コマンドリファレンスの「[aws s3](#)」を参照してください。

AWS CloudFormation

次の[create-change-set](#)コマンド例では、クラウドフォーメーションスタック名を `my-change-set` に変更します。

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-change-set
```

AWS CloudFormation コマンドの詳細については、AWS CLI 「コマンドリファレンス[aws cloudformation](#)」の「」を参照してください。

Wait コマンド

一部の AWS サービスではwait、コマンドを使用できます。aws wait を使用するコマンドは、通常、コマンドが完了するまで待機してから、次のステップに進みます。wait コマンドを使用すると、wait コマンドが失敗した場合に後続のステップに移動するのを防ぐことができるため、マルチパートコマンドやスクリプトに特に便利です。

は、コマンドラインにマルチパート構造 AWS CLI を使用します。waitこの構造は、次の順序で指定する必要があります。

1. aws プログラムのベースコール。
2. 最上位のコマンド。通常、 がサポートする AWS サービスに対応します AWS CLI。
3. wait コマンド。
4. 実行する操作を指定するサブコマンド。
5. オペレーションに必要な一般的なCLIオプションまたはパラメータ。これらは、最初の 3 つのパートに続く限り、任意の順序で指定することができます。排他的パラメータが複数回指定された場合は、最後の値のみ適用されます。

```
$ aws <command> wait <subcommand> [options and parameters]
```

パラメータは、数値、文字列、リスト、マップ、JSON構造など、さまざまなタイプの入力値を取ることができます。サポートされる内容は、指定したコマンドおよびサブコマンドによって異なります。

Note

すべてのAWSサービスがwaitコマンドをサポートしているわけではありません。[AWS CLI バージョン 2 リファレンスガイド](#)を参照して、サービスがwaitコマンドをサポートしているかどうかを確認してください。

例

AWS CloudFormation

次の[wait change-set-create-complete](#)コマンド例は、 が *my-change-set* の変更セット *my-stack* スタックを実行する準備ができました。

```
$ aws cloudformation wait change-set-create-complete --stack-name my-stack --change-set-name my-change-set
```

AWS CloudFormation wait コマンドの詳細については、AWS CLI コマンドリファレンスの「[wait](#)」を参照してください。

AWS CodeDeploy

次の[wait deployment-successful](#)コマンド例は、が `d-A1B2C3111` デプロイが正常に完了しました。

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

AWS CodeDeploy wait コマンドの詳細については、AWS CLI コマンドリファレンスの「[wait](#)」を参照してください。

でのパラメータ値の指定 AWS CLI

AWS Command Line Interface (AWS CLI) で使用される多くのパラメータは、次のaws ec2 create-key-pairコマンド例my-key-pairのキーペア名など、単純な文字列または数値です。

```
$ aws ec2 create-key-pair --key-name my-key-pair
```

コマンドのフォーマットはターミナルによって異なる場合があります。たとえば、ほとんどのターミナルは大文字と小文字を区別しますが、Powershell は大文字と小文字を区別しません。つまり、以下の2つのコマンド例では、MyFile*.txt と myfile*.txt を異なるパラメータとして表示するため、大文字と小文字が区別されるターミナルでは異なる結果になります。

ただし、これらのリクエストは、表示されるものと同じとしてMyFile*.txt、同じパラメータmyfile*.txtとして処理 PowerShell されます。次のコマンド例は、aws s3 cp コマンドを使用したこれらのパラメータを示しています。

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "MyFile*.txt"
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "myfile*.txt"
```

PowerShellの大文字と小文字が区別されない方法の詳細については、PowerShellドキュメントの「[about_Case-Sensitivity](#)」を参照してください。

特殊文字やスペース文字を含む文字列を引用符やリテラルで囲む必要がある場合があります。このフォーマットに関する規則は、ターミナルによっても異なる場合があります。複雑なパラメータを引用符で囲む方法の詳細については、「[の文字列での引用符とリテラルの使用 AWS CLI](#)」を参照してください。

これらのトピックでは、最も一般的なターミナルフォーマットルールについて説明します。ターミナルでパラメータ値を認識できない場合は、このセクションのトピックを確認し、ターミナルのドキュメントで特定の構文ルールを確認してください。

パラメータトピック

- [の一般的なパラメータタイプ AWS CLI](#)
- [の文字列での引用符とリテラルの使用 AWS CLI](#)
- [のファイルからパラメータをロードする AWS CLI](#)
- [AWS CLI のスケルトンと入力ファイル AWS CLI](#)
- [での省略構文の使用 AWS CLI](#)

の一般的なパラメータタイプ AWS CLI

このセクションでは、いくつかの一般的なパラメータタイプと一般的に必要な形式について説明します。

特定のコマンドでパラメータの書式化に問題がある場合には、コマンド名の後に **help** と入力することによって、ヘルプを確認してみてください。各サブコマンドのヘルプには、オプションの名前と説明が示されています。オプションのパラメータタイプは、括弧内に一覧表示されています。ヘルプ表示の詳細については、「[the section called “ヘルプが必要な場合”](#)」を参照してください。

パラメータタイプには次のものがあります。

- [文字列](#)
- [タイムスタンプ](#)
- [リスト](#)
- [ブール値](#)
- [整数](#)
- [バイナリ/blob \(バイナリラージオブジェクト\) とストリーミング blob](#)
- [マッピング](#)
- [ドキュメント](#)

文字列

文字列パラメータには、文字 [ASCII](#) セットの英数字、記号、空白を含めることができます。空白文字を含む文字列は引用符で囲まれている必要があります。予期しない結果を避けるため、標準の空白文字以外の記号や空白文字は使用せず、お使いの端末の [引用符のルール](#) に従うことをお勧めします。

一部の文字列パラメータはファイルからバイナリデータを受け取ることができます。例については、「[バイナリファイル](#)」を参照してください。

タイムスタンプ

タイムスタンプは [ISO8601](#) 標準に従ってフォーマットされます。これらは「DateTime」または「Date」パラメータと呼ばれることもあります。

```
$ aws ec2 describe-spot-price-history --start-time 2014-10-13T19:00:00Z
```

有効な形式は次のとおりです。

- `YYYY-MM-DDThh:mm:ss.sssTZD (UTC)`、例えば、`2014-10-01T20:30:00.000Z`
- `YYYY-MM-DDThh:mm:ss.sssTZD (with offset)`、例えば、`2014-10-01T12:30:00.000-08:00`
- `YYYY-MM-DD` 例えば、`2014-10-01`
- Unix 時間 (秒)、例: `1412195400`。これは [Unix Epoch 時間](#) と呼ばれることがあり、1970 年 1 月 1 日深夜以降の秒数を表します UTC。

デフォルトでは、AWS CLI バージョン 2 はすべてのレスポンス DateTime 値を 8601 ISO 形式に変換します。

[cli_timestamp_format](#) ファイル設定を使用して、タイムスタンプ形式を設定できます。

リスト

スペースで区切られた 1 つ以上の文字列。文字列項目にスペースがある場合は、その項目を引用符で囲む必要があります。予期しない結果を避けるため。お使いの端末の [引用のルール](#) を遵守してください。

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge m1.medium
```

ブール値

オプションをオンまたはオフにするバイナリフラグです。例えば、`ec2 describe-spot-price-history` にはブール型の `--dry-run` パラメータがあり、このパラメータが指定されると、クエリを実際に行うことなくサービスのクエリを検証します。

```
$ aws ec2 describe-spot-price-history --dry-run
```


出力にはコマンドが正しい形式だったかが示されます。このコマンドには、`--no-dry-run` バージョンのパラメータも含まれ、これを使用して、コマンドを通常どおりに実行することを明示的に示すことができます。これは、デフォルトの動作であるため、含める必要はありません。

整数

符号なしの整数。

```
$ aws ec2 describe-spot-price-history --max-items 5
```

バイナリ/blob (バイナリラージオブジェクト) とストリーミング blob

では AWS CLI、バイナリ値をコマンドラインで文字列として直接渡すことができます。blob には 2 つのタイプがあります。

- [blob](#)
- [ストリーミング blob](#)

blob

タイプ blob で、パラメータに値を渡すには、`fileb://` プレフィックスを使用してバイナリデータを含むローカルファイルへのパスを指定する必要があります。`fileb://` プレフィックスを使用して参照されるファイルは、常に、エンコードされていない raw バイナリとして扱われます。指定されたパスは、現在の作業ディレクトリに対する相対パスとして解釈されます。例えば、`--plaintext` の `aws kms encrypt` パラメータは blob です。

```
$ aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  --decode > ExampleEncryptedFile
```

Note

下位互換性を保つため、`file://` プレフィックスを使用できます。ファイル設定 [cli_binary_format](#) または `--cli-binary-format` コマンドラインオプションに応じて、次の 2 つの形式が使用されます。

- AWS CLI バージョン 2 のデフォルト。設定の値が `base64` の場合、`file://` プレフィックスを使用して参照されるファイルは、`base64` でエンコードされたテキストとして扱われます。
- AWS CLI バージョン 1 のデフォルト。設定の値が `raw-in-base64-out` の場合、`file://` プレフィックスを使用して参照されるファイルはテキストとして読み取られます。AWS CLI は、これをバイナリにエンコードしようとします。

詳細については、ファイル設定 [cli_binary_format](#) または `--cli-binary-format` コマンドラインオプションを参照してください。

ストリーミング blob

`aws cloudsearchdomain upload-documents` などのストリーミング blob はプレフィックスを使用しません。代わりに、ストリーミング blob パラメータは直接ファイルパスを使用してフォーマットされます。次の例では、`aws cloudsearchdomain upload-documents` コマンドに直接ファイルパス `document-batch.json` を使用しています。

```
$ aws cloudsearchdomain upload-documents \  
  --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com \  
  --content-type application/json \  
  --documents document-batch.json
```

マッピング

JSON または の CLI [省略構文](#) を使用して指定されたキーと値のペアのセット。次の JSON 例では、`my-table` という名前の Amazon DynamoDB テーブルから、マップパラメータを使用して項目を読み取ります `--key`。パラメータは、`ID` という名前のプライマリキーを指定し、ネストされた JSON 構造体の数値は 1 です。

コマンドラインでより高度な JSON 使用法を得るには、`jq` のようなコマンドライン JSON プロセッサを使用して JSON 文字列を作成する `jq` ことを検討してください。の詳細については `jq`、[「jq リポジトリ」](#) を参照してください GitHub。

```
$ aws dynamodb get-item --table-name my-table --key '{"id": {"N": "1"}}'  
  
{  
  "Item": {
```

```
    "name": {
      "S": "John"
    },
    "id": {
      "N": "1"
    }
  }
}
```

ドキュメント

Note

[短縮構文](#)は、ドキュメントタイプと互換性がありません。

ドキュメントタイプは、文字列JSON内に埋め込むことなくデータを送信するために使用されます。ドキュメントタイプによってサービスが任意のスキーマを提供することで、より柔軟なデータ型を使用できます。

これにより、値をエスケープすることなくJSONデータを送信できます。例えば、次のエスケープされたJSON入力を使用する代わりに、次のようにします。

```
{"document": "{\"key\":true}"}
```

次のドキュメントタイプが使用できます。

```
{"document": {"key": true}}
```

ドキュメントタイプに対し有効な値

ドキュメントタイプは柔軟な性質を持っているため、有効な値のタイプは複数あります。有効な値には次のようなものがあります。

文字列

```
--option "value"
```

数

```
--option 123
```

```
--option 123.456
```

ブール値

```
--option true
```

Null

```
--option null
```

配列

```
--option ["value1", "value2", "value3"]  
--option ["value", 1, true, null, ["key1", 2.34], {"key2": "value2"}]
```

オブジェクト

```
--option {"key": "value"}  
--option {"key1": "value1", "key2": 123, "key3": true, "key4": null, "key5":  
["value3", "value4"], "key6": {"value5": "value6"}}
```

の文字列での引用符とリテラルの使用 AWS CLI

AWS CLIでは、主に、一重引用符と二重引用符の使用方法が 2 つあります。

- [空白を含む文字列を引用符で囲む](#)
- [文字列内での引用符の使用](#)

空白を含む文字列を引用符で囲む

コマンドラインでパラメータ名とその値はスペースで区切ります。文字列値に埋め込みスペースが含まれている場合は、文字列全体を引用符で囲んで、 がスペースを値と次のパラメータ名の区切り文字として AWS CLI 誤って解釈しないようにする必要があります。使用する引用符のタイプは、 を実行しているオペレーティングシステムによって異なります AWS CLI。

Linux and macOS

一重引用符 ' ' を使用します。

```
$ aws ec2 create-key-pair --key-name 'my key pair'
```

引用符の使用方法の詳細については、使用するシェルのユーザードキュメントを参照してください。

PowerShell

一重引用符 (推奨)

一重引用符 ' ' は verbatim 文字列と呼ばれます。文字列は、入力したとおりに コマンドに渡されます。つまり、PowerShell 変数は渡されません。

```
PS C:\> aws ec2 create-key-pair --key-name 'my key pair'
```

二重引用符

二重引用符 " " は expandable 文字列と呼ばれます。変数は拡張可能な文字列で渡すことができます。

```
PS C:\> aws ec2 create-key-pair --key-name "my key pair"
```

引用符の使用の詳細については、Microsoft PowerShell Docs の [「引用ルールについて」](#) を参照してください。

Windows command prompt

二重引用符 " " を使用します。

```
C:\> aws ec2 create-key-pair --key-name "my key pair"
```

オプションとして、パラメータ名と値をスペースの代わりに等号 = で区切ることができます。通常、これはパラメータの値がハイフンで始まる場合にのみ必要です。

```
$ aws ec2 delete-key-pair --key-name=-mykey
```

文字列内での引用符の使用

文字列に引用符が含まれている場合があり、シェルが正しく動作するためには、引用符のエスケープが必要になることがあります。一般的なパラメータ値タイプの 1 つは JSON 文字列です。これ

は、JSON構造内の各要素名と値 " "の周りにスペースと二重引用符が含まれているため、複雑です。コマンドラインにJSON形式のパラメータを入力する方法は、オペレーティングシステムによって異なります。

コマンドラインでより高度なJSON使用法を得るには、などのコマンドラインJSONプロセッサを使用してJSON文字列を作成するjqことを検討してください。の詳細についてはjq、「」の「[jq リポジトリ](#)」を参照してくださいGitHub。

Linux and macOS

Linux と macOS が文字列を文字通り解釈するには、次の例のように ' '、一重引用符を使用してJSONデータ構造を囲みます。文字通り扱われるため、JSON文字列に埋め込まれた二重引用符をエスケープする必要はありません。JSONは一重引用符で囲まれているため、文字列内の一重引用符はエスケープする必要があります。これは通常、一重引用符の前にバックスラッシュを使用して達成されます\ '。

```
$ aws ec2 run-instances \  
  --image-id ami-12345678 \  
  --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":  
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]'
```

引用符の使用方法の詳細については、使用するシェルのユーザードキュメントを参照してください。

PowerShell

一重引用符 ' ' または二重引用符 " " を使用します。

一重引用符 (推奨)

一重引用符 ' ' は verbatim 文字列と呼ばれます。文字列は、入力したとおりに コマンドに渡されます。つまり、PowerShell 変数は渡されません。

JSON データ構造には二重引用符が含まれているため、一重引用符 ' ' で囲むことをお勧めします。一重引用符を使用する場合、JSON文字列に埋め込まれた二重引用符をエスケープする必要はありません。ただし、JSON構造`内のバックティックで各引用符をエスケープする必要があります。

```
PS C:\> aws ec2 run-instances `  
  --image-id ami-12345678 `
```

```
--block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]'
```

二重引用符

二重引用符 " " は expandable 文字列と呼ばれます。変数は拡張可能な文字列で渡すことができます。

二重引用符を使用する場合は、JSON文字列に埋め込まれた一重引用符をエスケープする必要はありません。ただし、次の例のように、JSON構造内のバックティックで各二重引用符をエスケープする必要があります。

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings [{"DeviceName`":`"/dev/sdb`,`",`"Ebs`":
{"VolumeSize`":20,`"DeleteOnTermination`":false,`"VolumeType`":`"standard`"}]`
```

引用符の使用の詳細については、Microsoft PowerShell Docs の [「引用ルールについて」](#) を参照してください。

Warning

が コマンドを PowerShell に送信する前に AWS CLI、コマンドが一般的なルール PowerShell または CommandLineToArgvW 引用ルールを使用して解釈されるかどうかを決定します。を使用して PowerShell を処理する場合 CommandLineToArgvW、バックスラッシュで文字をエスケープする必要があります。

CommandLineToArgvW のの詳細については PowerShell、Microsoft DevBlogs の [CommandLineToArgv 「W による引用符とバックスラッシュの奇妙な処理」](#)、Microsoft Docs ブログの [「W 関数でコマンドライン引数を間違った方法で引用する」](#)、および [「Microsoft Docs の」](#) を参照してください。 [CommandLineToArgv](#)

一重引用符

一重引用符 ' ' は verbatim 文字列と呼ばれます。文字列は、入力したとおりに コマンドに渡されます。つまり、PowerShell 変数は渡されません。バックスラッシュ \ で文字をエスケープします。

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings '[{"DeviceName\":"`"/dev/sdb\`,`",\`"Ebs\`":
{\`"VolumeSize\`":20,\`"DeleteOnTermination\`":false,\`"VolumeType\`":\`"standard\`"}]`'
```

二重引用符

二重引用符 " " は expandable 文字列と呼ばれます。変数は expandable 文字列で渡すことができます。二重引用符文字列の場合、を使用して 2 回エスケープする必要があります。 \ バックティックのみを使用するのではなく、各引用符に対して。バックティックはバックスラッシュをエスケープし、バックスラッシュは CommandLineToArgvW プロセスのエスケープ文字として使用されます。

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings "[{"DeviceName`":`"/dev/sdb`",`"Ebs`":
{"VolumeSize`":20,`"DeleteOnTermination`":false,`"VolumeType`":`
`"standard`"}]"`
```

Blobs (推奨)

JSON データ入力の PowerShell 引用ルールをバイパスするには、Blobs を使用して JSON データを に直接渡します AWS CLI。BLOB の詳細については、「[blob](#)」を参照してください。

Windows command prompt

Windows コマンドプロンプトでは、JSON データ構造を囲む " " ために二重引用符が必要です。また、コマンドプロンプトが に埋め込まれた二重引用符を誤って解釈しないようにするには JSON、次の例のように、JSON データ構造自体 " 内の各二重引用符をエスケープ (先頭にバックスラッシュ \ 文字) する必要があります。

```
C:\> aws ec2 run-instances ^
  --image-id ami-12345678 ^
  --block-device-mappings "[{"DeviceName\`":`"/dev/sdb`",`"Ebs\`":
{\`"VolumeSize\`":20,`"DeleteOnTermination\`":false,`"VolumeType\`":`"standard\`"}]"`
```

最も外側の二重引用符のみエスケープしません。

の ファイルからパラメータをロードする AWS CLI

一部のパラメータでは、ファイル名を引数として想定し、そこから がデータを AWS CLI ロードします。他のパラメータを使用すると、コマンドラインで入力するテキストまたはファイルから読み取るテキストとしてパラメータ値を指定できます。ファイルが必要かオプションかにかかわらず、 が

ファイル AWS CLI を理解できるように、ファイルを正しくエンコードする必要があります。ファイルのエンコーディングは、読み取りシステムのデフォルトロケールと一致する必要があります。これは、Python の `locale.getpreferredencoding()` メソッドを使用して判断できます。

Note

デフォルトでは、Windows はテキストを UTF-16 として PowerShell 出力します。これは、JSONファイルや多くの Linux システムで使用されている UTF-8 エンコーディングと競合します。が結果のファイル AWS CLI を読み取れるように、`Command-Encoding ascii` で PowerShell Out-File を使用することをお勧めします。

トピック

- [ファイルからパラメータをロードする方法](#)
- [バイナリファイル](#)

ファイルからパラメータをロードする方法

パラメータが複雑なJSON文字列である場合など、すべてコマンドラインパラメータ値として入力する代わりに、ファイルからパラメータ値をロードするのが便利な場合があります。値を含むファイルを指定するには、URL次の形式でファイルを指定します。

```
file:///complete/path/to/file
```

- 最初の 2 つのスラッシュ「/」文字は仕様の一部です。必要なパスが「/」で始まる場合、結果は 3 つのスラッシュ文字 `file:///folder/file` になります。
- URL は、実際のパラメータコンテンツを含むファイルへのパスを提供します。
- スペースまたは特殊文字を含むファイルを使用する場合は、お使いの端末の[引用符とエスケープのルール](#)に従ってください。

次の例のファイルパスは、現在の作業ディレクトリに対する相対値として解釈されます。

Linux or macOS

```
// Read from a file in the current directory
$ aws ec2 describe-instances --filters file:///filter.json
```

```
// Read from a file in /tmp
$ aws ec2 describe-instances --filters file:///tmp/filter.json

// Read from a file with a filename with whitespaces
$ aws ec2 describe-instances --filters 'file://filter content.json'
```

Windows command prompt

```
// Read from a file in C:\temp
C:\> aws ec2 describe-instances --filters file://C:\temp\filter.json

// Read from a file with a filename with whitespaces
C:\> aws ec2 describe-instances --filters "file://C:\temp\filter content.json"
```

`file://` プレフィックスオプションは、「~/」、「./」、および「../」など、Unix 形式の拡張子をサポートしています。Windows では、「~/」式は、%USERPROFILE% 環境変数に格納されているユーザーディレクトリに展開されます。例えば、Windows 10 では、一般にユーザーディレクトリは `C:\Users\UserName\` にあります。

別のJSONドキュメントの値として埋め込まれたJSONドキュメントはエスケープする必要があります。

```
$ aws sqs create-queue --queue-name my-queue --attributes file://attributes.json
```

attributes.json

```
{
  "RedrivePolicy": "{ \"deadLetterTargetArn\": \"arn:aws:sqs:us-west-2:0123456789012:deadletter\", \"maxReceiveCount\": \"5\" }"
}
```

バイナリファイル

バイナリデータをパラメータとして取るコマンドでは、`fileb://` プレフィックスを使用して、データがバイナリコンテンツであることを指定します。バイナリデータを受け入れるコマンドは次のとおりです。

- **aws ec2 run-instances:--user-data**パラメータ

- **aws s3api put-object**:--sse-customer-keyパラメータ
- **aws kms decrypt**:--ciphertext-blobパラメータ

次の例では、Linux コマンドラインツールを使用してバイナリ 256 ビットAESキーを生成し、Amazon S3 に提供してアップロードされたファイルサーバー側を暗号化します。

```
$ dd if=/dev/urandom bs=1 count=32 > sse.key
32+0 records in
32+0 records out
32 bytes (32 B) copied, 0.000164441 s, 195 kB/s
$ aws s3api put-object \
  --bucket amzn-s3-demo-bucket \
  --key test.txt \
  --body test.txt \
  --sse-customer-key fileb://sse.key \
  --sse-customer-algorithm AES256
{
  "SSECustomerKeyMD5": "iVg8oWa8sy714+FjtesrJg==",
  "SSECustomerAlgorithm": "AES256",
  "ETag": "\"a6118e84b76cf98bf04bbe14b6045c6c\""
}
```

形式のパラメータを含むファイルを参照する別の例については、JSON「」を参照してください [IAM 管理ポリシーのユーザーへのアタッチ](#)。

AWS CLI のスケルトンと入力ファイル AWS CLI

ほとんどの AWS CLI コマンドは、ファイルからのすべてのパラメータ入力を受け入れます。これらのテンプレートは、generate-cli-skeleton オプションを使用して生成できます。

トピック

- [AWS CLI スケルトンと入力ファイルについて](#)
- [コマンドスケルトンを生成生成する](#)

AWS CLI スケルトンと入力ファイルについて

ほとんどの AWS Command Line Interface (AWS CLI) コマンドは、--cli-input-json および --cli-input-yaml パラメータを使用して、ファイルからのすべてのパラメータ入力を受け入れる機能をサポートしています。

これらの同じコマンドは、編集して入力できるすべての `--generate-cli-skeleton` パラメータを含む JSON または YAML 形式のファイルを生成するパラメータを提供します。その後、関連する `--cli-input-json` または `--cli-input-yaml` パラメータを使用してコマンドを実行し、入力済みのファイルを指定できます。

⚠ Important

AWS CLI コマンドなど、いくつかの [aws s3](#) は個々の AWS API オペレーションに直接マッピングされません。このようなコマンドは、このトピックで説明している `--generate-cli-skeleton` または `--cli-input-json` および `--cli-input-yaml` パラメータもサポートしていません。特定のコマンドがこれらのパラメータをサポートしているかどうか分からない場合は、次のコマンドを実行して、*service* または *command* 関心のある名前。

```
$ aws service command help
```

出力に、指定されたコマンドがサポートするパラメータを示す Synopsis セクションが含まれています。

```
$ aws iam list-users help
...
SYNOPSIS
    list-users
    ...
    [--cli-input-json | --cli-input-yaml]
    ...
    [--generate-cli-skeleton <value>]
...
```

`--generate-cli-skeleton` パラメータを指定すると、コマンドは実行せず、パラメータテンプレートを生成して表示します。これをカスタマイズして、後でコマンドに対する入力として使用することができます。生成されるテンプレートには、そのコマンドによってサポートされているすべてのパラメータが含まれています。

`--generate-cli-skeleton` パラメータには、次のいずれかの値を指定できます。

- `input` – 生成されたテンプレートには、としてフォーマットされたすべての入力パラメータが含まれます。これは、デフォルト値です。

- `yaml-input` – 生成されたテンプレートには、としてフォーマットされたすべての入力パラメータが含まれますYAML。
- `output` – 生成されたテンプレートには、としてフォーマットされたすべての出力パラメータが含まれますJSON。現在、出力パラメータをとしてリクエストすることはできませんYAML。

AWS CLI は基本的にサービスの を囲む「ラッパー」であるためAPI、スケルトンファイルでは、すべてのパラメータを基盤となるAPIパラメータ名で参照することが期待されます。これはパラメータAWS CLI 名とは異なる可能性があります。例えば、という名前の AWS CLI パラメータは、という名前の AWS サービスのAPIパラメータにマッピングuser-nameされる場合があります Username (大文字と小文字が変更されたこととダッシュが欠落していることに注意)。エラーを回避するために、`--generate-cli-skeleton` オプションを使用して「正しい」パラメータ名でテンプレートを生成することをお勧めします。サービスのAPIリファレンスガイドを参照して、予想されるパラメータ名を確認することもできます。テンプレートから、値を指定しない必須でないパラメータを削除できます。

例えば、次のコマンドを実行すると、Amazon Elastic Compute Cloud (Amazon EC2) コマンドのパラメータテンプレートが生成されますrun-instances。

JSON

次の例は、`--generate-cli-skeleton`パラメータのデフォルト値 (input) JSON を使用してフォーマットされたテンプレートを生成する方法を示しています。

```
$ aws ec2 run-instances --generate-cli-skeleton
```

```
{
  "DryRun": true,
  "ImageId": "",
  "MinCount": 0,
  "MaxCount": 0,
  "KeyName": "",
  "SecurityGroups": [
    ""
  ],
  "SecurityGroupIds": [
    ""
  ],
  "UserData": "",
  "InstanceType": "",
  "Placement": {
```

```
    "AvailabilityZone": "",
    "GroupName": "",
    "Tenancy": ""
  },
  "KernelId": "",
  "RamdiskId": "",
  "BlockDeviceMappings": [
    {
      "VirtualName": "",
      "DeviceName": "",
      "Ebs": {
        "SnapshotId": "",
        "VolumeSize": 0,
        "DeleteOnTermination": true,
        "VolumeType": "",
        "Iops": 0,
        "Encrypted": true
      },
      "NoDevice": ""
    }
  ],
  "Monitoring": {
    "Enabled": true
  },
  "SubnetId": "",
  "DisableApiTermination": true,
  "InstanceInitiatedShutdownBehavior": "",
  "PrivateIpAddress": "",
  "ClientToken": "",
  "AdditionalInfo": "",
  "NetworkInterfaces": [
    {
      "NetworkInterfaceId": "",
      "DeviceIndex": 0,
      "SubnetId": "",
      "Description": "",
      "PrivateIpAddress": "",
      "Groups": [
        ""
      ],
      "DeleteOnTermination": true,
      "PrivateAddresses": [
        {
          "PrivateIpAddress": "",
```

```

        "Primary": true
      }
    ],
    "SecondaryPrivateIpAddressCount": 0,
    "AssociatePublicIpAddress": true
  }
],
"IamInstanceProfile": {
  "Arn": "",
  "Name": ""
},
"EbsOptimized": true
}

```

YAML

次の例は、`--generate-cli-skeleton`パラメータ`yaml-input`の値YAMLを使用してフォーマットされたテンプレートを生成する方法を示しています。

```
$ aws ec2 run-instances --generate-cli-skeleton yaml-input
```

```

BlockDeviceMappings: # The block device mapping entries.
- DeviceName: '' # The device name (for example, /dev/sdh or xvdh).
  VirtualName: '' # The virtual device name (ephemeralN).
  Ebs: # Parameters used to automatically set up Amazon EBS volumes when the
instance is launched.
    DeleteOnTermination: true # Indicates whether the EBS volume is deleted on
instance termination.
    Iops: 0 # The number of I/O operations per second (IOPS) that the volume
supports.
    SnapshotId: '' # The ID of the snapshot.
    VolumeSize: 0 # The size of the volume, in GiB.
    VolumeType: st1 # The volume type. Valid values are: standard, io1, gp2, sc1,
st1.
    Encrypted: true # Indicates whether the encryption state of an EBS volume is
changed while being restored from a backing snapshot.
    KmsKeyId: '' # Identifier (key ID, key alias, ID ARN, or alias ARN) for a
customer managed KMS key under which the EBS volume is encrypted.
    NoDevice: '' # Suppresses the specified device included in the block device
mapping of the AMI.
ImageId: '' # The ID of the AMI.
InstanceType: c4.4xlarge # The instance type. Valid values are: t1.micro, t2.nano,
t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge, t3.nano, t3.micro,

```

t3.small, t3.medium, t3.large, t3.xlarge, t3.2xlarge, t3a.nano, t3a.micro, t3a.small, t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge, m1.small, m1.medium, m1.large, m1.xlarge, m3.medium, m3.large, m3.xlarge, m3.2xlarge, m4.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m4.10xlarge, m4.16xlarge, m2.xlarge, m2.2xlarge, m2.4xlarge, cr1.8xlarge, r3.large, r3.xlarge, r3.2xlarge, r3.4xlarge, r3.8xlarge, r4.large, r4.xlarge, r4.2xlarge, r4.4xlarge, r4.8xlarge, r4.16xlarge, r5.large, r5.xlarge, r5.2xlarge, r5.4xlarge, r5.8xlarge, r5.12xlarge, r5.16xlarge, r5.24xlarge, r5.metal, r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge, r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge, r5d.large, r5d.xlarge, r5d.2xlarge, r5d.4xlarge, r5d.8xlarge, r5d.12xlarge, r5d.16xlarge, r5d.24xlarge, r5d.metal, r5ad.large, r5ad.xlarge, r5ad.2xlarge, r5ad.4xlarge, r5ad.8xlarge, r5ad.12xlarge, r5ad.16xlarge, r5ad.24xlarge, x1.16xlarge, x1.32xlarge, x1e.xlarge, x1e.2xlarge, x1e.4xlarge, x1e.8xlarge, x1e.16xlarge, x1e.32xlarge, i2.xlarge, i2.2xlarge, i2.4xlarge, i2.8xlarge, i3.large, i3.xlarge, i3.2xlarge, i3.4xlarge, i3.8xlarge, i3.16xlarge, i3.metal, i3en.large, i3en.xlarge, i3en.2xlarge, i3en.3xlarge, i3en.6xlarge, i3en.12xlarge, i3en.24xlarge, i3en.metal, hi1.4xlarge, hs1.8xlarge, c1.medium, c1.xlarge, c3.large, c3.xlarge, c3.2xlarge, c3.4xlarge, c3.8xlarge, c4.large, c4.xlarge, c4.2xlarge, c4.4xlarge, c4.8xlarge, c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge, c5.9xlarge, c5.12xlarge, c5.18xlarge, c5.24xlarge, c5.metal, c5d.large, c5d.xlarge, c5d.2xlarge, c5d.4xlarge, c5d.9xlarge, c5d.18xlarge, c5n.large, c5n.xlarge, c5n.2xlarge, c5n.4xlarge, c5n.9xlarge, c5n.18xlarge, cc1.4xlarge, cc2.8xlarge, g2.2xlarge, g2.8xlarge, g3.4xlarge, g3.8xlarge, g3.16xlarge, g3s.xlarge, g4dn.xlarge, g4dn.2xlarge, g4dn.4xlarge, g4dn.8xlarge, g4dn.12xlarge, g4dn.16xlarge, cg1.4xlarge, p2.xlarge, p2.8xlarge, p2.16xlarge, p3.2xlarge, p3.8xlarge, p3.16xlarge, p3dn.24xlarge, d2.xlarge, d2.2xlarge, d2.4xlarge, d2.8xlarge, f1.2xlarge, f1.4xlarge, f1.16xlarge, m5.large, m5.xlarge, m5.2xlarge, m5.4xlarge, m5.8xlarge, m5.12xlarge, m5.16xlarge, m5.24xlarge, m5.metal, m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge, m5d.large, m5d.xlarge, m5d.2xlarge, m5d.4xlarge, m5d.8xlarge, m5d.12xlarge, m5d.16xlarge, m5d.24xlarge, m5d.metal, m5ad.large, m5ad.xlarge, m5ad.2xlarge, m5ad.4xlarge, m5ad.8xlarge, m5ad.12xlarge, m5ad.16xlarge, m5ad.24xlarge, h1.2xlarge, h1.4xlarge, h1.8xlarge, h1.16xlarge, z1d.large, z1d.xlarge, z1d.2xlarge, z1d.3xlarge, z1d.6xlarge, z1d.12xlarge, z1d.metal, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, u-24tb1.metal, a1.medium, a1.large, a1.xlarge, a1.2xlarge, a1.4xlarge, a1.metal, m5dn.large, m5dn.xlarge, m5dn.2xlarge, m5dn.4xlarge, m5dn.8xlarge, m5dn.12xlarge, m5dn.16xlarge, m5dn.24xlarge, m5n.large, m5n.xlarge, m5n.2xlarge, m5n.4xlarge, m5n.8xlarge, m5n.12xlarge, m5n.16xlarge, m5n.24xlarge, r5dn.large, r5dn.xlarge, r5dn.2xlarge, r5dn.4xlarge, r5dn.8xlarge, r5dn.12xlarge, r5dn.16xlarge, r5dn.24xlarge, r5n.large, r5n.xlarge, r5n.2xlarge, r5n.4xlarge, r5n.8xlarge, r5n.12xlarge, r5n.16xlarge, r5n.24xlarge.

Ipv6AddressCount: 0 # [EC2-VPC] The number of IPv6 addresses to associate with the primary network interface.


```
Ipv6Addresses: # [EC2-VPC] The IPv6 addresses from the range of the subnet to
  associate with the primary network interface.
- Ipv6Address: '' # The IPv6 address.
KernelId: '' # The ID of the kernel.
KeyName: '' # The name of the key pair.
MaxCount: 0 # [REQUIRED] The maximum number of instances to launch.
MinCount: 0 # [REQUIRED] The minimum number of instances to launch.
Monitoring: # Specifies whether detailed monitoring is enabled for the instance.
  Enabled: true # [REQUIRED] Indicates whether detailed monitoring is enabled.
Placement: # The placement for the instance.
  AvailabilityZone: '' # The Availability Zone of the instance.
  Affinity: '' # The affinity setting for the instance on the Dedicated Host.
  GroupName: '' # The name of the placement group the instance is in.
  PartitionNumber: 0 # The number of the partition the instance is in.
  HostId: '' # The ID of the Dedicated Host on which the instance resides.
  Tenancy: dedicated # The tenancy of the instance (if the instance is running in a
  VPC). Valid values are: default, dedicated, host.
  SpreadDomain: '' # Reserved for future use.
RamdiskId: '' # The ID of the RAM disk to select.
SecurityGroupIds: # The IDs of the security groups.
- ''
SecurityGroups: # [default VPC] The names of the security groups.
- ''
SubnetId: '' # [EC2-VPC] The ID of the subnet to launch the instance into.
UserData: '' # The user data to make available to the instance.
AdditionalInfo: '' # Reserved.
ClientToken: '' # Unique, case-sensitive identifier you provide to ensure the
  idempotency of the request.
DisableApiTermination: true # If you set this parameter to true, you can't terminate
  the instance using the Amazon EC2 console, CLI, or API; otherwise, you can.
DryRun: true # Checks whether you have the required permissions for the action,
  without actually making the request, and provides an error response.
EbsOptimized: true # Indicates whether the instance is optimized for Amazon EBS I/O.
IamInstanceProfile: # The IAM instance profile.
  Arn: '' # The Amazon Resource Name (ARN) of the instance profile.
  Name: '' # The name of the instance profile.
InstanceInitiatedShutdownBehavior: stop # Indicates whether an instance stops or
  terminates when you initiate shutdown from the instance (using the operating system
  command for system shutdown). Valid values are: stop, terminate.
NetworkInterfaces: # The network interfaces to associate with the instance.
- AssociatePublicIpAddress: true # Indicates whether to assign a public IPv4
  address to an instance you launch in a VPC.
  DeleteOnTermination: true # If set to true, the interface is deleted when the
  instance is terminated.
```

```
Description: '' # The description of the network interface.
DeviceIndex: 0 # The position of the network interface in the attachment order.
Groups: # The IDs of the security groups for the network interface.
- ''
Ipv6AddressCount: 0 # A number of IPv6 addresses to assign to the network
interface.
Ipv6Addresses: # One or more IPv6 addresses to assign to the network interface.
- Ipv6Address: '' # The IPv6 address.
NetworkInterfaceId: '' # The ID of the network interface.
PrivateIpAddress: '' # The private IPv4 address of the network interface.
PrivateIpAddresses: # One or more private IPv4 addresses to assign to the network
interface.
- Primary: true # Indicates whether the private IPv4 address is the primary
private IPv4 address.
  PrivateIpAddress: '' # The private IPv4 addresses.
SecondaryPrivateIpAddressCount: 0 # The number of secondary private IPv4
addresses.
SubnetId: '' # The ID of the subnet associated with the network interface.
InterfaceType: '' # The type of network interface.
PrivateIpAddress: '' # [EC2-VPC] The primary IPv4 address.
ElasticGpuSpecification: # An elastic GPU to associate with the instance.
- Type: '' # [REQUIRED] The type of Elastic Graphics accelerator.
ElasticInferenceAccelerators: # An elastic inference accelerator to associate with
the instance.
- Type: '' # [REQUIRED] The type of elastic inference accelerator.
TagSpecifications: # The tags to apply to the resources during launch.
- ResourceType: network-interface # The type of resource to tag. Valid values
are: client-vpn-endpoint, customer-gateway, dedicated-host, dhcp-options, elastic-
ip, fleet, fpga-image, host-reservation, image, instance, internet-gateway,
launch-template, natgateway, network-acl, network-interface, reserved-instances,
route-table, security-group, snapshot, spot-instances-request, subnet, traffic-
mirror-filter, traffic-mirror-session, traffic-mirror-target, transit-gateway,
transit-gateway-attachment, transit-gateway-route-table, volume, vpc, vpc-peering-
connection, vpn-connection, vpn-gateway.
Tags: # The tags to apply to the resource.
- Key: '' # The key of the tag.
  Value: '' # The value of the tag.
LaunchTemplate: # The launch template to use to launch the instances.
LaunchTemplateId: '' # The ID of the launch template.
LaunchTemplateName: '' # The name of the launch template.
Version: '' # The version number of the launch template.
InstanceMarketOptions: # The market (purchasing) option for the instances.
MarketType: spot # The market type. Valid values are: spot.
SpotOptions: # The options for Spot Instances.
```

```
MaxPrice: '' # The maximum hourly price you're willing to pay for the Spot
Instances.
SpotInstanceType: one-time # The Spot Instance request type. Valid values are:
one-time, persistent.
BlockDurationMinutes: 0 # The required duration for the Spot Instances (also
known as Spot blocks), in minutes.
ValidUntil: 1970-01-01 00:00:00 # The end date of the request.
InstanceInterruptionBehavior: terminate # The behavior when a Spot Instance is
interrupted. Valid values are: hibernate, stop, terminate.
CreditSpecification: # The credit option for CPU usage of the T2 or T3 instance.
CpuCredits: '' # [REQUIRED] The credit option for CPU usage of a T2 or T3
instance.
CpuOptions: # The CPU options for the instance.
CoreCount: 0 # The number of CPU cores for the instance.
ThreadsPerCore: 0 # The number of threads per CPU core.
CapacityReservationSpecification: # Information about the Capacity Reservation
targeting option.
CapacityReservationPreference: none # Indicates the instance's Capacity
Reservation preferences. Valid values are: open, none.
CapacityReservationTarget: # Information about the target Capacity Reservation.
CapacityReservationId: '' # The ID of the Capacity Reservation.
HibernationOptions: # Indicates whether an instance is enabled for hibernation.
Configured: true # If you set this parameter to true, your instance is enabled
for hibernation.
LicenseSpecifications: # The license configurations.
- LicenseConfigurationArn: '' # The Amazon Resource Name (ARN) of the license
configuration.
```

コマンドスケルトンを生成生成する

パラメータスケルトンファイルを生成して使用するには

1. `--generate-cli-skeleton` パラメータを使用してコマンドを実行して JSONまたは YAML を生成し、出力をファイルに転送して保存します。

JSON

```
$ aws ec2 run-instances --generate-cli-skeleton input > ec2runinst.json
```

YAML

```
$ aws ec2 run-instances --generate-cli-skeleton yaml-input > ec2runinst.yaml
```

2. テキストエディタでパラメータスケルトンファイルを開き、不要なパラメータを削除します。例えば、テンプレートを次のように削除できます。不要な要素を削除した後JSONもYAML、ファイルがまだ有効であることを確認してください。

JSON

```
{
  "DryRun": true,
  "ImageId": "",
  "KeyName": "",
  "SecurityGroups": [
    ""
  ],
  "InstanceType": "",
  "Monitoring": {
    "Enabled": true
  }
}
```

YAML

```
DryRun: true
ImageId: ''
KeyName: ''
SecurityGroups:
- ''
InstanceType:
Monitoring:
  Enabled: true
```

この例では、Amazon EC2ドライラン機能を使用するtrueのように DryRunパラメータを に設定します。この機能を使用すると、実際にリソースを作成または変更することなく、コマンドを安全にテストできます。

3. 残りの値には、シナリオに適した値を入力します。この例では、使用する Amazon マシンイメージ (AMI) のインスタンスタイプ、キー名、セキュリティグループ、識別子を指定します。こ

の例では、デフォルトの AWS リージョンを前提としています。AMI `ami-dfc39aef` は、`us-west-2` リージョンでホストされている 64 ビットの Amazon Linux イメージです。別のリージョンを使用する場合は、[を使用するための正しい AMI ID を見つける](#)必要があります。

JSON

```
{
  "DryRun": true,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

YAML

```
DryRun: true
ImageId: 'ami-dfc39aef'
KeyName: 'mykey'
SecurityGroups:
- 'my-sg'
InstanceType: 't2.micro'
Monitoring:
  Enabled: true
```

4. `file://` プレフィックスを使用して、完了したテンプレートファイルを `--cli-input-json` または `cli-input-yaml` パラメータに渡すことによって、入力済みパラメータでコマンドを実行します。パスを現在の作業ディレクトリに相対している場合、AWS CLI 解釈するため、パスのないファイル名のみを表示する次の例では、現在の作業ディレクトリでファイルを直接検索します。

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json
```

```
A client error (DryRunOperation) occurred when calling the RunInstances
operation: Request would have succeeded, but DryRun flag is set.
```

YAML

```
$ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml
```

```
A client error (DryRunOperation) occurred when calling the RunInstances
operation: Request would have succeeded, but DryRun flag is set.
```

ドライランエラーは、JSONまたはYAMLが正しく形成され、パラメータ値が有効であることを示します。出力でその他の問題が報告された場合は、それを修正し、「Request would have succeeded」というメッセージが表示されるまで前のステップを繰り返します。

5. これで、DryRun パラメータを `false` に設定して、dry run を無効にできます。

JSON

```
{
  "DryRun": false,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

YAML

```
DryRun: false
ImageId: 'ami-dfc39aef'
KeyName: 'mykey'
SecurityGroups:
- 'my-sg'
InstanceType: 't2.micro'
```

```
Monitoring:
  Enabled: true
```

6. コマンドを実行し、`run-instances`実際に Amazon EC2 インスタンスを起動して、正常に起動したときに生成された詳細を表示します。出力の形式は、入力パラメータテンプレートの形式とは別に、`--output` パラメータによって制御されます。

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --output json
```

```
{
  "OwnerId": "123456789012",
  "ReservationId": "r-d94a2b1",
  "Groups": [],
  "Instances": [
    ...
  ]
}
```

YAML

```
$ aws ec2 run-instances --cli-input-yaml file://ec2runinst.yaml --output yaml
```

```
OwnerId: '123456789012'
ReservationId: 'r-d94a2b1',
Groups:
- ''
Instances:
...

```

での省略構文の使用 AWS CLI

AWS Command Line Interface (AWS CLI) は、多くのオプションパラメータを JSON 形式で受け入れることができます。ただし、コマンドラインに大きな JSON リストや構造を入力するのは面倒かもしれません。これを容易にするために、は、フル JSON フォーマットを使用するよりも簡単なオプションパラメータの表現を可能にする簡潔な構文 AWS CLI もサポートしています。

トピック

- [構造体パラメータ](#)

- [での省略構文の使用 AWS Command Line Interface](#)

構造体パラメータ

の省略構文 AWS CLI により、ユーザーはフラットなパラメータ (ネストされていない構造) を簡単に入力できます。形式は、キーと値のペアのカンマ区切りリストです。短縮構文は文字列であるため、必ず、お使いのターミナルに適した[引用](#)とエスケープの規則を使用してください。

Linux or macOS

```
--option key1=value1,key2=value2,key3=value3
```

PowerShell

```
--option "key1=value1,key2=value2,key3=value3"
```

これらはどちらも次の例と同等で、 でフォーマットされていますJSON。

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

それぞれのカンマ区切りのキーと値のペアの間に空白があってははいけません。以下に、Amazon DynamoDB update-table コマンドの例を示します。--provisioned-throughput オプションは、省略表記で指定されています。

```
$ aws dynamodb update-table \  
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10 \  
  --table-name MyDDBTable
```

これは、 でフォーマットされた次の例に相当しますJSON。

```
$ aws dynamodb update-table \  
  --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \  
  --table-name MyDDBTable
```


での省略構文の使用 AWS Command Line Interface

入力パラメータは、JSONまたは省略形の 2 つの方法でリスト形式で指定できます。AWS CLI の短縮構文は、数値、文字列、またはネストされていない構造体が含まれるリストを簡単に渡せるように設計されています。

基本的な形式を次に示します。ここで、リストの値は、1 つのスペースで区切られます。

```
--option value1 value2 value3
```

これは、でフォーマットされた次の例に相当しますJSON。

```
--option '[value1,value2,value3]'
```

前述したように、数字のリスト、文字列のリスト、またはネストされていない構造の省略表現のリストを指定できます。Amazon Elastic Compute Cloud (Amazon EC2) の stop-instances コマンドの例を次に示します。ここでは、--instance-ids オプションの入力パラメータ (文字列のリスト) を簡潔に指定します。

```
$ aws ec2 stop-instances \  
  --instance-ids i-1486157a i-1286157c i-ec3a7e87
```

これは、でフォーマットされた次の例に相当しますJSON。

```
$ aws ec2 stop-instances \  
  --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

次の例は、--tags オプションのネストされていない構造のリストを取得する Amazon EC2 create-tags コマンドを示しています。--resources オプションは、タグを付けるインスタンスの ID を指定します。

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags Key=My1stTag,Value=Value1 Key=My2ndTag,Value=Value2  
Key=My3rdTag,Value=Value3
```

これは、でフォーマットされた次の例に相当しますJSON。JSON パラメータは、読みやすくするために複数の行に記述されます。

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags '[  
    {"Key": "My1stTag", "Value": "Value1"},  
    {"Key": "My2ndTag", "Value": "Value2"},  
    {"Key": "My3rdTag", "Value": "Value3"}  
  ]'
```

でのコマンドプロンプトの有効化と使用 AWS CLI

AWS CLI バージョン 2 では、コマンドの実行時にawsコマンド、パラメータ、リソースをプロンプトできます。

トピック

- [仕組み](#)
- [自動プロンプト機能](#)
- [自動プロンプトモード](#)
- [自動プロンプトの設定](#)

仕組み

有効にすると、自動プロンプトにより、ENTERキーを使用して部分的に入力されたコマンドを完了できます。ENTER キーを押すと、入力した内容に基づいてコマンド、パラメータ、リソースが提案されます。提案には、左側にコマンド、パラメーター、またはリソースの名前、右側にそれらの説明が記載されます。提案を選択して使用するには、矢印キーを使用して行を強調表示し、SPACEキーを押します。コマンドの入力が完了したら、ENTERを押してコマンドを使用します。以下の例は、自動プロンプトから提案されたリストがどのように表示されるかを示しています。

```
$ aws  
> aws a  
    accessanalyzer           Access Analyzer  
    acm                      AWS Certificate Manager  
    acm-pca                  AWS Certificate Manager Private Certificate  
Authority  
    alexaforbusiness         Alexa For Business  
    amplify                  AWS Amplify
```

自動プロンプト機能

自動プロンプトには、以下の便利な機能が含まれています。

ドキュメントパネル

現在のコマンドのヘルプドキュメントを提供します。ドキュメントを開くには、F3 キーを押します。

コマンド補完

使用する aws コマンドを提案します。リストを表示するには、コマンドを部分的に入力します。以下の例は、a の文字で始まるサービスを検索します。

```
$ aws
> aws a
    accessanalyzer           Access Analyzer
    acm                      AWS Certificate Manager
    acm-pca                  AWS Certificate Manager Private Certificate
Authority
    alexaforbusiness        Alexa For Business
    amplify                 AWS Amplify
```

パラメータ補完

コマンドを入力すると、自動プロンプトがパラメータの提案を開始します。パラメータの説明には、値のタイプ、およびパラメータが何であるかの説明が含まれます。必須のパラメータが最初にリストされ、「required」というラベルが付けられます。以下の例は、aws dynamodb describe-table に対するパラメータの自動プロンプトリストを示しています。

```
$ aws dynamodb describe-table
> aws dynamodb describe-table
    --table-name (required) [string] The name of the
table to describe.
    --cli-input-json        [string] Reads arguments
from the JSON string provided. The JSON string follows the format provide...
    --cli-input-yaml       [string] Reads arguments
from the YAML string provided. The YAML string follows the format provide...
    --generate-cli-skeleton [string] Prints a JSON
skeleton to standard output without sending an API request. If provided wit...
```

リソース補完

自動プロンプトは、利用可能なAWSリソースプロパティを使用してAWSAPI呼び出しを行い、リソース値を提案します。これは、パラメータの入力時に、自動プロンプトが使用可能なユーザー所有のリソースを提案することを可能にします。以下の例では、`--table-name` コマンドの `aws dynamodb describe-table` パラメータを入力するときに、自動プロンプトがテーブル名をリストアップします。

```
$ aws dynamodb describe-table
> aws dynamodb describe-table --table-name
                                     Table1
                                     Table2
                                     Table3
```

短縮構文補完

短縮構文を使用するパラメータには、自動プロンプトは使用する値を提案します。以下の例では、自動プロンプトが `--placement` コマンドの `aws ec2 run-instances` パラメータに対する短縮構文をリストアップします。

```
$ aws ec2 run-instances
> aws ec2 run-instances --placement
AvailabilityZone= [string] The Availability Zone of the instance. If not
specified, an Availability Zone wil...
Affinity= [string] The affinity setting for the instance on the
Dedicated Host. This parameter is no...
GroupName= [string] The name of the placement group the instance is in.
PartitionNumber= [integer] The number of the partition the instance is in.
Valid only if the placement grou...
```

ファイル補完

`aws` コマンドのパラメータを入力するときに、自動入力が `file://` または `fileb://` プレフィックスを使用した後に続くローカルファイル名を提案します。以下の例では、`--item file://` コマンドに `aws ec2 run-instances` を入力した後で、自動プロンプトがローカルファイルを提案します。

```
$ aws ec2 run-instances
> aws ec2 run-instances --item file://
                                     item1.txt
```

```
file1.json  
file2.json
```

リージョン補完

グローバルパラメータの `--region` を使用するときには、自動プロンプトが選択可能なリージョンをリストアップします。以下の例では、`aws dynamodb list-tables` コマンドの `--region` を入力した後で、自動プロンプトがリージョンをアルファベット順に提案します。

```
$ aws dynamodb list-tables  
> aws dynamodb list-tables --region  
af-south-1  
ap-east-1  
ap-northeast-1  
ap-northeast-2
```

プロファイル補完

グローバルパラメータの `--profile` を使用するときには、自動プロンプトがプロファイルをリストアップします。以下の例では、`--profile` コマンドに `aws dynamodb list-tables` を入力した後で、自動プロンプトがユーザーのプロファイルを提案します。

```
$ aws dynamodb list-tables  
> aws dynamodb list-tables --profile  
profile1  
profile2  
profile3
```

あいまい検索

特定の文字セットを含むコマンドと値を補完します。以下の例では、`aws dynamodb list-tables` コマンドに `--region eu` を入力した後で、自動プロンプトが `eu` が含まれるリージョンを提案します。

```
$ aws dynamodb list-tables  
> aws dynamodb list-tables --region west  
eu-west-1  
eu-west-2  
eu-west-3  
us-west-1
```

履歴

自動プロンプトモードで以前に使用したコマンドを表示して実行するには、CTRL+ R を押しします。履歴には、矢印キーを使用して選択できる以前のコマンドが一覧表示されます。以下の例では、自動プロンプトモードの履歴が表示されています。

```
$ aws
> aws
    dynamodb list-tables
    s3 ls
```

自動プロンプトモード

AWS CLI バージョン 2 の自動プロンプトには、設定できる 2 つのモードがあります。

- フルモード: `aws` パラメータを使用して手動で呼び出すか、永続的に有効化したかにかかわらず、`--cli-auto-prompt` コマンドを実行しようとするたびに自動プロンプトを使用します。これには、完全なコマンドまたは不完全なコマンドの両方のENTER後に を押すことが含まれます。
- 部分モード: コマンドが不完全であるか、クライアント側の検証エラーのために実行できない場合に自動プロンプトを使用します。このモードは、既存のスクリプトまたはランブックがある場合、あるいはすべてのコマンドに対してプロンプトを表示するのではなく、不慣れなコマンドにのみ自動プロンプトを表示したい場合に特に便利です。

自動プロンプトの設定

自動プロンプトを設定するには、次の方法を優先順に使用することができます。

- コマンドラインオプションは、単一のコマンドに対して自動プロンプトを有効化または無効化します。`--cli-auto-prompt` を使用して自動プロンプトを呼び出し、`--no-cli-auto-prompt` を使用して自動プロンプトを無効化してください。
- 環境変数では、`aws_cli_auto_prompt` 変数が使用されます。
- 共有設定ファイルでは、`cli_auto_prompt` 設定が使用されます。

でのコマンド出力の制御 AWS CLI

このセクションでは、AWS Command Line Interface (AWS CLI) からの出力を制御するさまざまな方法を示します。ターミナルの AWS CLI 出力をカスタマイズすることで、読みやすさが向上し、スクリプトの自動化が合理化され、大規模なデータセットをより簡単にナビゲートできます。

は、[、](#)、[json](#)、[textyaml](#) および を含む複数の [出力形式](#) AWS CLI をサポートしています [table](#)。一部のサービスでは、データに対してサーバー側の [ページ分割](#) が行われており、AWS CLI には追加の [ページ分割オプション](#) 用の独自のクライアント側の機能が用意されています。

最後に、AWS CLI には [サーバー側とクライアント側のフィルタリング](#) の両方があり、個別に、または一緒に使用して出力をフィルタリングできます AWS CLI。

トピック

- [機密出力](#)
- [サーバー側の出力オプションとクライアント側の出力オプション](#)
- [での出力形式の設定 AWS CLI](#)
- [でのページ分割オプションの使用 AWS CLI](#)
- [での出力のフィルタリング AWS CLI](#)

機密出力

のオペレーションによっては、環境変数からの情報など、機密性が高いと見なされる可能性のある情報を返 AWS CLI す場合があります。この情報の公開は、特定のシナリオでセキュリティリスクを表す可能性があります。例えば、情報は継続的統合と継続的デプロイ (CI/CD) ログに含まれる可能性があります。したがって、ログの一部としてこのような出力を含めるときは を確認し、不要な場合は出力を抑制することが重要です。

機密データの保護の詳細については、「[」](#)を参照してください [the section called “データ保護”](#)。

以下のベストプラクティスを考慮します。

- などのシークレットストアからシークレットをプログラムで取得することを検討してください AWS Secrets Manager。
- ビルドログの内容を確認して、機密情報が含まれていないことを確認します。コマンド出力を抑制するために、出力を [へのパイプ/dev/null](#) や [bash](#) または [PowerShell](#) 変数としてキャプチャするなどのアプローチを検討してください。

以下は、エラーではなく出力を `/dev/null` にリダイレクトするための `bash` の例です。

```
$ aws s3 ls > /dev/null
```

ターミナルの出力の抑制の詳細については、使用するターミナルのユーザードキュメントを参照してください。

- ログへのアクセスを検討し、ユースケースに応じてアクセスの範囲を適切に指定します。

サーバー側の出力オプションとクライアント側の出力オプション

AWS CLI には [サーバー側とクライアント側のフィルタリング](#) の両方があり、個別に、またはまとめて出力をフィルタリングできます AWS CLI。サーバー側のフィルタリングが最初に処理され、クライアント側のフィルタリングのために出力が返されます。サーバー側のフィルタリングは、サービスでサポートされています API。クライアント側のフィルタリングは、`--query` パラメータを使用して AWS CLI クライアントによってサポートされます。

サーバー側の出力オプションは、が直接サポートする機能です AWS のサービス API。フィルタリングまたはページアウトされたデータはクライアントに送信されないため、HTTP 応答時間が短縮され、より大きなデータセットの帯域幅が向上します。

クライアント側の出力オプションは、AWS CLI によって作成される機能です。すべてのデータはクライアントに送信され、次に、表示されたコンテンツを AWS CLI フィルターまたはページ化します。クライアント側のオペレーションでは、大規模なデータセットの速度や帯域幅を節約することができません。

サーバー側のオプションとクライアント側のオプションを同時に使用する場合、サーバー側のオペレーションが最初に完了し、その後クライアント側のオペレーションのためにクライアントに送信されます。これにより、サーバー側のオプションによる潜在的な速度の向上と帯域幅の節約を実現しながら、追加の AWS CLI 機能を使用して必要な出力を取得できます。

での出力形式の設定 AWS CLI

このトピックでは、AWS Command Line Interface () のさまざまな出力形式について説明します AWS CLI。AWS CLI は以下の出力形式をサポートしています。

- [json](#) – 出力は [JSON](#) 文字列としてフォーマットされます。
- [yaml](#) – 出力は [YAML](#) 文字列としてフォーマットされます。

- **yaml-stream** - 出力はストリーミングされ、**YAML**文字列としてフォーマットされます。ストリーミングにより、大きなデータタイプの処理を高速化できます。
- **text** - 出力は、複数行のタブ区切りの文字列値としてフォーマットされます。これは、grep、sed、または awk などのテキストプロセッサに出力を渡すのに役立ちます。
- **table** - 出力は、テーブルとしてフォーマットされ、文字の「+|-」を使用してセルの境界を形成します。通常、情報は他の形式よりも読みやすい「わかりやすい」形式で表示されますが、プログラムとしては役立ちません。

出力形式を選択する方法

「[設定](#)」トピックで説明したように、出力形式は 3 つの異なる方法で指定できます。

- **output** ファイル内の名前付きプロファイルで**config**オプションを使用する - 次の例では、デフォルトの出力形式を text に設定します。

```
[default]
output=text
```

- **AWS_DEFAULT_OUTPUT**環境変数の使用 - 次の出力は、変数が変更されるか、セッションが終了するまで、このコマンドラインセッションのコマンドの形式を table に設定します。この環境変数を使用すると、config ファイルで設定された値が上書きされます。

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- コマンドラインで「**--output**」オプションを使用 - 次の例では、この 1 つのコマンドのみの出力を json に設定します。このコマンドでこのオプションを使用すると、現在設定されている環境変数または config ファイルの値をオーバーライドします。

```
$ aws swf list-domains --registration-status REGISTERED --output json
```

Important

指定する出力タイプによって、**--query** オプションの動作が変更されます。

- を指定すると**--output text**、**--query**フィルターが適用される前に出力がページ分割され、は出力の各ページでクエリを 1 回 AWS CLI 実行します。このため、クエリには各ページで最初に一致する要素が含まれており、予期しない余分な出力が発生する可能性が

あります。出力をさらにフィルタリングするには、`head` や `tail` などの他のコマンドラインツールを使用できます。

- `--output json`、`--output yaml`、`--output yaml-stream` を指定する場合、単一のネイティブな構造として完全に処理されてから `--query` フィルターが適用されます。は、構造全体に対してクエリを 1 回だけ AWS CLI 実行し、フィルタリングされた結果を生成して出力します。

JSON 出力形式

[JSON](#) は、のデフォルトの出力形式です AWS CLI。ほとんどのプログラミング言語は、組み込み関数または公開されているライブラリを使用して JSON 文字列を簡単にデコードできます。JSON 出力を [--query オプション](#) と組み合わせると、フォーマット AWS CLI JSON された出力をフィルタリングおよびフォーマットする強力な方法があります。

では実行できない高度なフィルタリングについては `--query`、コマンドライン JSON プロセッサ `jq` である を検討してください。これをダウンロードし、公式のチュートリアルを <http://stedolan.github.io/jq/> で見るすることができます。

JSON 出力の例を次に示します。

```
$ aws iam list-users --output json
```

```
{
  "Users": [
    {
      "Path": "/",
      "UserName": "Admin",
      "UserId": "AIDA111111111111EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/Admin",
      "CreateDate": "2014-10-16T16:03:09+00:00",
      "PasswordLastUsed": "2016-06-03T18:37:29+00:00"
    },
    {
      "Path": "/backup/",
      "UserName": "backup-user",
      "UserId": "AIDA222222222222EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/backup/backup-user",
      "CreateDate": "2019-09-17T19:30:40+00:00"
    },
  ],
}
```

```
{
  "Path": "/",
  "UserName": "cli-user",
  "UserId": "AIDA333333333333EXAMPLE",
  "Arn": "arn:aws:iam::123456789012:user/cli-user",
  "CreateDate": "2019-09-17T19:11:39+00:00"
}
]
```

YAML 出力形式

[YAML](#) は、形式の[YAMLテンプレート](#) AWS CloudFormation のサポートなど、[YAML](#)形式の文字列を出力または消費するサービスやツールを使用して、プログラムで出力を処理する場合に適しています。

では実行できない高度なフィルタリングについては--query、コマンドラインYAMLプロセッサyqである を検討してください。yq の [yq リポジトリ](#)でダウンロードできますGitHub。

YAML 出力の例を次に示します。

```
$ aws iam list-users --output yaml
```

```
Users:
- Arn: arn:aws:iam::123456789012:user/Admin
  CreateDate: '2014-10-16T16:03:09+00:00'
  PasswordLastUsed: '2016-06-03T18:37:29+00:00'
  Path: /
  UserId: AIDA111111111111EXAMPLE
  UserName: Admin
- Arn: arn:aws:iam::123456789012:user/backup/backup-user
  CreateDate: '2019-09-17T19:30:40+00:00'
  Path: /backup/
  UserId: AIDA222222222222EXAMPLE
  UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
- Arn: arn:aws:iam::123456789012:user/cli-user
  CreateDate: '2019-09-17T19:30:40+00:00'
  Path: /
  UserId: AIDA333333333333EXAMPLE
  UserName: cli-user
```

YAML ストリーム出力形式

このyaml-stream形式は、[YAML](#)形式を利用すると同時に、データをストリーミングすることで、大きなデータセットをより応答性が高く、より高速に表示できます。クエリ全体がダウンロードされる前に、YAMLデータの表示と使用を開始できます。

では実行できない高度なフィルタリングについては--query、コマンドラインYAMLプロセッサyqである を検討してください。yq の [yq リポジトリ](#)でダウンロードできますGitHub。

yaml-stream 出力例を次に示します。

```
$ aws iam list-users --output yaml-stream
```

```
- IsTruncated: false
  Users:
  - Arn: arn:aws:iam::123456789012:user/Admin
    CreateDate: '2014-10-16T16:03:09+00:00'
    PasswordLastUsed: '2016-06-03T18:37:29+00:00'
    Path: /
    UserId: AIDA111111111111EXAMPLE
    UserName: Admin
  - Arn: arn:aws:iam::123456789012:user/backup/backup-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /backup/
    UserId: AIDA222222222222EXAMPLE
    UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
  - Arn: arn:aws:iam::123456789012:user/cli-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /
    UserId: AIDA333333333333EXAMPLE
    UserName: cli-user
```

以下は、--page-sizeパラメータを使用してストリーミングされたYAMLコンテンツをページ分割することに伴うyaml-stream出力の例です。

```
$ aws iam list-users --output yaml-stream --page-size 2
```

```
- IsTruncated: true
  Marker: ab1234cdef5ghi67jk8lmo9p/
q012rs3t445uv6789w0x1y2z/345a6b78c9d00/1efgh234ij56klmno78pqrstu90vwxyx
  Users:
```

```

- Arn: arn:aws:iam::123456789012:user/Admin
  CreateDate: '2014-10-16T16:03:09+00:00'
  PasswordLastUsed: '2016-06-03T18:37:29+00:00'
  Path: /
  UserId: AIDA111111111111EXAMPLE
  UserName: Admin
- Arn: arn:aws:iam::123456789012:user/backup/backup-user
  CreateDate: '2019-09-17T19:30:40+00:00'
  Path: /backup/
  UserId: AIDA222222222222EXAMPLE
  UserName: arq-45EFD6D1-CE56-459B-B39F-F9C1F78FBE19
- IsTruncated: false
  Users:
  - Arn: arn:aws:iam::123456789012:user/cli-user
    CreateDate: '2019-09-17T19:30:40+00:00'
    Path: /
    UserId: AIDA333333333333EXAMPLE
    UserName: cli-user

```

テキストの出力形式

このtext形式は、AWS CLI 出力をタブ区切りの行に整理します。grep、などの従来の Unix テキストツールやsedawk、によって実行されるテキスト処理とうまく連携します PowerShell。

text 出力形式は、以下に示す基本的な構造に従います。列は、基になるJSONオブジェクトの対応するキー名でアルファベット順にソートされます。

```

IDENTIFIER sorted-column1 sorted-column2
IDENTIFIER2 sorted-column1 sorted-column2

```

text 出力例を次に示します。各フィールドは他のフィールドからタブで区切られ、空のフィールドがある追加のタブが含まれます。

```
$ aws iam list-users --output text
```

```

USERS arn:aws:iam::123456789012:user/Admin 2014-10-16T16:03:09+00:00
      2016-06-03T18:37:29+00:00 / AIDA111111111111EXAMPLE Admin
USERS arn:aws:iam::123456789012:user/backup/backup-user 2019-09-17T19:30:40+00:00
      /backup/ AIDA222222222222EXAMPLE backup-user
USERS arn:aws:iam::123456789012:user/cli-user 2019-09-17T19:11:39+00:00
      / AIDA333333333333EXAMPLE cli-user

```

4 番目の列は PasswordLastUsed フィールドで、最後の 2 つのエントリは空です。これらのユーザーは AWS Management Console にサインインしないためです。

⚠ Important

`text` 出力を指定する場合は、`--query` オプションも必ず使用して、一貫した動作を確保することを強くお勧めします。

これは、テキスト形式が、AWS サービスによって返される基盤となる JSON オブジェクトのキー名で出力列をアルファベット順に順序付けし、同様のリソースが同じキー名を持たない可能性があるためです。例えば、Linux ベースの Amazon EC2 インスタンスの JSON 表現には、Windows ベースのインスタンスの JSON 表現に存在しない要素がある場合もあれば、その逆もあります。また、リソースのキー値要素が将来の更新で追加または削除されて、列の順序が変わる可能性があります。このような場合、`--query` は `text` 出力の機能を補強して、出力形式に対する完全な制御を提供します。

次の例では、コマンドは表示する要素を指定し、列の順序をリスト表記 [`key1`, `key2`, ...] で定義します。これにより、正しいキー値が常に予期される列に表示されることを確信できます。最後に、が存在しないキーの値 `None` として AWS CLI 出力する方法に注意してください。

```
$ aws iam list-users --output text --query 'Users[*].
[UserName,Arn,CreateDate,PasswordLastUsed,UserId]'
```

```
Admin          arn:aws:iam::123456789012:user/Admin
2014-10-16T16:03:09+00:00  2016-06-03T18:37:29+00:00  AIDA111111111111EXAMPLE
backup-user    arn:aws:iam::123456789012:user/backup-user
2019-09-17T19:30:40+00:00  None                        AIDA222222222222EXAMPLE
cli-user       arn:aws:iam::123456789012:user/cli-backup
2019-09-17T19:11:39+00:00  None                        AIDA333333333333EXAMPLE
```

以下の例では、`grep` および `awk` を `text` コマンドからの `aws ec2 describe-instances` 出力で使用方法を示しています。最初のコマンドは各インスタンスのアベイラビリティゾーン、現在の状態、およびインスタンス ID を `text` 出力で表示します。2 番目のコマンドは、`us-west-2a` アベイラビリティゾーンで実行中のすべてのインスタンス ID のインスタンスのみを表示するように出力を処理します。

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text
```

```
us-west-2a      running i-4b41a37c
us-west-2a      stopped i-a071c394
us-west-2b      stopped i-97a217a0
us-west-2a      running i-3045b007
us-west-2a      running i-6fc67758
```

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text | grep us-west-2a |
grep running | awk '{print $3}'
```

```
i-4b41a37c
i-3045b007
i-6fc67758
```

次の例は、さらに一步踏み込んで、出力をフィルタリングする方法だけでなく、その出力を使用して、停止した各インスタンスのインスタンスタイプの変更を自動化する方法も示しています。

```
$ aws ec2 describe-instances --query 'Reservations[*].Instances[*].[State.Name,
InstanceId]' --output text |
> grep stopped |
> awk '{print $2}' |
> while read line;
> do aws ec2 modify-instance-attribute --instance-id $line --instance-type '{"Value":
"m1.medium"}';
> done
```

text 出力は、でも役立ちます PowerShell。text 出力の列はタブ区切りであるため、PowerShell の `t` 区切り文字を使用して出力を配列に簡単に分割できます。次のコマンドは、最初の列 (InstanceId) が文字列 AvailabilityZone に一致する場合に 3 列目 (us-west-2a) の値を表示します。

```
PS C:\>aws ec2 describe-instances --query 'Reservations[*].Instances[*].
[Placement.AvailabilityZone, State.Name, InstanceId]' --output text |
%{if ($_.split("`t")[0] -match "us-west-2a") { $_.split("`t")[2]; } }
```

```
-4b41a37c
```

```
i-a071c394
i-3045b007
i-6fc67758
```

前の例では、`--query`パラメータを使用して基盤となるJSONオブジェクトを解析し、目的の列をプルアウトする方法を示していますが、クロスプラットフォームの互換性が問題でない場合はJSON、を処理する独自の機能 PowerShell があることに注意してください。ほとんどのコマンドシェルが必要とするように、出力をテキストとして扱う代わりに、`ConvertFrom-Json` コマンドレット PowerShell を使用して階層構造化オブジェクトを生成します。その後、そのオブジェクトから必要なメンバーに直接アクセスできます。

```
(aws ec2 describe-instances --output json | ConvertFrom-
Json).Reservations.Instances.InstanceId
```

Tip

テキスト出力を行い、`--query` パラメータを使用して出力を単一のフィールドにフィルタリングすると、出力は 1 行のタブ区切り値になります。次の例に示すように、各値を別々の行に入れるには、出力フィールドを角括弧で囲みます。

タブ区切りの単一の行の出力

```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].GroupName"
```

```
HRDepartment    Developers      SpreadsheetUsers  LocalAdmins
```

`[GroupName]` を角括弧で囲むことで、各値を 1 行におさめることができます。

```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].[GroupName]"
```

```
HRDepartment
Developers
SpreadsheetUsers
LocalAdmins
```


テーブルの出力形式

table 形式は、複雑な AWS CLI 出力を人間が読み取れる表現で、表形式で生成します。

```
$ aws iam list-users --output table
```

```
-----
|
| ListUsers |
+-----+
|
||
| Users |
+-----+
|+-----+-----+-----+-----+
+-----+-----+-----+-----+
||                Arn | CreateDate |
PasswordLastUsed | Path | UserId | Username ||
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|| arn:aws:iam::123456789012:user/Admin | 2014-10-16T16:03:09+00:00 |
2016-06-03T18:37:29+00:00 | / | AIDA111111111111EXAMPLE | Admin ||
|| arn:aws:iam::123456789012:user/backup/backup-user | 2019-09-17T19:30:40+00:00 | |
| /backup/ | AIDA222222222222EXAMPLE | backup-user ||
|| arn:aws:iam::123456789012:user/cli-user | 2019-09-17T19:11:39+00:00 |
| / | AIDA333333333333EXAMPLE | cli-user ||
+-----+-----+-----+-----+
+
+

```

--query オプションを table 形式と組み合わせて、raw 出力から事前に選択された要素のセットを表示することができます。デクシヨナリ表記とリスト表記の出力の違いに注意してください。最初の例では、列名はアルファベット順ですが、2 番目の例では、名前のない列がユーザーによって定義された順序になっています。--query オプションの詳細については、「[での出力のフィルタリング AWS CLI](#)」を参照してください。

```
$ aws ec2 describe-volumes --query 'Volumes[*].
{ID:VolumeId,InstanceId:Attachments[0].InstanceId,AZ:AvailabilityZone,Size:Size}' --
output table
```

```
-----
|
| DescribeVolumes |
+-----+

```

AZ	ID	InstanceId	Size
us-west-2a	vol-e11a5288	i-a071c394	30
us-west-2a	vol-2e410a47	i-4b41a37c	8

```
$ aws ec2 describe-volumes --query 'Volumes[*].
[VolumeId,Attachments[0].InstanceId,AvailabilityZone,Size]' --output table
```

```
-----
| DescribeVolumes |
+-----+-----+-----+-----+
| vol-e11a5288| i-a071c394 | us-west-2a | 30 |
| vol-2e410a47| i-4b41a37c | us-west-2a | 8 |
+-----+-----+-----+-----+
```

でのページ分割オプションの使用 AWS CLI

このトピックでは、AWS CLIからの出力をページネーションするためのさまざまな方法を示します。

AWS CLIからページネーションを制御する方法は主に 2 つあります。

- [サーバー側のページ分割パラメータの使用。](#)
- [出力用のデフォルトのクライアント側のページングプログラムの使用。](#)

サーバー側のページ分割パラメータが最初に処理され、出力がクライアント側のページ分割に送信されます。

サーバー側のページ分割

項目の大きなリストを返すことができるコマンドの場合、AWS Command Line Interface (AWS CLI) には、`aws` がサービスの API を AWS CLI 呼び出してリストに入力するとき出力に含まれる項目の数を制御するための複数のオプションがあります。

オプションは以下のとおりです。

- [--no-paginate パラメータの使用法](#)
- [--page-size パラメータの使用法](#)
- [--max-items パラメータの使用法](#)

- [--starting-token](#) パラメータの使用方法

デフォルトでは、は個々のサービスによって決定されるページサイズ AWS CLI を使用し、使用可能なすべての項目を取得します。例えば、Amazon S3 では、デフォルトのページサイズは 1,000 です。3,500 のオブジェクトを含む Amazon S3 バケットで `aws s3api list-objects` を実行すると、AWS CLI は Simple Storage Service (Amazon S3) に対して 4 つの呼び出しを自動的に実行し、サービス固有の分割ロジックをバックグラウンドで処理して、最終的な出力で 3,500 オブジェクトのすべてを返します。

--no-paginate パラメータの使用方法

--no-paginate オプションでは、クライアント側で分割トークンの追従を無効にします。コマンドを使用する場合、デフォルトで、はページ分割を作成するために可能なすべての結果を返すために AWS CLI、自動的に複数の呼び出しを行います。ページごとに 1 回の呼び出し。ページ分割を無効にすると、コマンド結果の最初のページに対して 1 回 AWS CLI だけ呼び出しが行われます。

例えば、3,500 個のオブジェクトを含む Amazon S3 バケット `aws s3api list-objects` で を実行する場合、は Amazon S3 への最初の呼び出し AWS CLI のみを行い、最終出力の最初の 1,000 個のオブジェクトのみを返します。

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --no-paginate  
{  
  "Contents": [  
  ...
```

--page-size パラメータの使用方法

大量のリソースに対してリストコマンドを実行しているときに問題が発生する場合は、デフォルトのページサイズが大きすぎる可能性があります。これにより、AWS サービスへの呼び出しが最大許容時間を超え、「タイムアウト」エラーが発生する可能性があります。--page-size オプションを使用して、AWS サービスへの各呼び出しから、より少数の項目を AWS CLI リクエストするように指定できます。AWS CLI は引き続き完全なリストを取得しますが、バックグラウンドでより多くのサービスAPI呼び出しを実行し、呼び出しごとに少数の項目を取得します。このため、個々の呼び出しがタイムアウトにならずに成功する可能性が高くなります。ページサイズを変更しても出力には影響しません。出力の生成に必要なAPI呼び出しの数にのみ影響します。

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --page-size 100
```

```
--page-size 100
{
  "Contents": [
  ...
```

--max-items パラメータの使用方法

AWS CLI 一度に出力に含める項目を減らすには、`--max-items` オプションを使用します。は、前述のように サービスのページ分割 AWS CLI を引き続き処理しますが、指定した一度に出力される項目の数のみを指定します。

```
$ aws s3api list-objects \
  --bucket amzn-s3-demo-bucket \
  --max-items 100
{
  "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==",
  "Contents": [
  ...
```

--starting-token パラメータの使用方法

出力される項目の数 (`--max-items`) が基盤となる API 呼び出しによって返される項目の合計数よりも少ない `NextToken` 場合、出力には、次の項目セットを取得するために後続のコマンドに渡すことができるが含まれます。次の例は、前の例で返された `NextToken` 値を使用して、2 番目の 100 項目を取得する方法を示しています。

Note

パラメータ `--starting-token` を null または空にすることはできません。前のコマンドが `NextToken` 値を返さない場合、返す項目はこれ以上存在せず、このコマンドを再度呼び出す必要はありません。

```
$ aws s3api list-objects \
  --bucket amzn-s3-demo-bucket \
  --max-items 100 \
  --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==
{
  "Contents": [
  ...
```

指定された AWS サービスは、 を呼び出すたびに同じ順序で項目を返さない場合があります。 --page-size と --max-items に異なる値を指定した場合、項目の不足や重複など、予期しない結果になることがあります。これを防ぐには、 --page-size と --max-items に同じ数を使用して、AWS CLI のページ分割と基本のサービスを同期させます。リスト全体を取得し、必要な解析オペレーションをローカルで実行することもできます。

クライアント側のページャー

AWS CLI バージョン 2 では、出力にクライアント側のページャープログラムを使用できます。デフォルトでは、この機能はオペレーティングシステムのデフォルトのページャープログラムを介してすべての出力を返します。

次の方法を優先順に使用して、出力ページャーを指定できます。

- default または名前付きプロファイルの config ファイルの cli_pager 設定を使用する。
- AWS_PAGER 環境変数を使用する。
- PAGER 環境変数を使用する。

次の方法を優先順に使用して、外部ページ分割プログラムのすべての使用を無効にすることができます。

- --no-cli-pager コマンドラインオプションを使用して 1 回のコマンド使用のページャーを無効にする。
- cli_pager 設定または AWS_PAGER 変数を空の文字列に設定する。

クライアント側のページャーのトピック:

- [cli_pager 設定の使用方法](#)
- [AWS_PAGER 環境変数の設定方法](#)
- [--no-cli-pager オプションの使用方法](#)
- [ページャーフラグの使用方法](#)

cli_pager 設定の使用方法

頻繁に利用される構成設定および認証情報を AWS CLI が維持するファイルに保存することができます。名前プロファイルの設定は、default プロファイルの設定よりも優先されます。構成設定の詳細については、「[の設定と認証情報ファイルの設定 AWS CLI](#)」を参照してください。

次の例では、デフォルトの出カページャーを `less` プログラムに設定します。

```
[default]
cli_pager=less
```

以下の例では、ページャーの使用を無効にするようにデフォルトを設定します。

```
[default]
cli_pager=
```

AWS_PAGER 環境変数の設定方法

次の例では、デフォルトの出カページャーを `less` プログラムに設定します。環境変数の詳細については、「[の環境変数の設定 AWS CLI](#)」を参照してください。

Linux and macOS

```
$ export AWS_PAGER="less"
```

Windows

```
C:\> setx AWS_PAGER "less"
```

--no-cli-pager オプションの使用方法

1 つのコマンドでのページャーの使用を無効にするには、`--no-cli-pager` オプションを使用します。コマンドラインオプションの詳細については、「[のコマンドラインオプション AWS CLI](#)」を参照してください。

```
$ aws s3api list-objects \
  --bucket amzn-s3-demo-bucket \
  --no-cli-pager
{
  "Contents": [
  ...
```

ページャーフラグの使用方法

ページ分割プログラムで自動的に使用するフラグを指定できます。フラグは、使用するページ分割プログラムによって異なります。次に示すのは、less および more の一般的なデフォルトの例です。

Linux and macOS

特に指定しない場合、ページャー AWS CLI バージョン 2 はデフォルトで less を使用します。LESS 環境変数が設定されていない場合、AWS CLI バージョン 2 は FRX フラグを使用します。フラグは、AWS CLI ページャーを設定するときに指定することで組み合わせることができます。

次の例では、S フラグを使用しています。このフラグはデフォルトの FRX フラグと組み合わせられて、最終的な FRXS フラグを作成します。

```
$ export AWS_PAGER="less -S"
```

FRX フラグのいずれも必要ない場合は、これらを無効にすることができます。次の例では、F フラグを無効にして、最終的な RX フラグを作成します。

```
$ export AWS_PAGER="less -+F"
```

less フラグの詳細については、mandpages.org の「[less](http://mandpages.org)」を参照してください。

Windows

特に指定しない場合、ページャー AWS CLI バージョン 2 がデフォルトで使用するには more、追加のフラグはありません。

次の例では、/c パラメータを使用しています

```
C:\> setx AWS_PAGER "more /c"
```

more フラグの詳細については、Microsoft Docs [more](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/more) の「」を参照してください。

での出力のフィルタリング AWS CLI

AWS Command Line Interface (AWS CLI) には、サーバー側とクライアント側のフィルタリングの両方があり、個別に、または一緒に使用して AWS CLI 出力をフィルタリングできます。サーバー側のフィルタリングが最初に処理され、クライアント側のフィルタリングのために出力が返されます。

- サーバー側のフィルタリングは、サポートされておりAPI、通常は `--filter` パラメータを使用して実装します。このサービスは、大規模なデータセットのHTTP応答時間を短縮できる一致する結果のみを返します。
- クライアント側のフィルタリングは、`--query` パラメータを使用して AWS CLI クライアントによってサポートされます。このパラメータには、サーバー側のフィルタリングにはない可能性がある機能があります。

トピック

- [サーバー側のフィルタリング](#)
- [クライアント側のフィルタリング](#)
- [サーバー側とクライアント側のフィルタリングを組み合わせる](#)
- [追加リソース](#)

サーバー側のフィルタリング

のサーバー側のフィルタリング AWS CLI は、AWS サービスによって提供されますAPI。このAWS サービスは、フィルターに一致するHTTPレスポンス内のレコードのみを返します。これにより、大規模なデータセットのHTTPレスポンス時間が短縮される可能性があります。サーバー側のフィルタリングはサービスによって定義されるためAPI、パラメータ名と関数はサービスによって異なります。フィルタリングに使用される一般的なパラメータ名は次のとおりです。

- `--filter` ([ses](#)、[ce](#)など)。
- `--filters` ([ec2](#)、[autoscaling](#)、[rds](#)など)。
- 単語「filter」で始まる名前 ([aws dynamodb scan](#) コマンドの `--filter-expression` など)。

特定のコマンドにサーバー側のフィルタリングとフィルタリングルールがあるかどうかについては、[AWS CLI バージョン 2 リファレンスガイド](#) を参照してください。

クライアント側のフィルタリング

AWS CLI には、`--query` パラメータを使用した組み込みJSONのベースのクライアント側のフィルタリング機能があります。`--query` パラメータは、出力の内容とスタイルをカスタマイズするために使用できる強力なツールです。`--query` パラメータは、サーバーから返されるHTTPレスポンスを受け取り、結果を表示する前にフィルタリングします。フィルタリングの前にHTTPレスポンス全

体がクライアントに送信されるため、クライアント側のフィルタリングは、大規模なデータセットのサーバー側のフィルタリングよりも遅くなる可能性があります。

クエリは[JMESPath構文](#)を使用して、出力をフィルタリングするための式を作成します。JMESPath構文については、JMESPathウェブサイトの「[チュートリアル](#)」を参照してください。

Important

指定する出力タイプによって、`--query` オプションの動作が変更されます。

- `text` を指定すると `--output text`、出力は `--query` フィルターが適用される前にページ分割され、は出力の各ページでクエリを 1 回 AWS CLI 実行します。このため、クエリには各ページで最初に一致する要素が含まれており、予期しない余分な出力が発生する可能性があります。出力をさらにフィルタリングするには、`head` や `tail` などの他のコマンドラインツールを使用できます。
- `json`、`yaml`、`yaml-stream` を指定する場合、単一のネイティブな構造として完全に処理されてから `--query` フィルターが適用されます。は、構造全体に対してクエリを 1 回だけ AWS CLI 実行し、フィルタリングされた結果を生成して出力します。

クライアント側のフィルタリングのトピック

- [開始する前に](#)
- [識別子](#)
- [リストから選択する](#)
- [ネストされたデータをフィルタリングする](#)
- [結果をフラット化する](#)
- [特定の値をフィルタリングする](#)
- [パイピング式](#)
- [複数の識別子の値をフィルタリングする](#)
- [識別子の値にラベルを追加する](#)
- [関数](#)
- [高度な `--query` の例](#)

開始する前に

これらの例で使用されているフィルター式を使用する場合は、ターミナルシェルに正しい引用規則を使用してください。詳細については、「[the section called “文字列を含む引用符”](#)」を参照してください。

次のJSON出力は、`--query` パラメータが生成できる内容の例を示しています。出力は、個別の Amazon EC2 インスタンスにアタッチされた 3 つの Amazon EBS ボリュームを記述します。

出力例

```
$ aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-17T00:55:03.000Z",
          "InstanceId": "i-a071c394",
          "VolumeId": "vol-e11a5288",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "VolumeType": "standard",
      "VolumeId": "vol-e11a5288",
      "State": "in-use",
      "SnapshotId": "snap-f23ec1c8",
      "CreateTime": "2013-09-17T00:55:03.000Z",
      "Size": 30
    },
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
        {
          "AttachTime": "2013-09-18T20:26:16.000Z",
          "InstanceId": "i-4b41a37c",
          "VolumeId": "vol-2e410a47",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-2e410a47",
  "State": "in-use",
  "SnapshotId": "snap-708e8348",
  "CreateTime": "2013-09-18T20:26:15.000Z",
  "Size": 8
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-a1b3c7nd",
  "State": "in-use",
  "SnapshotId": "snap-234087fb",
  "CreateTime": "2020-11-20T19:54:05.000Z",
  "Size": 15
}
]
}
```

識別子

識別子は、出力値のラベルです。フィルターを作成するときは、識別子を使用してクエリ結果を絞り込みます。次の出力例では、`Volumes`、`AvailabilityZone`、`AttachTime` などのすべての識別子が強調表示されます。

```
$ aws ec2 describe-volumes
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2a",
      "Attachments": [
```

```
{
  "AttachTime": "2013-09-17T00:55:03.000Z",
  "InstanceId": "i-a071c394",
  "VolumeId": "vol-e11a5288",
  "State": "attached",
  "DeleteOnTermination": true,
  "Device": "/dev/sda1"
}
],
"VolumeType": "standard",
"VolumeId": "vol-e11a5288",
"State": "in-use",
"SnapshotId": "snap-f23ec1c8",
"CreateTime": "2013-09-17T00:55:03.000Z",
"Size": 30
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-2e410a47",
  "State": "in-use",
  "SnapshotId": "snap-708e8348",
  "CreateTime": "2013-09-18T20:26:15.000Z",
  "Size": 8
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
```

```

    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-a1b3c7nd",
"State": "in-use",
"SnapshotId": "snap-234087fb",
"CreateTime": "2020-11-20T19:54:05.000Z",
"Size": 15
}
]
}

```

詳細については、JMESPathウェブサイトの「[識別子](#)」を参照してください。

リストから選択する

リストまたは配列は、[における Volumes や Attachments などの角括弧「[the section called “開始する前に”](#)」の後に続く識別子です。

[Syntax] (構文)

```
<listName>[ ]
```

配列からのすべての出力をフィルタリングするには、ワイルドカード表記を使用できます。[ワイルドカード](#)式は、* 表記法を使用して要素を返すために使用される式です。

次の例では、すべての Volumes コンテンツに対するクエリを実行します。

```

$ aws ec2 describe-volumes \
  --query 'Volumes[*]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ]
  }
]

```

```
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-e11a5288",
    "State": "in-use",
    "SnapshotId": "snap-f23ec1c8",
    "CreateTime": "2013-09-17T00:55:03.000Z",
    "Size": 30
  },
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2020-11-20T19:54:06.000Z",
        "InstanceId": "i-1jd73kv8",
        "VolumeId": "vol-a1b3c7nd",
        "State": "attached",
        "DeleteOnTermination": true,
        "Device": "/dev/sda1"
      }
    ],
    "VolumeType": "standard",
    "VolumeId": "vol-a1b3c7nd",
    "State": "in-use",
    "SnapshotId": "snap-234087fb",
    "CreateTime": "2020-11-20T19:54:05.000Z",
    "Size": 15
  }
]
```

配列内の特定のボリュームをインデックス別に表示するには、配列インデックスを呼び出します。例えば、Volumes 配列の最初の項目のインデックスは 0 で、Volumes[0] クエリが生成されます。配列インデックスの詳細については、JMESPathウェブサイトの「[インデックス式](#)」を参照してください。

```
$ aws ec2 describe-volumes \
  --query 'Volumes[0]'
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
```

```

    "State": "attached",
    "DeleteOnTermination": true,
    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-e11a5288",
"State": "in-use",
"SnapshotId": "snap-f23ec1c8",
"CreateTime": "2013-09-17T00:55:03.000Z",
"Size": 30
}

```

ボリュームの特定範囲をインデックス別に表示するには、次の構文とともに `slice` を使用します。ここで、`start` は開始配列インデックス、`stop` はフィルターが処理を停止するインデックス、`step` はスキップ間隔です。

[Syntax] (構文)

```
<arrayName>[<start>:<stop>:<step>]
```

スライス式からこれらのいずれかを省略すると、次のデフォルト値が使用されます。

- Start - リストの最初のインデックス、0。
- Stop - リストの最後のインデックス。
- Step - ステップスキップなし。値は 1 です。

最初の 2 つのボリュームだけを返すには、次の例に示すように、`start` 値 0、`stop` 値 2、`step` 値 1 を使用します。

```

$ aws ec2 describe-volumes \
  --query 'Volumes[0:2:1]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
      {
        "AttachTime": "2013-09-17T00:55:03.000Z",
        "InstanceId": "i-a071c394",
        "VolumeId": "vol-e11a5288",
        "State": "attached",

```

```
    "DeleteOnTermination": true,
    "Device": "/dev/sda1"
  }
],
"VolumeType": "standard",
"VolumeId": "vol-e11a5288",
"State": "in-use",
"SnapshotId": "snap-f23ec1c8",
"CreateTime": "2013-09-17T00:55:03.000Z",
"Size": 30
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-2e410a47",
  "State": "in-use",
  "SnapshotId": "snap-708e8348",
  "CreateTime": "2013-09-18T20:26:15.000Z",
  "Size": 8
}
]
```

この例にはデフォルト値が含まれているため、スライスを `Volumes[0:2:1]` から `Volumes[:2]` に短縮できます。

次の例では、デフォルト値を省略し、配列全体におけるあらゆる 2 つのボリュームを返します。

```
$ aws ec2 describe-volumes \
  --query 'Volumes[:2]'
[
  {
    "AvailabilityZone": "us-west-2a",
    "Attachments": [
```



```
{
  "AttachTime": "2013-09-17T00:55:03.000Z",
  "InstanceId": "i-a071c394",
  "VolumeId": "vol-e11a5288",
  "State": "attached",
  "DeleteOnTermination": true,
  "Device": "/dev/sda1"
},
{
  "AttachTime": "2020-11-20T19:54:06.000Z",
  "InstanceId": "i-1jd73kv8",
  "VolumeId": "vol-a1b3c7nd",
  "State": "attached",
  "DeleteOnTermination": true,
  "Device": "/dev/sda1"
},
{
  "AttachTime": "2013-09-17T00:55:03.000Z",
  "InstanceId": "i-a071c394",
  "VolumeId": "vol-e11a5288",
  "State": "in-use",
  "SnapshotId": "snap-f23ec1c8",
  "CreateTime": "2013-09-17T00:55:03.000Z",
  "Size": 30
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-a1b3c7nd",
  "State": "in-use",
  "SnapshotId": "snap-234087fb",
  "CreateTime": "2020-11-20T19:54:05.000Z",
  "Size": 15
}
]
```

次の例に示すように、step には、配列の逆の順序でフィルタリングするために負の数を使用することもできます。

```
$ aws ec2 describe-volumes \
  --query 'Volumes[::-2]'
[
```

```
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-a1b3c7nd",
  "State": "in-use",
  "SnapshotId": "snap-234087fb",
  "CreateTime": "2020-11-20T19:54:05.000Z",
  "Size": 15
},
{
  "AvailabilityZone": "us-west-2a",
  "Attachments": [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  "VolumeType": "standard",
  "VolumeId": "vol-e11a5288",
  "State": "in-use",
  "SnapshotId": "snap-f23ec1c8",
  "CreateTime": "2013-09-17T00:55:03.000Z",
  "Size": 30
}
]
```

詳細については、JMESPathウェブサイトの「[スライス](#)」を参照してください。

ネストされたデータをフィルタリングする

ネストされた値の `Volumes[*]` のフィルタリング結果を絞り込むには、ピリオドとフィルター条件を追加して部分式を使用します。

[Syntax] (構文)

```
<expression>.<expression>
```

次の例は、すべてのボリュームのすべての Attachments 情報を示しています。

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments'
[
  [
    {
      "AttachTime": "2013-09-17T00:55:03.000Z",
      "InstanceId": "i-a071c394",
      "VolumeId": "vol-e11a5288",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2013-09-18T20:26:16.000Z",
      "InstanceId": "i-4b41a37c",
      "VolumeId": "vol-2e410a47",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ],
  [
    {
      "AttachTime": "2020-11-20T19:54:06.000Z",
      "InstanceId": "i-1jd73kv8",
      "VolumeId": "vol-a1b3c7nd",
      "State": "attached",
      "DeleteOnTermination": true,
      "Device": "/dev/sda1"
    }
  ]
]
```

```
]
]
```

ネストされた値までさらに絞り込むには、ネストされた各識別子の式を追加します。次の例では、すべての State の Volumes をリストします。

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[*].State'
[
  [
    "attached"
  ],
  [
    "attached"
  ],
  [
    "attached"
  ]
]
```

結果をフラット化する

詳細については、JMESPathウェブサイトの[SubExpressions](#)「」を参照してください。

Volumes[*].Attachments[*].State クエリの結果として生成されたワイルドカード表記を削除することで、Volumes[*].Attachments[].State の結果をフラット化することができます。フラット化は、多くの場合、結果の可読性を向上させるのに役立ちます。

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[].State'
[
  "attached",
  "attached",
  "attached"
]
```

詳細については、JMESPathウェブサイトの[「フラット化」](#)を参照してください。

特定の値をフィルタリングする

リスト内の特定の値をフィルタリングするには、次の構文に示すように、フィルター式を使用します。

[Syntax] (構文)

```
? <expression> <comparator> <expression>]
```

式の比較演算子には、==、!=、<、<=、>、>= などがあります。次の例では、VolumeIds Volumes のすべての Attached の State をフィルタリングします。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId'  
[  
  [  
    "vol-e11a5288"  
  ],  
  [  
    "vol-2e410a47"  
  ],  
  [  
    "vol-a1b3c7nd"  
  ]  
]
```

その後、これをフラット化することができ、次の例になります。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId[]'  
[  
  "vol-e11a5288",  
  "vol-2e410a47",  
  "vol-a1b3c7nd"  
]
```

次の例では、サイズが 20 未満のすべての VolumeIds の Volumes をフィルタリングします。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[?Size < `20`].VolumeId'  
[  
  "vol-2e410a47",  
  "vol-a1b3c7nd"  
]
```

詳細については、JMESPathウェブサイトの「[式をフィルタリングする](#)」を参照してください。

パイピング式

フィルターの結果を新しいリストにパイプ処理し、次の構文を使用して別の式で結果をフィルタリングできます。

[Syntax] (構文)

```
<expression> | <expression>
```

次の例では、`Volumes[*].Attachments[].InstanceId` 式のフィルタリング結果を取得し、配列の最初の結果を出力します。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId | [0]'  
"i-a071c394"
```

この例では、最初に次の式から配列を作成することによってこれを行います。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId'  
"i-a071c394",  
"i-4b41a37c",  
"i-1jd73kv8"
```

その後、その配列の最初の要素を返します。

```
"i-a071c394"
```

詳細については、JMESPathウェブサイトの「[パイプ式](#)」を参照してください。

複数の識別子の値をフィルタリングする

複数の識別子をフィルタリングするには、次の構文を使用して、複数選択リストを使用します。

[Syntax] (構文)

```
<listName>[].[<expression>, <expression>]
```

次の例では、`VolumeId` および `VolumeType` が `Volumes` リストでフィルタリングされ、次の式になります。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[][VolumeId, VolumeType]'  
[  
  [  
    "vol-e11a5288",  
    "standard"  
  ],  
  [  
    "vol-2e410a47",  
    "standard"  
  ],  
  [  
    "vol-a1b3c7nd",  
    "standard"  
  ]  
]
```

ネストされたデータをリストに追加するには、別の複数選択リストを追加します。次の例では、ネストされた InstanceId リストの State および Attachments もフィルタリングすることによって、前の例を拡張します。これにより、式は次のようになります。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[][VolumeId, VolumeType, Attachments[][InstanceId, State]]'  
[  
  [  
    "vol-e11a5288",  
    "standard",  
    [  
      [  
        "i-a071c394",  
        "attached"  
      ]  
    ]  
  ],  
  [  
    "vol-2e410a47",  
    "standard",  
    [  
      [  
        "i-4b41a37c",  
        "attached"  
      ]  
    ]  
  ]  
]
```

```
]
],
[
  "vol-a1b3c7nd",
  "standard",
  [
    [
      "i-1jd73kv8",
      "attached"
    ]
  ]
]
]
```

可読性を向上させるには、次の例に示すように、式をフラット化します。

```
$ aws ec2 describe-volumes \
  --query 'Volumes[][VolumeId, VolumeType, Attachments[][InstanceId, State][]][]'
[
  "vol-e11a5288",
  "standard",
  [
    "i-a071c394",
    "attached"
  ],
  "vol-2e410a47",
  "standard",
  [
    "i-4b41a37c",
    "attached"
  ],
  "vol-a1b3c7nd",
  "standard",
  [
    "i-1jd73kv8",
    "attached"
  ]
]
```

詳細については、JMESPathウェブサイトの「[マルチ選択リスト](#)」を参照してください。

識別子の値にラベルを追加する

この出力を読みやすくするには、次の構文で複数選択ハッシュを使用します。

[Syntax] (構文)

```
<listName>[].{<label>: <expression>, <label>: <expression>}
```

識別子のラベルは、識別子の名前と同じである必要はありません。次の例では、VolumeType 値のラベル VolumeType を使用します。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[].{VolumeType: VolumeType}'  
[  
  {  
    "VolumeType": "standard",  
  },  
  {  
    "VolumeType": "standard",  
  },  
  {  
    "VolumeType": "standard",  
  }  
]
```

簡単にするために、以下の例では、各ラベルの識別子名を保持し、すべてのボリュームの VolumeId、VolumeType、InstanceId、および State を表示します。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[].{VolumeId: VolumeId, VolumeType: VolumeType, InstanceId:  
  Attachments[0].InstanceId, State: Attachments[0].State}'  
[  
  {  
    "VolumeId": "vol-e11a5288",  
    "VolumeType": "standard",  
    "InstanceId": "i-a071c394",  
    "State": "attached"  
  },  
  {  
    "VolumeId": "vol-2e410a47",  
    "VolumeType": "standard",  
    "InstanceId": "i-4b41a37c",  
  }  
]
```

```
    "State": "attached"
  },
  {
    "VolumeId": "vol-a1b3c7nd",
    "VolumeType": "standard",
    "InstanceId": "i-1jd73kv8",
    "State": "attached"
  }
]
```

詳細については、JMESPathウェブサイトの[「マルチセレクトハッシュ」](#)を参照してください。

関数

JMESPath 構文には、クエリに使用できる多くの関数が含まれています。JMESPath 関数の詳細については、JMESPathウェブサイトの[「組み込み関数」](#)を参照してください。

クエリに関数を組み込む方法を示すために、次の例では `sort_by` 関数を使用します。`sort_by` 関数は、次の構文を使用して、ソートキーとして式を使用して配列をソートします。

[Syntax] (構文)

```
sort_by(<listName>, <sort expression>)[].<expression>
```

次の例では、前述の[複数選択ハッシュの例](#)を使用して、出力を `VolumeId` でソートします。

```
$ aws ec2 describe-volumes \  
  --query 'sort_by(Volumes, &VolumeId)[].{VolumeId: VolumeId, VolumeType: VolumeType, InstanceId: Attachments[0].InstanceId, State: Attachments[0].State}'  
[  
  {  
    "VolumeId": "vol-2e410a47",  
    "VolumeType": "standard",  
    "InstanceId": "i-4b41a37c",  
    "State": "attached"  
  },  
  {  
    "VolumeId": "vol-a1b3c7nd",  
    "VolumeType": "standard",  
    "InstanceId": "i-1jd73kv8",  
    "State": "attached"  
  },  
  {
```

```
"VolumeId": "vol-e11a5288",
"VolumeType": "standard",
"InstanceId": "i-a071c394",
"State": "attached"
}
]
```

詳細については、JMESPathウェブサイトの「[sort_by](#)」を参照してください。

高度な `--query` の例

特定のアイテムから情報を抽出するには

次の例では、`--query` パラメータを使用してリスト上の特定の項目を検索し、その項目から情報を抽出します。この例では、指定されたサービスエンドポイントに関連付けられているすべての `AvailabilityZones` をリストします。指定された `ServiceDetails` が含まれる `ServiceName` リストから項目を抽出し、選択されたその項目から `AvailabilityZones` フィールドを出力します。

```
$ aws --region us-east-1 ec2 describe-vpc-endpoint-services \
  --query 'ServiceDetails[?ServiceName==`com.amazonaws.us-
east-1.ecs`].AvailabilityZones'
[
  [
    "us-east-1a",
    "us-east-1b",
    "us-east-1c",
    "us-east-1d",
    "us-east-1e",
    "us-east-1f"
  ]
]
```

指定した作成日より後のスナップショットを表示するには

次の例は、指定された日付以降に作成されたすべてのスナップショットを一覧表示する方法を示しています (例: 出力の利用可能な一部のフィールド)。

```
$ aws ec2 describe-snapshots --owner self \
  --output json \
  --query 'Snapshots[?StartTime>=`2018-02-07`].
{Id:SnapshotId, VId:VolumeId, Size:VolumeSize}'
```

```
[
  {
    "id": "snap-0effb42b7a1b2c3d4",
    "vid": "vol-0be9bb0bf12345678",
    "Size": 8
  }
]
```

最新の を表示するには AMIs

次の例では、作成した最新の Amazon マシンイメージ (AMIs) を、最新のものから古いものまでソートして 5 つ示しています。

```
$ aws ec2 describe-images \
  --owners self \
  --query 'reverse(sort_by(Images,&CreationDate))[:5].{id:ImageId,date:CreationDate}'
[
  {
    "id": "ami-0a1b2c3d4e5f60001",
    "date": "2018-11-28T17:16:38.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60002",
    "date": "2018-09-15T13:51:22.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60003",
    "date": "2018-08-19T10:22:45.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60004",
    "date": "2018-05-03T12:04:02.000Z"
  },
  {
    "id": "ami-0a1b2c3d4e5f60005",
    "date": "2017-12-13T17:16:38.000Z"
  }
]
```

異常な Auto Scaling インスタンスを表示する方法

次の例は、指定した Auto Scaling グループで異常のあるインスタンスのみの InstanceId を示しています。

```
$ aws autoscaling describe-auto-scaling-groups \  
  --auto-scaling-group-name My-AutoScaling-Group-Name \  
  --output text \  
  --query 'AutoScalingGroups[*].Instances[?HealthStatus=='Unhealthy'].InstanceId'
```

指定したタグを持つボリュームを含めるには

次の例では、test タグを持つすべてのインスタンスを説明します。ボリュームにアタッチされた、test ではない別のタグがある限り、ボリュームは結果に含まれて返されます。

以下の式は、test タグを含むすべてのタグを配列に含めて返します。test タグでないタグには null 値が含まれています。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Tags[?Value == `test`]'
```

指定したタグを持つボリュームを除外するには

次の例では、test タグを持たないすべてのインスタンスを説明します。ボリュームは複数のタグを持つことができるため、シンプルな ?Value != `test` 式を使用しても、ボリュームを除外することはできません。ボリュームにアタッチされた、test ではない別のタグがある限り、ボリュームは結果に含まれて返されます。

test タグを持つすべてのボリュームを除外するには、以下の式から開始して、test タグを含むすべてのタグを配列に含めて返します。test タグでないタグには null 値が含まれています。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Tags[?Value == `test`]'
```

その後、test 関数を使用してすべての肯定的な not_null の結果をフィルタリングで抽出します。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)]'
```

結果をパイプ処理して、結果をフラット化すると、次のクエリになります。

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)] | []'
```

サーバー側とクライアント側のフィルタリングを組み合わせる

サーバー側とクライアント側のフィルタリングを一緒に使用できます。サーバー側のフィルタリングが最初に完了し、得られたデータがクライアントに送信され、`--query` パラメータがそのデータをフィルタリングします。大規模なデータセットを使用している場合、サーバー側のフィルタリングを最初に使用すると、クライアント側のフィルタリングが提供する強力なカスタマイズを維持しながら、AWS CLI 呼び出しごとにクライアントに送信されるデータの量を減らすことができます。

次の例では、サーバー側とクライアント側のフィルタリングの両方を使用する Amazon EC2 ポリ्यूームを一覧表示します。このサービスでは、`us-west-2a` アベイラビリティゾーンでアタッチされたすべてのポリ्यूームのリストがフィルタリングされます。`--query` パラメータでは、`Size` 値が 50 を超えるポリ्यूームにのみ出力を制限し、ユーザー定義の名前を持つ指定されたフィールドのみ表示されます。

```
$ aws ec2 describe-volumes \
  --filters "Name=availability-zone,Values=us-west-2a" "Name=status,Values=attached" \
  --query 'Volumes[?Size > `50`].{Id:VolumeId,Size:Size,Type:VolumeType}'
[
  {
    "Id": "vol-0be9bb0bf12345678",
    "Size": 80,
    "VolumeType": "gp2"
  }
]
```

次の例では、いくつかの基準を満たすイメージのリストを取得します。`--query` パラメータを使用して、`CreationDate` で出力を絞り込み、最新のイメージのみ選択します。最後に、1 つのイメージの `ImageId` が表示されます。

```
$ aws ec2 describe-images \
  --owners amazon \
  --filters "Name=name,Values=amzn*gp2" "Name=virtualization-type,Values=hvm" \
  "Name=root-device-type,Values=ebs" \
  --query "sort_by(Images, &CreationDate)[-1].ImageId" \
  --output text
ami-00ced3122871a4921
```

次の例では、を使用してリストに含まれるポリ्यूームの数 `length` をカウント IOPS することで、1000 を超える使用可能なポリ्यूームの数を表示します。

```
$ aws ec2 describe-volumes \  
  --filters "Name=status,Values=available" \  
  --query 'length(Volumes[?Iops > `1000`])'  
3
```

追加リソース

AWS CLI 自動プロンプト

フィルター式の使用を開始するときは、AWS CLI バージョン 2 の自動プロンプト機能を使用できます。F5 キーを押すと、自動プロンプト機能でプレビューが表示されます。詳細については、「[the section called “自動プロンプト”](#)」を参照してください。

JMESPath ターミナル

JMESPath Terminal は、クライアント側のフィルタリングに使用される JMESPath 式を試すためのインタラクティブなターミナルコマンドです。jpterm コマンドを使用すると、ターミナルに入力すると即時にクエリ結果が表示されます。AWS CLI 出力をターミナルに直接パイプ処理できるため、高度なクエリ実験が可能になります。

次の例では、JMESPath ターミナルに直接 `aws ec2 describe-volumes` 出力をパイプします。

```
$ aws ec2 describe-volumes | jpterm
```

JMESPath ターミナルとインストール手順の詳細については、「[の JMESPath ターミナル](#)」を参照してください GitHub。

jq ユーティリティ

jq ユーティリティは、クライアント側の出力を必要な出力形式に変換する方法を提供します。jq およびインストール手順の詳細については、「[」の「jq」](#)を参照してください GitHub。

のコマンドラインリターンコード AWS CLI

リターンコードは通常、コマンドのステータスを記述する AWS Command Line Interface (AWS CLI) コマンドの実行後に送信される非表示コードです。echo コマンドを使用して、最後の AWS CLI コマンドから送信されたコードを表示し、これらのコードを使用して、コマンドが成功したか失敗したか、コマンドにエラーがある理由を特定できます。リターンコードに加えて、`--debug` スイッチを使用してコマンドを実行すると、エラーに関する詳細情報を表示できます。このスイッチに

より、コマンドを処理するために AWS CLI が使用するステップおよび各ステップの結果についての詳細なレポートが作成されます。

AWS CLI コマンドのリターンコードを確認するには、コマンドの実行直後に次のいずれかの CLI コマンドを実行します。

Linux and macOS

```
$ echo $?  
0
```

Windows PowerShell

```
PS> echo $lastexitcode  
0
```

Windows Command Prompt

```
C:\> echo %errorlevel%  
0
```

以下は、AWS Command Line Interface (AWS CLI) コマンドの実行終了時に返されるリターンコード値です。

コード	意味
0	サービスはレスポンスHTTPステータスコード 200 で応答し、リクエストが送信された AWS CLI および AWS サービスによってエラーが発生しなかったことを示します。
1	1 つ以上の Amazon S3 転送操作が失敗しました。S3 コマンドに制限されます。
2	このリターンコードの意味は、コマンドによって異なります。 <ul style="list-style-type: none">すべての AWS CLI コマンドに適用可能 - 入力されたコマンドを解析できませんでした。解析エラーは、必要なサブコマンドまたは引数の欠如、不明なコマンドまたは引数の使用、あるいはその他の要因によって生じることがあります。S3 コマンドに限定 - 転送対象としてマークされた 1 つまたは複数のファイルが、転送プロセス中にスキップされました。ただし、転送対象としてマークされた他のすべての

コード	意味
	ファイルは正常に転送されました。転送プロセス中にスキップされるファイルには、存在しないファイル、文字の特別なデバイス、ブロックの特別なデバイス、FIFOキュー、またはソケットであるファイル、およびユーザーが読み取りアクセス許可を持たないファイルが含まれます。
130	コマンドは によって中断されましたSIGINT。これは、Ctrl + C キーでコマンドをキャンセルするために送信されるシグナルです。
252	コマンド構文が無効、不明なパラメータが指定されている、またはパラメータの値が正しくないため、コマンドを実行できませんでした。
253	システム環境または設定が無効です。指定されたコマンドは構文的に有効である可能性がありますが、設定または認証情報がないため、コマンドを実行できませんでした。
254	コマンドは正常に解析され、指定されたサービスに対してリクエストが行われましたが、サービスはエラーを返しました。これは通常、誤ったAPI使用やその他のサービス固有の問題を示します。
255	コマンドが失敗しました。AWS CLI またはリクエストが送信されたサービスによってAWS エラーが発生しました。

カスタムウィザードを使用して インタラクティブコマンドを実行する AWS CLI

AWS Command Line Interface (AWS CLI) では、一部のコマンドにウィザードを使用できます。使用可能な AWS CLI ウィザードの完全なリストを提供または表示するには、「」の[AWS CLI 「ウィザードフォルダ」](#)を参照してください GitHub。

仕組み

AWS コンソールと同様に、AWS CLI には UI ウィザードがあり、AWS リソースの管理をガイドします。ウィザードを使用するには、コマンド内のサービス名に続いて、wizard サブコマンドとウィザード名を呼び出します。コマンド構造は以下のとおりです。

構文:

```
$ aws <command> wizard <wizardName>
```

以下の例は、新しい dynamodb テーブルを作成するためのウィザードを呼び出しています。

```
$ aws dynamodb wizard new-table
```

aws configure は、ウィザード名を持たない唯一のウィザードです。このウィザードを実行するときは、以下の例にあるように、aws configure wizard コマンドを実行します。

```
$ aws configure wizard
```

ウィザードを呼び出すと、シェルにフォームが表示されます。各パラメータについて、選択するオプションのリストが提供されるか、文字列を入力するように促されます。リストから を選択するには、上矢印キーと下矢印キーを使用して、 を押しますENTER。オプションの詳細を表示するには、右矢印キーを押します。パラメータの入力が完了したら、 を押しますENTER。

```
$ aws configure wizard
What would you like to configure
> Static Credentials
  Assume Role
  Process Provider
  Additional CLI configuration
Enter the name of the profile:
Enter your Access Key Id:
Enter your Secret Access Key:
```

前のプロンプトを編集するには、SHIFT + を使用しますTAB。一部のウィザードでは、すべてのプロンプトを入力した後、AWS CloudFormation テンプレートまたは情報が入力された AWS CLI コマンドをプレビューできます。このプレビューモードは AWS CLI、サービスAPIs、スクリプトのテンプレートの作成について学ぶのに役立ちます。

プレビューENTER後または最後のプロンプトを押して、最終コマンドを実行します。

```
$ aws configure wizard
What would you like to configure
Enter the name of the profile: testWizard
Enter your Access Key Id: AB1C2D3EF4GH5I678J90K
Enter your Secret Access Key: ab1c2def34gh5i67j8k90l1mnop2qr3s45tu678v90
<ENTER>
```

でのエイリアスの作成と使用 AWS CLI

エイリアスは、頻繁に使用するコマンドやスクリプトを短縮するために AWS Command Line Interface (AWS CLI) で作成できるショートカットです。設定フォルダにある `alias` ファイルにエイリアスを作成します。

トピック

- [前提条件](#)
- [ステップ 1: エイリアスファイルを作成する](#)
- [ステップ 2: エイリアスを作成する](#)
- [ステップ 3: エイリアスを呼び出す](#)
- [エイリアスリポジトリの例](#)
- [リソース](#)

前提条件

エイリアスコマンドを使用するには、以下を完了する必要があります。

- AWS CLI をインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- 1.11.24 または 2.0.0 の最小 AWS CLI バージョンを使用します。
- (オプション) AWS CLI エイリアス bash スクリプトを使用するには、bash 互換ターミナルを使用する必要があります。

ステップ 1: エイリアスファイルを作成する

`alias` ファイルを作成するには、ファイルナビゲーションとテキストエディタを使用するか、プロシージャを使用して任意のターミナルを使用します step-by-step。エイリアスファイルをすばやく作成するには、次のコマンドブロックを使用します。

Linux and macOS

```
$ mkdir -p ~/.aws/cli
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> md %USERPROFILE%\.aws\cli
C:\> echo [toplevel] > %USERPROFILE%\.aws/cli/alias
```

エイリアスファイルを作成するには

1. AWS CLI 設定フォルダ `cli` という名前のフォルダを作成します。デフォルトでは、構成フォルダは Linux または macOS では「`~/.aws/`」、Windows 上は「`%USERPROFILE%\.aws\`」にあります。これは、ファイルナビゲーションまたは次のコマンドを使用して作成できます。

Linux and macOS

```
$ mkdir -p ~/.aws/cli
```

Windows

```
C:\> md %USERPROFILE%\.aws\cli
```

作成されるフォルダ「`cli`」のデフォルトパスは、Linux または macOS では「`~/.aws/cli/`」、Windows では「`%USERPROFILE%\.aws\cli`」上にあります。

2. `cli` フォルダで、拡張子なしの `alias` という名前のテキストファイルを作成し、最初の行に `[toplevel]` を追加します。任意のテキストエディタを使用してこのファイルを作成するか、次のコマンドを使用できます。

Linux and macOS

```
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> echo [toplevel] > %USERPROFILE%\.aws/cli/alias
```

ステップ 2: エイリアスを作成する

エイリアスは、基本的なコマンドまたは `bash` スクリプトを使用して作成できます。

基本的なコマンドエイリアスの作成

前の手順で作成した alias ファイルで、次の構文を使用してコマンドを追加することにより、エイリアスを作成できます。

[Syntax] (構文)

```
aliasname = command [--options]
```

- **aliasname** は、エイリアスと呼ばれるものです。- **command** は呼び出したいコマンドで、他のエイリアスを含めることができます。エイリアスにオプションやパラメータを含めたり、エイリアスを呼び出すときに追加したりできます。

次の例では、[aws sts get-caller-identity](#) コマンドを使用して、「aws whoami」という名前のエイリアスを作成します。このエイリアスは既存の AWS CLI コマンドを呼び出すことから、aws プレフィックスなしでコマンドを記述できます。

```
whoami = sts get-caller-identity
```

次の例は、前の whoami の例を用いて、Account フィルターオプションと output テキストオプションを追加します。

```
whoami2 = sts get-caller-identity --query Account --output text
```

サブコマンドエイリアスの作成

Note

サブコマンドエイリアス機能には、1.11.24 または 2.0.0 の最小 AWS CLI バージョンが必要です。

前の手順で作成した alias ファイルで、次の構文を使用してコマンドを追加することにより、サブコマンドのエイリアスを作成できます。

[Syntax] (構文)

```
[command commandGroup]  
aliasname = command [--options]
```

- **commandGroup** はコマンド名前空間です。例: コマンド `aws ec2 describe-regions` は `ec2` コマンドグループの下にあります。- **aliasname** は、エイリアスと呼ばれるものです。- **command** は呼び出したいコマンドで、他のエイリアスを含めることができます。エイリアスにオプションやパラメータを含めたり、エイリアスを呼び出すときに追加したりできます。

次の例では、[aws ec2 describe-regions](#) コマンドを使用して、「aws ec2 regions」という名前のエイリアスを作成します。このエイリアスは `ec2` コマンド名前空間の既存の AWS CLI コマンドを呼び出すため、`aws ec2` プレフィックスなしでコマンドを記述できます。

```
[command ec2]
regions = describe-regions --query Regions[].RegionName
```

コマンド名前空間外のコマンドからエイリアスを作成するには、コマンド全体の前に感嘆符を付けます。次の例では、[aws iam list-instance-profiles](#) コマンドを使用して、「aws ec2 instance-profiles」という名前のエイリアスを作成します。

```
[command ec2]
instance-profiles = !aws iam list-instance-profiles
```

Note

エイリアスは既存のコマンド名前空間のみを使用します。新しいコマンド名前空間を作成することはできません。例えば、`johnsmith` コマンド名前空間がまだ存在していない場合、`[command johnsmith]` セクションでエイリアスを作成することはできません。

bash スクリプトエイリアスの作成

Warning

AWS CLI エイリアス bash スクリプトを使用するには、bash 互換ターミナルを使用する必要があります。

次の構文を使用して、より高度なプロセス用の bash スクリプトを使用してエイリアスを作成できます。

[Syntax] (構文)

```
aliasname =  
    !f() {  
        script content  
    }; f
```

- **aliasname** は、エイリアスと **script content** は、エイリアスを呼び出すときに実行するスクリプトです。

次の例では、`opendns` を使用して現在の IP アドレスを出力します。他のエイリアスでもエイリアスを使用できるため、次の `myip` エイリアスは、他のエイリアス内から IP アドレスのアクセスを許可または取り消す場合に便利です。

```
myip =  
    !f() {  
        dig +short myip.opendns.com @resolver1.opendns.com  
    }; f
```

次のスクリプト例では、前の `aws myip` エイリアスを呼び出して、Amazon EC2 セキュリティグループ進入の IP アドレスを承認します。

```
authorize-my-ip =  
    !f() {  
        ip=$(aws myip)  
        aws ec2 authorize-security-group-ingress --group-id ${1} --cidr $ip/32 --protocol  
        tcp --port 22  
    }; f
```

bash スクリプトを使用するエイリアスを呼び出すと、変数は常に入力した順序で渡されます。bash スクリプトでは、変数名は考慮されず、表示される順序のみが考慮されます。次の `textalert` エイリアスの例では、`--message` オプションの変数が最初で、`--phone-number` オプションが 2 番目です。

```
textalert =  
    !f() {  
        aws sns publish --message "${1}" --phone-number ${2}  
    }; f
```

ステップ 3: エイリアスを呼び出す

alias ファイルに作成したエイリアスを実行するには、次の構文を使用します。エイリアスを呼び出すときに、追加のオプションを追加できます。

[Syntax] (構文)

```
$ aws aliasname
```

次の例では、aws whoami コマンドエイリアスを使用しています。

```
$ aws  
whoami  
{  
  "UserId": "A12BCD34E5FGHI6JKLM",  
  "Account": "1234567890987",  
  "Arn": "arn:aws:iam::1234567890987:user/userName"  
}
```

次の例では、aws whoami エイリアスを追加オプションとともに使用して、Account 出力で text の数のみを返します。

```
$ aws whoami --query Account --output  
text  
1234567890987
```

次の例では、aws ec2 regions [サブコマンドエイリアス](#)を使用しています。

```
$ aws ec2  
regions  
[  
  "ap-south-1",  
  "eu-north-1",  
  "eu-west-3",  
  "eu-west-2",  
  ...  
]
```

bash スクリプト変数を使用したエイリアスの呼び出し

bash スクリプトを使用するエイリアスを呼び出すと、変数は入力された順序で渡されます。bash スクリプトでは、変数の名前は考慮されず、表示される順序のみが考慮されます。例えば、次の

textalert エイリアスでは、オプション `--message` の変数が最初で、`--phone-number` が 2 番目です。

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

textalert エイリアス呼び出すときは、エイリアスで実行されるのと同じ順序で変数を渡す必要があります。次の例では、変数 `$message` と `$phone` を使用します。`$message` 変数は `${1}` オプションの `--message` として渡され、`$phone` 変数は `${2}` オプションの `--phone-number` として渡されます。これにより、textalert エイリアスを正常に呼び出してメッセージを送信できます。

```
$ aws textalert $message
$phone
{
  "MessageId": "1ab2cd3e4-fg56-7h89-i01j-2klmn34567"
}
```

次の例では、エイリアスを `$phone` および `$message` に呼び出すときに順序が切り替わります。`$phone` 変数は `${1}` オプションの `--message` として渡され、`$message` 変数は `${2}` オプションの `--phone-number` として渡されます。変数の順序が間違っているため、エイリアスは変数を間違えて渡します。`$message` の内容が `--phone-number` オプションの電話番号の書式要件と一致しないため、これはエラーを引き起こします。

```
$ aws textalert $phone
$message
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help

Unknown options: text
```

エイリアスリポジトリの例

の[AWS CLI エイリアスリポジトリ](#) GitHubには、デ AWS CLI ベロツパーチームとコミュニティによって作成された AWS CLI エイリアスの例が含まれています。alias ファイルサンプル全体を使用することも、個々のエイリアスを使うこともできます。

Warning

このセクションのコマンドを実行すると、既存の alias ファイルが削除されます。既存のエイリアスファイルが上書きされないようにするには、ダウンロード先の場所を変更します。

リポジトリのエイリアスを使用するには

1. Git をインストールします。インストール手順については、Git の Documentation の [Getting Started - Installing Git](#) をご参照ください。
2. jp コマンドをインストールします。jp コマンドは、tostring エイリアスで使用されます。インストール手順については、の [JMESPath \(jp\) README.md](#) を参照してくださいGitHub。
3. jq コマンドをインストールします。jq コマンドは、tostring-with-jq エイリアスで使用されます。インストール手順については、の [JSONプロセッサ \(jq\)](#) を参照してくださいGitHub。
4. 次のいずれかの操作を行って、alias ファイルをダウンロードします。
 - 次のコマンドを実行して、alias ファイルをリポジトリからダウンロードし、設定フォルダにコピーします。

Linux and macOS

```
$ git clone https://github.com/aws-labs/awscli-aliases.git
$ mkdir -p ~/.aws/cli
$ cp awscli-aliases/alias ~/.aws/cli/alias
```

Windows

```
C:\> git clone https://github.com/aws-labs/awscli-aliases.git
C:\> md %USERPROFILE%\aws\cli
C:\> copy awscli-aliases\alias %USERPROFILE%\aws\cli
```

- リポジトリから直接ダウンロードし、AWS CLI 設定cliフォルダのフォルダに保存します。デフォルトでは、構成フォルダは Linux または macOS では「~/aws/」、Windows 上は「%USERPROFILE%\aws\」にあります。
5. エイリアスが機能していることを確認するには、次のエイリアスを実行します。

```
$ aws whoami
```

aws sts get-caller-identity コマンドと同じ応答が表示されます。

```
{
  "Account": "012345678901",
  "UserId": "AIUAINBADX2VEG2TC6HD6",
  "Arn": "arn:aws:iam::012345678901:user/myuser"
}
```

リソース

- の[AWS CLI エイリアスリポジトリ](#) GitHubには、デ AWS CLI ベロッパーチームによって作成された AWS CLI エイリアスの例とコミュニティの貢献が含まれています AWS CLI 。
- re[AWS :Invent 2016: The Effective AWS CLI User](#) on からのエイリアス機能の発表YouTube。
- [aws sts get-caller-identity](#)
- [aws ec2 describe-instances](#)
- [aws sns publish](#)

のコード例 AWS CLI

この章では、(AWS CLI) AWS Command Line Interface を で使用する方法を示す例のコレクションを提供します AWS のサービス。

このガイド AWS CLI では、 に以下のタイプの例を示します。

- [ガイド付きコマンドの例](#) - 一部の AWS CLI で を使用する方法に関する AWS CLI ユーザーガイドのガイド付きコマンドの例 AWS のサービス。これらは、[AWS CLI バージョン 2 リファレンスガイド](#) の例よりも詳細な例であることがよくあります。
- [AWS CLI コマンド例](#) - [AWS CLI バージョン 2 リファレンスガイド](#) でも利用できるオープンソースのコマンド例。コマンド例は、 の[AWS CLI](#)リポジトリでホストされますGitHub。
- [AWS CLI Bash スクリプティングコード例の使用](#) - オープンソースの bash スクリプティング例。Bash スクリプトの例は、 の [AWS Code Examples Repository](#) でホストされていますGitHub。

フィードバックの例

必要なものが見つからなかった場合。このページの下部にあるフィードバックを提供するリンク、または[AWS CLI バージョン 2 リファレンスガイド](#) の関連するコマンドページで、コマンド例をリクエストします。

貢献する場合の [AWS Code Examples Repository](#) で AWS CLI コマンド例を指定しますGitHub。寄稿の詳細については、GitHub 「」 ページの[AWS CLI コード例の寄稿クイックステップ](#)を参照してください。

のガイド付きコマンドの例 AWS CLI

AWS Command Line Interface (AWS CLI) は、コマンドラインシェル内のコマンドを幅広く AWS のサービス 使用して操作できるオープンソースツールです。このセクションでは、 を活用しての一部 AWS CLI にアクセスする方法を示すガイド付き例を示します AWS のサービス。これには、高レベルの AWS CLI コマンドなど、いくつかのカスタムaws s3コマンドが含まれます。これらのコマンド例は、一部の で使用される一般的なアクションを示し AWS のサービス、詳細については追加のリソースを提供します。

経験豊富な AWS ユーザーでも、を初めて使用する場合でも AWS CLI、これらのガイド付き例は、AWS オペレーションを合理化するためのリソースとして機能します。

各で使用できるすべてのコマンドの完全なリファレンスについては AWS のサービス、[AWS CLI バージョン 2 リファレンスガイド](#) を参照してください。さらに、[組み込みのコマンドラインヘルプ](#)を使用して、の AWS のサービス、コマンド、オプション、機能の配列を調べることができます AWS CLI。

このセクションで使用できないその他のコマンド例については、[AWS CLI コマンドの例](#)「」セクションを参照してください。これらは、[AWS CLI バージョン 2 リファレンスガイド](#)でも利用可能なオープンソースコマンドの例です。コマンド例は、の[AWS CLI](#)リポジトリでホストされます GitHub。

オープンソースの bash スクリプティングの例については、「」を参照してください [the section called “Bash スクリプトの例”](#)。Bash スクリプトの例は、の [AWS Code Examples Repository](#) でホストされています GitHub。

サービス

- [での Amazon DynamoDB の使用 AWS CLI](#)
- [EC2 での Amazon の使用 AWS CLI](#)
- [での Amazon S3 Glacier の使用 AWS CLI](#)
- [IAM での の使用 AWS CLI](#)
- [での Amazon S3 の使用 AWS CLI](#)
- [SNS での Amazon へのアクセス AWS CLI](#)

での Amazon DynamoDB の使用 AWS CLI

Amazon DynamoDB のご紹介

[Amazon DynamoDB とは](#)

AWS Command Line Interface (AWS CLI) は、Amazon DynamoDB を含むすべての AWS データベースサービスをサポートします。テーブルの作成など、AWS CLI 即座のオペレーションにはを使用できます。また、ユーティリティスクリプト内に DynamoDB オペレーションを埋め込むときにも使用できます。

DynamoDB AWS CLI を使用する方法の詳細については、[dynamodb](#) AWS CLI コマンドリファレンスの「」を参照してください。DynamoDB

DynamoDB の AWS CLI コマンドを一覧表示するには、次のコマンドを使用します。

```
$ aws dynamodb help
```

トピック

- [前提条件](#)
- [DynamoDB テーブルの作成と使用](#)
- [DynamoDB Local を使用する](#)
- [リソース](#)

前提条件

dynamodb コマンドを実行するには、次のことが必要です。

- AWS CLIをインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。

DynamoDB テーブルの作成と使用

コマンドラインの形式は、DynamoDB コマンド名の後に、そのコマンドのパラメータが続きます。は、パラメータ値のCLI [省略構文](#)と完全な AWS CLI をサポートしますJSON。

次の例では、MusicCollectionという名前のテーブルを作成します。

```
$ aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-definitions AttributeName=Artist,AttributeType=S  
  AttributeName=SongTitle,AttributeType=S \  
  --key-schema AttributeName=Artist,KeyType=HASH  
  AttributeName=SongTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1
```

次の例に示されているようなコマンドと同様のコマンドで、新しい行をテーブルに追加できます。これらの例では、省略構文と の組み合わせを使用しますJSON。

```
$ aws dynamodb put-item \  

```

```
--table-name MusicCollection \  
--item '{  
    "Artist": {"S": "No One You Know"},  
    "SongTitle": {"S": "Call Me Today"} ,  
    "AlbumTitle": {"S": "Somewhat Famous"}  
}' \  
--return-consumed-capacity TOTAL  
{  
  "ConsumedCapacity": {  
    "CapacityUnits": 1.0,  
    "TableName": "MusicCollection"  
  }  
}
```

```
$ aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item '{  
    "Artist": {"S": "Acme Band"},  
    "SongTitle": {"S": "Happy Day"} ,  
    "AlbumTitle": {"S": "Songs About Life"}  
}' \  
  --return-consumed-capacity TOTAL  
{  
  "ConsumedCapacity": {  
    "CapacityUnits": 1.0,  
    "TableName": "MusicCollection"  
  }  
}
```

単一行コマンドJSONで有効な を構成することは難しい場合があります。これを容易にするために、AWS CLI はJSONファイルを読み取ることができます。例えば、 という名前のファイルに格納されている次のJSONスニペットを考えてみましょう `expression-attributes.json`。

```
{  
  ":v1": {"S": "No One You Know"},  
  ":v2": {"S": "Call Me Today"}  
}
```

そのファイルを使用して、`query` を使用する AWS CLI リクエストを発行することができます。次の例では、`expression-attributes.json` ファイルの内容が `--expression-attribute-values` パラメータの値に使用されます。

```
$ aws dynamodb query --table-name MusicCollection \  
  --key-condition-expression "Artist = :v1 AND SongTitle = :v2" \  
  --expression-attribute-values file://expression-attributes.json  
{  
  "Count": 1,  
  "Items": [  
    {  
      "AlbumTitle": {  
        "S": "Somewhat Famous"  
      },  
      "SongTitle": {  
        "S": "Call Me Today"  
      },  
      "Artist": {  
        "S": "No One You Know"  
      }  
    }  
  ],  
  "ScannedCount": 1,  
  "ConsumedCapacity": null  
}
```

DynamoDB Local を使用する

DynamoDB に加えて、DynamoDB Local AWS CLI で使用できます。DynamoDB Local は、小規模のクライアント側データベースとサーバーで、DynamoDB サービスに似せて作られています。DynamoDB Local を使用すると、DynamoDB ウェブサービスでテーブルやデータを操作せずに API、DynamoDB を使用するアプリケーションを記述できます。代わりに、すべての API アクションがローカルデータベースに再ルーティングされます。これにより、プロビジョニングされたスループット、データストレージ、およびデータ転送料金を節約できます。

DynamoDB Local の詳細とでの使用方法については AWS CLI、[Amazon DynamoDB デベロッパーガイド](#)の以下のセクションを参照してください。

- [DynamoDB Local](#)
- [DynamoDB Local での AWS CLI の使用](#)

リソース

AWS CLI リファレンス :

- [aws dynamodb](#)
- [aws dynamodb create-table](#)
- [aws dynamodb put-item](#)
- [aws dynamodb query](#)

サービスリファレンス:

- Amazon DynamoDB デベロッパーガイドの [DynamoDB Local](#)
- Amazon DynamoDB デベロッパーガイドの「[DynamoDB Local での AWS CLI の使用](#)」

EC2 での Amazon の使用 AWS CLI

Amazon Elastic Compute Cloud のご紹介

[Amazon の概要 EC2 - Elastic Cloud Server と でのホスティング AWS](#)

Amazon Elastic Compute Cloud (Amazon EC2) は、高度にスケーラブルで柔軟な仮想コンピューティング環境を提供します。Amazon EC2では、Amazon EC2インスタンスと呼ばれる仮想サーバーをプロビジョニングおよび管理して、さまざまなコンピューティングニーズに対応できます。

Amazon EC2インスタンスは、、メモリCPU、ストレージ、ネットワーク機能のさまざまな設定でカスタマイズできる仮想マシンです。アプリケーションの要件に応じて、軽量で費用対効果の高いオプションから強力で高性能なインスタンスまで、さまざまなインスタンスタイプを選択できます。この柔軟性により、コンピューティングのニーズに合わせてパフォーマンスとコスト効率を最適化できます。

さらに、Amazon EC2はコンピューティングリソースを効果的に管理できる一連の機能を提供しています。これには、新しいインスタンスをすばやく起動し、迅速にデプロイするためのカスタムマシンイメージ (AMIs) を作成し、必要に応じてコンピューティング容量を増減させる機能が含まれます。

AWS Command Line Interface () EC2を使用して Amazon の機能にアクセスできますAWS CLI。Amazon の AWS CLI コマンドを一覧表示するにはEC2、次のコマンドを使用します。

```
aws ec2 help
```

コマンドを実行する前に、デフォルトの認証情報を設定します。詳細については、「[の設定 AWS CLI](#)」を参照してください。

このトピックでは、Amazon の一般的なタスクを実行する AWS CLI コマンドの簡単な例を示します EC2。

AWS CLI コマンドの長い形式の例については、「」の[AWS CLI 「コード例リポジトリ」](#)を参照してくださいGitHub。

トピック

- [での Amazon EC2キーペアの作成、表示、削除 AWS CLI](#)
- [での Amazon EC2 セキュリティグループの作成、設定、削除 AWS CLI](#)
- [での Amazon EC2インスタンスの起動、一覧表示、および終了 AWS CLI](#)
- [の bash スクリプトを使用して Amazon EC2インスタンスタイプを変更する AWS CLI](#)

での Amazon EC2キーペアの作成、表示、削除 AWS CLI

AWS Command Line Interface (AWS CLI) を使用して、Amazon Elastic Compute Cloud (Amazon) のキーペアを作成、表示、削除できますEC2。キーペアを使用して Amazon EC2インスタンスに接続します。

インスタンスの作成EC2時にキーペアを Amazon に提供し、そのキーペアを使用してインスタンスへの接続時に認証する必要があります。

Note

その他のコマンド例については、[AWS CLI リファレンスガイド](#)を参照してください。

トピック

- [前提条件](#)
- [キーペアを作成する](#)
- [キーペアの表示](#)
- [キーペアの削除](#)
- [リファレンス](#)

前提条件

ec2 コマンドを実行するには、次のことが必要です。

- AWS CLIをインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- Amazon EC2 アクセスを許可するアクセスIAM許可を設定します。Amazon のIAMアクセス許可の詳細についてはEC2、「Amazon EC2 ユーザーガイド」の「Amazon [IAMのポリシーEC2](#)」を参照してください。

キーペアを作成する

キーペアを作成するには、[aws ec2 create-key-pair](#) コマンドを `--query` オプションおよび `--output text` オプションと共に使用して、プライベートキーをファイルに直接パイプします。

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text  
> MyKeyPair.pem
```

の場合 PowerShell、`> file`リダイレクトのデフォルトは UTF-8 エンコーディングで、一部のSSHクライアントでは使用できません。そのため、出力を `out-file` コマンドにパイプすることによって変換し、エンコードを `ascii` に明示的に設定する必要があります。

```
PS C:\>aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text  
| out-file -encoding ascii -filepath MyKeyPair.pem
```

出力された `MyKeyPair.pem` ファイルは以下のようになります。

```
-----BEGIN RSA PRIVATE KEY-----  
EXAMPLEKEYKCAQEAY7WZhaDsrA1W3mRlQtvhwYORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/  
vBoU7jLxx92pNHofnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7i05dSrvC7dQkW2duV5QuUdEQW  
Z/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449ilx9X1F  
G50TCFe0zf18dqCP6GzbPaIjiU19xX/az0R9V+tpU0zEL+wmXnZt3/nHPQ5xvD20JH67km6SuPW  
oPzev/D8V+x4+bHthfSjR9Y7DvQFjfbVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrqu  
/uler7vgIn5m7lN5LKw4hJLAIW6tUT/fzvtcHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/ufGxbl1  
mb5qwMGUnEpJaZD6QSSs3kICLwWUYUiGfc0uisbmJoap/GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2  
bahyWyJNfjLe4M86yd2YK3V2CmK+X/B0sShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/tJWSD9  
81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA30zdXzMQexXVJ1TLZVEH0E7bh1Y9d801ozR  
oQs/FiZNAx2iijCWyv0lpjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfq1+lIp1  
YkriL0DbLX1vRAH+yHPRit2hH0jtUNZ4Aaxv+cpq09qbUI3+43eEy24B7G/Uh+GTfbjsXs0xQx/x  
p9otyVwc7hsQ5TA5PZb+mvkJ50BEKzet9XcKw0NBVELGhnEPe7cCgYEA06Vgov6YHleHui9kHuws
```

```
ayav0e1c5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URipzWV0Eihvm+xTtmaZ1Sp//1kq75XDwnU
WA8gkn603QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
gYBjb0+0Zk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
oMo7yykABY70zd5wQewBQ4AdS1WSX4nGDtsiFxiI5sKuAAe0CbTosy1s8w8fxoJ5Tz1sdoxNeGs
Arq6Wv/G16zQuAE9zK9vwwKBgF+09VI/1wJBirsDGz9whVwfFPiTkJNvJZzYt69qezx1sjgFKshy
WBhd4xHZtmCqpBP1AymEjr/T01bxyARmXMnIOWIANNXMGB4KGSy11mzSVAoQ+fqR+cJ3d0dyP11j
jjb0Ed/NY8fr1NDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0i0egLda
NWUH38v/nDCgEpIXD5Hn3qAEcju1Ijmbw1vtW+nY2jVhv7UGd8MjwUTNGItdb6nsYqM2asrnF3qS
VRkAKKKYeGjkuFVTiW0YFjXkfcR/V+QFL50ndHAKJXjW7a4ejJLncTzmZSpYzwApc=
-----END RSA PRIVATE KEY-----
```

プライベートキーは に保存されず AWS、作成時にのみ取得できます。後で復元することはできません。代わりに、プライベートキーを紛失した場合は、新しいキーペアを作成する必要があります。

Linux コンピュータからインスタンスに接続している場合は、他のユーザーが読み取れないように、次のコマンドを使用してプライベートキーファイルのアクセス許可を設定することをお勧めします。

```
$ chmod 400 MyKeyPair.pem
```

キーペアの表示

「フィンガープリント」はキーペアから生成され、これを使用してローカルマシンのプライベートキーが AWS に保存されたパブリックキーと一致することを確認できます。

フィンガープリントは、プライベートキーの DER エンコードされたコピーから取得された SHA1 ハッシュです。この値は、キーペアの作成時にキャプチャされ、パブリックキー AWS とともに に保存されます。フィンガープリントは、Amazon EC2 コンソールで表示することも、AWS CLI コマンドを実行して表示することもできます [aws ec2 describe-key-pairs](#)。

次の例は、MyKeyPair のフィンガープリントを表示します。

```
$ aws ec2 describe-key-pairs --key-name MyKeyPair
{
  "KeyPairs": [
    {
      "KeyName": "MyKeyPair",
      "KeyFingerprint":
"1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f"
    }
  ]
}
```

キーとフィンガープリントの詳細については、[「Amazon ユーザーガイド」の「Amazon EC2 キーペア」](#)を参照してください。 EC2

キーペアの削除

キーペアを削除するには、[aws ec2 delete-key-pair](#) コマンドを実行し、*MyKeyPair* 削除するペアの名前。

```
$ aws ec2 delete-key-pair --key-name MyKeyPair
```

リファレンス

AWS CLI リファレンス :

- [aws ec2](#)
- [aws ec2 create-key-pair](#)
- [aws ec2 delete-key-pair](#)
- [aws ec2 describe-key-pairs](#)

その他のリファレンス:

- [Amazon Elastic Compute Cloud のドキュメント](#)
- および AWS CLI コード例を表示して貢献するには AWS SDK、の[AWS コード例リポジトリ](#)を参照してくださいGitHub。

での Amazon EC2 セキュリティグループの作成、設定、削除 AWS CLI

基本的にファイアウォールとして動作する Amazon Elastic Compute Cloud (Amazon EC2) インスタンスのセキュリティグループを作成し、どのネットワークトラフィックが出入りできるかを決定するルールを設定できます。

AWS Command Line Interface (AWS CLI) を使用して、セキュリティグループを作成し、既存のセキュリティグループにルールを追加し、セキュリティグループを削除します。

Note

その他のコマンド例については、リファレンス[AWS CLI ガイド](#)を参照してください。

トピック

- [前提条件](#)
- [セキュリティグループの作成](#)
- [ルールをセキュリティグループに追加する](#)
- [セキュリティグループを削除する](#)
- [リファレンス](#)

前提条件

ec2 コマンドを実行するには、次のことが必要です。

- AWS CLIをインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- Amazon EC2 アクセスを許可するアクセスIAM許可を設定します。Amazon のIAMアクセス許可の詳細についてはEC2、「Amazon EC2 ユーザーガイド」の「Amazon [IAMのポリシー-EC2](#)」を参照してください。

セキュリティグループの作成

仮想プライベートクラウド (VPCs) に関連付けられたセキュリティグループを作成できます。

次の[aws ec2 create-security-group](#)例は、指定された のセキュリティグループを作成する方法を示していますVPC。

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group" --  
vpc-id vpc-1a2b3c4d  
{  
  "GroupId": "sg-903004f8"  
}
```

セキュリティグループの初期情報を表示するには、[aws ec2 describe-security-groups](#) コマンドを実行します。EC2-VPC セキュリティグループは、名前ではなくvpc-id、 によってのみ参照できます。

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8  
{  
  "SecurityGroups": [  
    {
```

```
    "IpPermissionsEgress": [
      {
        "IpProtocol": "-1",
        "IpRanges": [
          {
            "CidrIp": "0.0.0.0/0"
          }
        ],
        "UserIdGroupPairs": []
      }
    ],
    "Description": "My security group"
  },
  "IpPermissions": [],
  "GroupName": "my-sg",
  "VpcId": "vpc-1a2b3c4d",
  "OwnerId": "123456789012",
  "GroupId": "sg-903004f8"
}
]
}
```

ルールをセキュリティグループに追加する

Amazon EC2インスタンスを実行するときは、セキュリティグループのルールを有効にして、イメージに接続する手段の受信ネットワークトラフィックを許可する必要があります。

例えば、Windows インスタンスを起動する場合、通常、リモートデスクトッププロトコル (RDP) をサポートするようにTCPポート 3389 のインバウンドトラフィックを許可するルールを追加します。Linux インスタンスを起動する場合は、通常、ポート 22 TCP のインバウンドトラフィックがSSH接続をサポートできるようにするルールを追加します。

セキュリティグループにルールを追加するには、[aws ec2 authorize-security-group-ingress](#) コマンドを使用します。このコマンドに必要なパラメータは、コンピュータのパブリック IP アドレス、またはコンピュータがアタッチされているネットワーク (アドレス範囲の形式) です [CIDR](#)。

Note

パブリック IP アドレスを判別できるように、次のサービス <https://checkip.amazonaws.com/> を提供しています。IP アドレスの識別に役立つその他のサービスを見つけるには、ブラウザを使用して「what is my IP address」を検索します。動的 IP アドレス (プライベートネットワークからのNATゲートウェイ経由) を使用して ISP またはファイアウォールの背後から接続

する場合、アドレスは定期的に変更される可能性があります。その場合、クライアントコンピュータによって使用される IP アドレスの範囲を見つける必要があります。

次の例は、IP アドレス `sg-903004f8` を使用して ID を持つ EC2-VPC セキュリティグループに RDP (TCPポート 3389) のルールを追加する方法を示しています。

まず、あなたの IP アドレスを探してください。

```
$ curl https://checkip.amazonaws.com  
x.x.x.x
```

その後、[aws ec2 authorize-security-group-ingress](#) コマンドを実行して、セキュリティグループに IP アドレスを追加できます。

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 3389 --cidr x.x.x.x/x
```

次のコマンドは、同じセキュリティグループのインスタンス SSH に を有効にする別のルールを追加します。

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 22 --cidr x.x.x.x/x
```

セキュリティグループに加えられた変更を表示するには、[aws ec2 describe-security-groups](#) コマンドを実行します。

```
$ aws ec2 describe-security-groups --group-ids sg-903004f8  
{  
  "SecurityGroups": [  
    {  
      "IpPermissionsEgress": [  
        {  
          "IpProtocol": "-1",  
          "IpRanges": [  
            {  
              "CidrIp": "0.0.0.0/0"  
            }  
          ],  
          "UserIdGroupPairs": []  
        }  
      ]  
    }  
  ]  
}
```



```
    ],
    "Description": "My security group"
    "IpPermissions": [
        {
            "ToPort": 22,
            "IpProtocol": "tcp",
            "IpRanges": [
                {
                    "CidrIp": "x.x.x.x/x"
                }
            ]
            "UserIdGroupPairs": [],
            "FromPort": 22
        }
    ],
    "GroupName": "my-sg",
    "OwnerId": "123456789012",
    "GroupId": "sg-903004f8"
}
]
```

セキュリティグループを削除する

セキュリティグループを削除するには、[aws ec2 delete-security-group](#) コマンドを実行します。

Note

環境に現在アタッチされているセキュリティグループは削除できません。

次のコマンド例では、EC2-VPC セキュリティグループを削除します。

```
$ aws ec2 delete-security-group --group-id sg-903004f8
```

リファレンス

AWS CLI リファレンス :

- [aws ec2](#)
- [aws ec2 authorize-security-group-ingress](#)

- [aws ec2 create-security-group](#)
- [aws ec2 delete-security-group](#)
- [aws ec2 describe-security-groups](#)

その他のリファレンス:

- [Amazon Elastic Compute Cloud のドキュメント](#)
- および AWS CLI コード例を表示して貢献するには AWS SDK、の[AWS コード例リポジトリ](#)を参照してくださいGitHub。

での Amazon EC2 インスタンスの起動、一覧表示、および終了 AWS CLI

AWS Command Line Interface (AWS CLI) を使用して、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを起動、一覧表示、終了できます。AWS 無料利用枠内にはないインスタンスを起動すると、インスタンスを起動した後に課金され、インスタンスがアイドル状態のままであっても、インスタンスが実行されている時間に対して課金されます。

Note

その他のコマンド例については、リファレンス[AWS CLI ガイド](#)を参照してください。

トピック

- [前提条件](#)
- [インスタンスの起動](#)
- [インスタンスへのブロックデバイスの追加](#)
- [インスタンスへのタグの追加](#)
- [インスタンスへの接続](#)
- [インスタンスの一覧表示](#)
- [インスタンスの終了](#)
- [リファレンス](#)

前提条件

このトピックの ec2 コマンドを実行するには、次の操作を行う必要があります。

- AWS CLIをインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- Amazon EC2 アクセスを許可するアクセスIAM許可を設定します。Amazon のIAMアクセス許可の詳細についてはEC2、「Amazon EC2 ユーザーガイド」の「Amazon [IAMのポリシーEC2](#)」を参照してください。
- [キーペア](#)と[セキュリティグループ](#)を作成します。
- Amazon マシンイメージ (AMI) を選択し、AMIID を書き留めます。詳細については、「Amazon EC2ユーザーガイド」の「[適切なを見つけるAMI](#)」を参照してください。

インスタンスの起動

AMI 選択した を使用して Amazon EC2インスタンスを起動するには、[aws ec2 run-instances](#) コマンドを使用します。インスタンスを仮想プライベートクラウド () で起動できますVPC。

当初、インスタンスは pending 状態で表示されますが、数分後に running 状態に変わります。

次の例は、 の指定されたサブネットでt2.microインスタンスを起動する方法を示しています VPC。を置き換える *italicized* 独自のパラメータ値。

```
$ aws ec2 run-instances --image-id ami-xxxxxxx --count 1 --instance-type t2.micro --
key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
{
  "OwnerId": "123456789012",
  "ReservationId": "r-5875ca20",
  "Groups": [
    {
      "GroupName": "my-sg",
      "GroupId": "sg-903004f8"
    }
  ],
  "Instances": [
    {
      "Monitoring": {
        "State": "disabled"
      },
      "PublicDnsName": null,
      "Platform": "windows",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
    }
  ],
}
```

```
"EbsOptimized": false,
"LaunchTime": "2013-07-19T02:42:39.000Z",
"PrivateIpAddress": "10.0.1.114",
"ProductCodes": [],
"VpcId": "vpc-1a2b3c4d",
"InstanceId": "i-5203422c",
"ImageId": "ami-173d747e",
"PrivateDnsName": "ip-10-0-1-114.ec2.internal",
"KeyName": "MyKeyPair",
"SecurityGroups": [
  {
    "GroupName": "my-sg",
    "GroupId": "sg-903004f8"
  }
],
"ClientToken": null,
"SubnetId": "subnet-6e7f829e",
"InstanceType": "t2.micro",
"NetworkInterfaces": [
  {
    "Status": "in-use",
    "SourceDestCheck": true,
    "VpcId": "vpc-1a2b3c4d",
    "Description": "Primary network interface",
    "NetworkInterfaceId": "eni-a7edb1c9",
    "PrivateIpAddresses": [
      {
        "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
        "Primary": true,
        "PrivateIpAddress": "10.0.1.114"
      }
    ]
  },
  {
    "PrivateDnsName": "ip-10-0-1-114.ec2.internal",
    "Attachment": {
      "Status": "attached",
      "DeviceIndex": 0,
      "DeleteOnTermination": true,
      "AttachmentId": "eni-attach-52193138",
      "AttachTime": "2013-07-19T02:42:39.000Z"
    }
  },
  {
    "Groups": [
      {
        "GroupName": "my-sg",
        "GroupId": "sg-903004f8"
      }
    ]
  }
]
```

```
        }
      ],
      "SubnetId": "subnet-6e7f829e",
      "OwnerId": "123456789012",
      "PrivateIpAddress": "10.0.1.114"
    }
  ],
  "SourceDestCheck": true,
  "Placement": {
    "Tenancy": "default",
    "GroupName": null,
    "AvailabilityZone": "us-west-2b"
  },
  "Hypervisor": "xen",
  "BlockDeviceMappings": [
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": true,
        "VolumeId": "vol-877166c8",
        "AttachTime": "2013-07-19T02:42:39.000Z"
      }
    }
  ],
  "Architecture": "x86_64",
  "StateReason": {
    "Message": "pending",
    "Code": "pending"
  },
  "RootDeviceName": "/dev/sda1",
  "VirtualizationType": "hvm",
  "RootDeviceType": "ebs",
  "Tags": [
    {
      "Value": "MyInstance",
      "Key": "Name"
    }
  ],
  "AmiLaunchIndex": 0
}
]
```

インスタンスへのブロックデバイスの追加

起動する各インスタンスにはルートデバイスボリュームが関連付けられています。ブロックデバイスマッピングを使用して、起動時にインスタンスにアタッチする追加の Amazon Elastic Block Store (Amazon EBS) ボリュームまたはインスタンスストアボリュームを指定できます。

ブロックデバイスをインスタンスに追加するには、`run-instances` を使用するとき `--block-device-mappings` オプションを指定します。

次のパラメータ例では、サイズが 20 GB の標準 Amazon EBS ボリュームをプロビジョニングし、識別子を使用してインスタンスにマッピングします `/dev/sdf`。

```
--block-device-mappings "[{\\"DeviceName\\":\"/dev/sdf\\\",\\"Ebs\\":{\\"VolumeSize\\":20,\\\"DeleteOnTermination\\":false}}]"
```

次の例では、既存のスナップショットに基づいて `/dev/sdf`、にマッピングされた Amazon EBS ボリュームを追加します。スナップショットは、ボリュームに自動的にロードされるイメージを表します。スナップショットを指定するとき、ボリュームサイズを指定する必要はありません。イメージを保持できる十分な大きさになります。ただし、サイズを指定する場合は、スナップショットのサイズ以上である必要があります。

```
--block-device-mappings "[{\\"DeviceName\\":\"/dev/sdf\\\",\\"Ebs\\":{\\"SnapshotId\\":\\"snap-a1b2c3d4\\"}}]"
```

次の例は、2 つのボリュームをインスタンスに追加します。インスタンスで使用できるボリュームの数は、インスタンスタイプによって異なります。

```
--block-device-mappings "[{\\"DeviceName\\":\"/dev/sdf\\\",\\"VirtualName\\":\\"ephemeral0\\"},{\\"DeviceName\\":\"/dev/sdg\\\",\\"VirtualName\\":\\"ephemeral1\\"}]"
```

次の例では、マッピング (`/dev/sdj`) を作成しますが、インスタンスのボリュームはプロビジョニングされません。

```
--block-device-mappings "[{\\"DeviceName\\":\"/dev/sdj\\\",\\"NoDevice\\":\\"\"}]"
```

詳細については、「Amazon ユーザーガイド」の [「デバイスマッピングのブロック」](#) を参照してください。 EC2

インスタンスへのタグの追加

タグは、AWS リソースに割り当てるラベルです。リソースにメタデータを追加して、さまざまな目的に使用できます。詳細については、「Amazon EC2ユーザーガイド」の[「リソースのタグ付け」](#)を参照してください。

次の例は、[aws ec2 create-tags](#) コマンドを使用して、キー名「Name」と値「MyInstance」を持つタグを指定されたインスタンスに追加する方法を示しています。

```
$ aws ec2 create-tags --resources i-5203422c --tags Key=Name,Value=MyInstance
```

インスタンスへの接続

実行中のインスタンスに接続して、目の前にあるコンピュータと同じように使用することができます。詳細については、「[Amazon ユーザーガイド](#)」の「[Amazon EC2 インスタンスに接続する](#)」を参照してください。 EC2

インスタンスの一覧表示

を使用して AWS CLI、インスタンスを一覧表示し、インスタンスに関する情報を表示できます。すべてのインスタンスを一覧表示することも、目的のインスタンスに基づいて結果をフィルタリングすることもできます。

次に、[aws ec2 describe-instances](#) コマンドの使用例を示します。

次のコマンドでは、すべてのインスタンスが一覧表示されます。

```
$ aws ec2 describe-instances
```

次のコマンドは、リストを t2.micro インスタンスのみにフィルタリングして、一致した InstanceId 値のみを出力します。

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query  
"Reservations[].Instances[].InstanceId"  
[  
  "i-05e998023d9c69f9a"  
]
```

次のコマンドは、タグ Name=MyInstance を持つインスタンスをリストします。

```
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=MyInstance"
```

次のコマンドは、AMIs、`ami-x0123456`、`ami-y0123456`のいずれかを使用して起動されたインスタンスを一覧表示します `ami-z0123456`。

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-y0123456,ami-z0123456"
```

インスタンスの終了

インスタンスを終了すると、インスタンスが削除されます。インスタンスの終了後に、インスタンスに再接続することはできません。

インスタンスの状態が `shutting-down` または `terminated` に変わったら、そのインスタンスへの課金は停止します。後でインスタンスに再接続する必要がある場合は、`terminate-instances` の代わりに [stop-instances](#) を使用します。詳細については、「Amazon EC2ユーザーガイド」の「[インスタンスの終了](#)」を参照してください。

インスタンスを削除するには、コマンド [aws ec2 terminate-instances](#) を使用して削除します。

```
$ aws ec2 terminate-instances --instance-ids i-5203422c
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-5203422c",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

リファレンス

AWS CLI リファレンス :

- [aws ec2](#)

- [aws ec2 create-tags](#)
- [aws ec2 describe-instances](#)
- [aws ec2 run-instances](#)
- [aws ec2 terminate-instances](#)

その他のリファレンス:

- [Amazon Elastic Compute Cloud のドキュメント](#)
- および AWS CLI コード例を表示して貢献するには AWS SDK、の [AWS コード例リポジトリ](#) を参照してくださいGitHub。

の bash スクリプトを使用して Amazon EC2 インスタンスタイプを変更する AWS CLI

Amazon のこの bash スクリプト例では、AWS Command Line Interface () を使用して Amazon インスタンスの EC2 インスタンスタイプ EC2 を変更します AWS CLI。実行中の場合はインスタンスを停止し、インスタンスタイプを変更してから、リクエストに応じてインスタンスを再起動します。シェルスクリプトは、コマンドラインインターフェイスで実行するように設計されたプログラムです。

Note

その他のコマンド例については、[AWS CLI リファレンスガイド](#) を参照してください。

トピック

- [開始する前に](#)
- [この例について](#)
- [パラメータ](#)
- [ファイル](#)
- [リファレンス](#)

開始する前に

以下の例を実行する前に、次のことを完了する必要があります。

- AWS CLI をインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。

- 使用するプロファイルには、例によって実行される AWS オペレーションを許可するアクセス許可が必要です。
- 停止および変更するアクセス許可があるアカウントで実行中の Amazon EC2 インスタンス。テストスクリプトを実行すると、インスタンスが起動し、タイプの変更がテストされてから、インスタンスが終了します。
- AWS ベストプラクティスとして、このコードの最小権限、またはタスクの実行に必要なアクセス許可のみを付与します。詳細については、[「アイデンティティとアクセス管理 \(\) ユーザーガイド」の「最小権限の付与」](#)を参照してください。AWS IAM
- このコードは、すべての AWS リージョンでテストされているわけではありません。一部の AWS サービスは、特定のリージョンでのみ利用できます。詳細については、AWS 全般のリファレンスガイドの[「サービスエンドポイントとクォータ」](#)を参照してください。
- このコードを実行すると、AWS アカウントに料金が発生する可能性があります。作業が終了したら、このスクリプトによって作成されたすべてのリソースを確実に削除するのはお客様の責任になります。

この例について

この例は、シェルスクリプトファイル `change_ec2_instance_type.sh` 内の関数として記述されており、別のスクリプトまたはコマンドラインから `source` できます。各スクリプトファイルには、各関数について説明するコメントが含まれています。関数がメモリに置かれたら、コマンドラインから呼び出すことができます。例えば、次のコマンドは、指定されたインスタンスのタイプを `t2.nano` に変更します。

```
$ source ./change_ec2_instance_type.sh
$ ./change_ec2_instance_type -i *instance-id* -t new-type
```

完全な例とダウンロード可能なスクリプトファイルについては、の AWS コード例リポジトリの [「Amazon EC2 インスタンスタイプの変更」](#) を参照してください GitHub。

パラメータ

`-i` - (文字列) 変更するインスタンス ID を指定します。

`-t` - (文字列) 切り替える Amazon EC2 インスタンスタイプを指定します。

`-r` - (スイッチ) デフォルトでは、これは設定されていません。`-r` が設定されている場合、タイプの切り替え後にインスタンスを再起動します。

-f - (スイッチ) デフォルトでは、スクリプトは切り替えを行う前にインスタンスのシャットダウンを確認するプロンプトを表示します。-f が設定されている場合、この関数はインスタンスをシャットダウンしてタイプを切り替える前にユーザーにプロンプトを表示しません。

-v - (スイッチ) デフォルトでは、スクリプトはサイレントに動作し、エラーが発生した場合のみ出力を表示します。-v が設定されている場合、関数はそのオペレーション全体にわたってステータスを表示します。

ファイル

change_ec2_instance_type.sh

メインスクリプトファイルには、次のタスクを実行する `change_ec2_instance_type()` 関数が含まれています。

- 指定された Amazon EC2 インスタンスが存在することを確認します。
- -f が選択されていない限り、インスタンスを停止する前にユーザーに警告を表示します。
- インスタンスタイプを変更する。
- -r を設定した場合、インスタンスを再起動し、インスタンスが実行中であることを確認します。

で [change_ec2_instance_type.sh](#) のコードを表示します GitHub。

test_change_ec2_instance_type.sh

ファイル `test_change_ec2_instance_type.sh` のスクリプトは、`change_ec2_instance_type` 関数のさまざまなコードパスをテストします。テストスクリプトのすべてのステップが正しく動作する場合、テストスクリプトは作成したすべてのリソースを削除します。

テストスクリプトは、次のパラメータを使用して実行できます。

- -v - (スイッチ) 各テストには pass/failure status as they run. By default, the tests runs silently and the output includes only the final overall pass/failure ステータスが表示されます。
- -i - (スイッチ) 各テストの後にスクリプトが一時停止し、各ステップの中間結果を参照できるようにします。Amazon EC2 コンソールを使用してインスタンスの現在のステータスを確認できます。プロンプト ENTER で を押すと、スクリプトは次のステップに進みます。

で [test_change_ec2_instance_type.sh](#) のコードを表示します GitHub。

awsdocs_general.sh

スクリプトファイル `awsdocs_general.sh` には、AWS CLI の高度な例全体で使用される汎用関数が格納されています。

で [awsdocs_general.sh](#) のコードを表示しますGitHub。

リファレンス

AWS CLI リファレンス :

- [aws ec2](#)
- [aws ec2 describe-instances](#)
- [aws ec2 modify-instance-attribute](#)
- [aws ec2 start-instances](#)
- [aws ec2 stop-instances](#)
- [aws ec2 wait instance-running](#)
- [aws ec2 wait instance-stopped](#)

その他のリファレンス:

- [Amazon Elastic Compute Cloud のドキュメント](#)
- および AWS CLI コード例を表示して貢献するには AWS SDK、の [AWS コード例リポジトリ](#) を参照してくださいGitHub。

での Amazon S3 Glacier の使用 AWS CLI

Amazon S3 Glacier のご紹介

[Amazon S3 Glacier のご紹介](#)

このトピックでは、S3 Glacier の一般的なタスクを実行する AWS CLI コマンドの例を示します。この例では、AWS CLI を使用して大きなファイルを S3 Glacier にアップロードする方法を、より小さな部分に分割し、コマンドラインからアップロードする方法を示しています。

AWS Command Line Interface () を使用して Amazon S3 Glacier の機能にアクセスできますAWS CLI。S3 Glacier の AWS CLI コマンドを一覧表示するには、次のコマンドを使用します。

```
aws glacier help
```

Note

コマンドリファレンスおよびその他の例については、AWS CLI リファレンスガイドの [aws glacier](#) を参照してください。

トピック

- [前提条件](#)
- [Amazon S3 Glacier ボールトを作成します](#)
- [ファイルのアップロードの準備](#)
- [マルチパートアップロードの開始とファイルのアップロード](#)
- [アップロードの完了](#)
- [リソース](#)

前提条件

glacier コマンドを実行するには、次のことが必要です。

- AWS CLIをインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- このチュートリアルでは、Linux や macOS など、Unix 互換オペレーティングシステムに通常プリインストールされている、いくつかのコマンドラインツールを使用します。Windows ユーザーは、[Cygwin](#) をインストールし、Cygwin ターミナルからコマンドを実行することで、同じツールを使用できます。同じ機能を実行する Windows のネイティブコマンドとユーティリティはそのように注記されています。

Amazon S3 Glacier ボールトを作成します

[create-vault](#) コマンドを使用してボールトを作成します。

```
$ aws glacier create-vault --account-id - --vault-name myvault
```

```
{  
  "location": "/123456789012/vaults/myvault"  
}
```

Note

すべての S3 Glacier コマンドにはアカウント ID パラメータが必要です。現在のアカウントを使用するには、ハイフン文字を使用します (--account-id -)。

ファイルのアップロードの準備

テストアップロード用のファイルを作成します。次のコマンドは、`myfile` という名前のファイルを作成します。`largefile` には、3 MiB のランダムデータのみが含まれます。

Linux または macOS

```
$ dd if=/dev/urandom of=largefile bs=3145728 count=1  
1+0 records in  
1+0 records out  
3145728 bytes (3.1 MB) copied, 0.205813 s, 15.3 MB/s
```

dd は、多数のバイトを入力ファイルから出力ファイルにコピーするユーティリティです。前の例では、ランダムデータのソースとしてシステムデバイスファイル `/dev/urandom` を使用します。fsutil は Windows で同様の関数を実行します。

Windows

```
C:\> fsutil file createnew largefile 3145728  
File C:\temp\largefile is created
```

次に、ファイルスプリッターを使用してファイルを 1 MiB (1,048,576 バイト) チャンクに分割します。

```
$ split -b 1048576 --verbose largefile chunk  
creating file `chunkaa'  
creating file `chunkab'  
creating file `chunkac'
```

マルチパートアップロードの開始とファイルのアップロード

[initiate-multipart-upload](#) コマンドを使用して、Amazon S3 Glacier でマルチパートアップロードを作成します。

```
$ aws glacier initiate-multipart-upload --account-id - --archive-description "multipart upload test" --part-size 1048576 --vault-name myvault
{
  "uploadId": "19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ",
  "location": "/123456789012/vaults/myvault/multipart-
uploads/19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
}
```

S3 Glacier では、マルチパートアップロードを設定するために各パートのサイズ (バイト) (この例では 1 MiB)、ボールド名、アカウント ID が必要です。は、オペレーションが完了するとアップロード ID を AWS CLI 出力します。後で使用できるように、アップロード ID をシェル変数に保存します。

Linux または macOS

```
$ UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
```

Windows

```
C:\> set UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
```

次に、[upload-multipart-part](#) コマンドを使用して 3 つのパートをそれぞれアップロードします。

```
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkaa --range 'bytes
0-1048575/*' --account-id - --vault-name myvault
{
  "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkab --range 'bytes
1048576-2097151/*' --account-id - --vault-name myvault
{
```

```
"checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
$ aws glacier upload-multipart-part --upload-id $UPLOADID --body chunkac --range 'bytes
2097152-3145727/*' --account-id - --vault-name myvault
{
  "checksum": "e1f2a7cd6e047fa606fe2f0280350f69b9f8cfa602097a9a026360a7edc1f553"
}
```

Note

前の例では、ドル記号 (\$) を使用して Linux の UPLOADID シェル変数の内容を参照しています。Windows コマンドラインでは、変数名の両側にパーセント記号 (%) を使用します (例:%UPLOADID%)。

S3 Glacier によって正しい順序で再構成されるように、アップロード時には各パートのバイト範囲を指定する必要があります。各部分は 1,048,576 バイトであるため、1 番目の部分は 0-1048575、2 番目は 1048576-2097151、3 番目は 2097152-3145727 に配置されます。

アップロードの完了

Amazon S3 Glacier では、アップロードされたすべてのピースが AWS インタクトに達したことを確認するために、元のファイルのツリーハッシュが必要です。

ツリーハッシュを計算するには、ファイルを 1 つの MiB 部分に分割し、各ピースのバイナリ SHA-256 ハッシュを計算する必要があります。次に、ハッシュのリストをペアに分割し、各ペアの 2 つのバイナリハッシュを結合して、結果のハッシュを取得します。ハッシュが 1 つだけになるまでこのプロセスを繰り返します。レベルのいずれかに奇数のハッシュがある場合は、変更せずに次のレベルに昇格させます。

コマンドラインユーティリティを使用して木構造ハッシュを正しく計算するために重要なことは、各ハッシュをバイナリ形式で保存し、最後のステップでのみ 16 進数に変換することです。木構造で 16 進数バージョンのハッシュを結合またはハッシュすると、正しい結果を得ることができません。

Note

Windows ユーザーは、type の代わりに cat コマンドを使用できます。OpenSSL は、Windows at [Open SSL.org](https://openSSL.org) で使用できます。

木構造ハッシュを計算するには

1. 元のファイルを分割していない場合は、1 MiB のパーツに分割します。

```
$ split --bytes=1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

2. 各チャンクのバイナリ SHA-256 ハッシュを計算して保存します。

```
$ openssl dgst -sha256 -binary chunkaa > hash1
$ openssl dgst -sha256 -binary chunkab > hash2
$ openssl dgst -sha256 -binary chunkac > hash3
```

3. 最初の 2 つのハッシュ結合を実行し、結果のバイナリハッシュを取得します。

```
$ cat hash1 hash2 > hash12
$ openssl dgst -sha256 -binary hash12 > hash12hash
```

4. チャンク aa および ab の親ハッシュをチャンク ac のハッシュと結合して結果をハッシュします。今回は 16 進数で出力します。シェル変数に結果を保存します。

```
$ cat hash12hash hash3 > hash123
$ openssl dgst -sha256 hash123
SHA256(hash123)= 9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
$ TREEHASH=9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
```

最後に、[complete-multipart-upload](#) コマンドを使用してアップロードを完了します。このコマンドは、元のファイルのサイズ (バイト単位)、最終的な 16 進数の木構造ハッシュ値、およびアカウント ID とボールド名を使用します。

```
$ aws glacier complete-multipart-upload --checksum $TREEHASH --archive-size 3145728 --
upload-id $UPLOADID --account-id - --vault-name myvault
{
  "archiveId": "d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrAllGAA0NJAzo5QdP-
N83MKqd96Unspoa5H51ItWX-sK8-QS0ZhwsyGiu9-R-
kWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg",
  "checksum": "9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
```

```
"location": "/123456789012/vaults/myvault/archives/
d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrALLGAA0NJAzo5QdP-N83MKqd96Unspoa5H51ItWX-sK8-
QS0ZhwsyGiu9-R-kwWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg"
}
```

[describe-vault](#) コマンドを使用して、ポールのステータスを確認することもできます。

```
$ aws glacier describe-vault --account-id - --vault-name myvault
{
  "SizeInBytes": 3178496,
  "VaultARN": "arn:aws:glacier:us-west-2:123456789012:vaults/myvault",
  "LastInventoryDate": "2018-12-07T00:26:19.028Z",
  "NumberOfArchives": 1,
  "CreationDate": "2018-12-06T21:23:45.708Z",
  "VaultName": "myvault"
}
```

Note

ポールのステータスは約 1 日 1 回更新されます。詳細については、「[Glacier のポールトに関する各種操作](#)」を参照してください。

これで、作成したチャンクおよびハッシュファイルを安全に削除できます。

```
$ rm chunk* hash*
```

マルチパートアップロードの詳細については、[Amazon S3 Glacier デベロッパーガイド](#)の「[パート単位での大きなアーカイブのアップロード](#)」および「[チェックサムの計算](#)」を参照してください。

リソース

AWS CLI リファレンス :

- [aws glacier](#)
- [aws glacier complete-multipart-upload](#)
- [aws glacier create-vault](#)
- [aws glacier describe-vault](#)

- [aws glacier initiate-multipart-upload](#)

サービスリファレンス:

- [Amazon S3 Glacier デベロッパーガイド](#)
- Amazon S3 Glacier デベロッパーガイド、[「パート単位での大きなアーカイブのアップロード」](#)
- Amazon S3 Glacier デベロッパーガイド、[「チェックサムの計算」](#)
- Amazon S3 Glacier デベロッパーガイド、[「バールトに関する各種操作」](#)

IAM での の使用 AWS CLI

の概要 AWS Identity and Access Management

[の概要 AWS Identity and Access Management](#)

AWS Identity and Access Management (IAM) の機能には AWS Command Line Interface、() を使用してアクセスできますAWS CLI。の AWS CLI コマンドを一覧表示するにはIAM、次のコマンドを使用します。

```
aws iam help
```

このトピックでは、の一般的なタスクを実行する AWS CLI コマンドの例を示しますIAM。

コマンドを実行する前に、デフォルトの認証情報を設定します。詳細については、「[の設定 AWS CLI](#)」を参照してください。

IAM サービスの詳細については、[AWS Identity and Access Management 「ユーザーガイド」](#)を参照してください。

トピック

- [IAM ユーザーおよびグループの作成](#)
- [IAM 管理ポリシーのユーザーへのアタッチ](#)
- [IAM ユーザーの初期パスワードの設定](#)
- [IAM ユーザーのアクセスキーの作成](#)

IAM ユーザーおよびグループの作成

グループを作成してそのグループに新しいユーザーを追加するには

1. グループを作成するには、[create-group](#) コマンドを使用します。

```
$ aws iam create-group --group-name MyIamGroup
{
  "Group": {
    "GroupName": "MyIamGroup",
    "CreateDate": "2018-12-14T03:03:52.834Z",
    "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
    "Path": "/"
  }
}
```

2. ユーザーを作成するには、[create-user](#) コマンドを使用します。

```
$ aws iam create-user --user-name MyUser
{
  "User": {
    "UserName": "MyUser",
    "Path": "/",
    "CreateDate": "2018-12-14T03:13:02.581Z",
    "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/MyUser"
  }
}
```

3. ユーザーをグループに追加するには、[add-user-to-group](#) コマンドを使用します。

```
$ aws iam add-user-to-group --user-name MyUser --group-name MyIamGroup
```

4. *MyIamGroup* グループが *MyUser* を含んでいることを確認するには、[get-group](#) コマンドを使用します。

```
$ aws iam get-group --group-name MyIamGroup
{
  "Group": {
    "GroupName": "MyIamGroup",
    "CreateDate": "2018-12-14T03:03:52Z",
```

```

    "GroupId": "AGPAJNUJ2W4IJVEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/MyIamGroup",
    "Path": "/"
  },
  "Users": [
    {
      "UserName": "MyUser",
      "Path": "/",
      "CreateDate": "2018-12-14T03:13:02Z",
      "UserId": "AIDAJY2PE5XUZ4EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/MyUser"
    }
  ],
  "IsTruncated": "false"
}

```

IAM 管理ポリシーのユーザーへのアタッチ

この例のポリシーは、ユーザーに「Power User Access」を付与します。

IAM マネージドポリシーをユーザーにアタッチするには

1. アタッチするポリシーの Amazon リソースネーム (ARN) を決定します。次のコマンドは `list-policies`、を使用してという名前のポリシーARNの を検索しますPowerUserAccess。その後、環境変数ARNに保存します。

```

$ export POLICYARN=$(aws iam list-policies --query 'Policies[?
PolicyName==`PowerUserAccess`].{ARN:Arn}' --output text) ~
$ echo $POLICYARN
arn:aws:iam::aws:policy/PowerUserAccess

```

2. ポリシーをアタッチするには、 [attach-user-policy](#) コマンドを使用し、ポリシー を保持する環境変数を参照しますARN。

```

$ aws iam attach-user-policy --user-name MyUser --policy-arn $POLICYARN

```

3. ポリシーがユーザーにアタッチされたことを確認するには、 [list-attached-user-policies](#) コマンドを実行します。

```

$ aws iam list-attached-user-policies --user-name MyUser
{

```

```
"AttachedPolicies": [  
  {  
    "PolicyName": "PowerUserAccess",  
    "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"  
  }  
]  
}
```

詳細については、「[アクセス管理リソース](#)」を参照してください。このトピックでは、アクセス許可とポリシーの概要へのリンクと、Amazon S3、Amazon EC2、およびその他のサービスにアクセスするためのポリシーの例へのリンクを提供します。

IAM ユーザーの初期パスワードの設定

次のコマンドは、[create-login-profile](#) を使用して指定されたユーザーの初期パスワードを設定します。初回サインイン時、ユーザーは自分だけが知っているパスワードに変更するように求められます。

```
$ aws iam create-login-profile --user-name MyUser --password My!User1Login8P@ssword --  
password-reset-required  
{  
  "LoginProfile": {  
    "UserName": "MyUser",  
    "CreateDate": "2018-12-14T17:27:18Z",  
    "PasswordResetRequired": true  
  }  
}
```

update-login-profile コマンドを使用して、ユーザーのパスワードを変更することができます。

```
$ aws iam update-login-profile --user-name MyUser --password My!User1ADifferentP@ssword
```

IAM ユーザーのアクセスキーの作成

[create-access-key](#) コマンドを使用して、ユーザーのアクセスキーを作成することができます。アクセスキーは、アクセスキー ID とシークレットキーで構成される一連のセキュリティ認証情報です。

ユーザーが作成できるアクセスキーは一度に 2 つのみです。3 番目のセットを作成しようとすると、コマンドは `LimitExceeded` エラーを返します。

```
$ aws iam create-access-key --user-name MyUser
{
  "AccessKey": {
    "UserName": "MyUser",
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "Status": "Active",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "CreateDate": "2018-12-14T17:34:16Z"
  }
}
```

[delete-access-key](#) コマンドを使用して、ユーザーのアクセスキーを削除します。アクセスキー ID を使用して、削除するアクセスキーを指定します。

```
$ aws iam delete-access-key --user-name MyUser --access-key-id AKIAIOSFODNN7EXAMPLE
```

での Amazon S3 の使用 AWS CLI

Amazon Simple Storage Service (Amazon S3) のご紹介

[Amazon Simple Storage Service \(Amazon S3\) のご紹介 - AWSでのクラウドストレージ](#)

AWS Command Line Interface () を使用して、Amazon Simple Storage Service (Amazon S3) の機能にアクセスできます AWS CLI。Amazon S3 は、スケーラビリティと耐久性に優れたオブジェクトストレージサービスです。Amazon S3 は、事実上無制限のストレージ容量を提供するように設計されているため、幅広いデータストレージと管理のニーズに最適なソリューションです。

Amazon S3 では、小さなファイルから大きなデータセットまで、任意の量のデータをオブジェクトの形式で保存および取得できます。各オブジェクトは、バケットと呼ばれるコンテナに格納されます。バケットは、SDKs ツール、および を通じて、AWS Management Console またはプログラムで AWS アクセスおよび管理できます AWS CLI。

基本的なストレージを含む Amazon S3 には、ライフサイクル管理、バージョンング、スケーラビリティ、セキュリティなどのさまざまな機能も用意されています。これらは他の AWS のサービスと統合されているため、ニーズに合わせて拡張できるクラウドベースのソリューションを構築できます。

AWS CLI には、Amazon S3 にアクセスするためのコマンドの 2 つの階層があります。

- `s3` – オブジェクトとバケットの作成、操作、削除、同期などの一般的なタスクの実行を簡素化 AWS CLI する、 専用に作成されたカスタムの高レベルコマンド。
- `s3api` – 高度なAPIオペレーションを実行できるようにするすべての Amazon S3 オペレーションへの直接アクセスを公開します。

このガイドのトピック

- [での高レベル \(s3\) コマンドの使用 AWS CLI](#)
- [での API-Level \(s3api\) コマンドの使用 AWS CLI](#)
- [の Amazon S3 バケッ lifecycle のスクリプト例 AWS CLI](#)

での高レベル (s3) コマンドの使用 AWS CLI

このトピックでは、AWS CLI で `aws s3` コマンドを使用して、Amazon S3 のバケットとオブジェクトを管理するために使用できるコマンドの一部について説明します。このトピックで説明されていないコマンドおよびその他のコマンドの例については、AWS CLI リファレンスの [aws s3](#) コマンドを参照してください。

高レベル `aws s3` コマンドを使用すると、Amazon S3 オブジェクトの管理が容易になります。これらのコマンドにより、Amazon S3 のコンテンツの管理をそれ自体の内部およびローカルディレクトリ内で管理することができます。

トピック

- [前提条件](#)
- [開始する前に](#)
- [バケットを作成する](#)
- [バケットとオブジェクトの一覧表示](#)
- [バケットの削除](#)
- [オブジェクトの削除](#)
- [オブジェクトの移動](#)
- [オブジェクトのコピー](#)
- [オブジェクトの同期](#)
- [s3 コマンドでよく使用されるオプション](#)

• [リソース](#)

前提条件

s3 コマンドを実行するには、次のことが必要です。

- AWS CLIをインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- 使用するプロファイルには、例によって実行される AWS オペレーションを許可するアクセス許可が必要です。
- Amazon S3 の以下の用語を解説します。
 - バケット - トップレベルの Amazon S3 フォルダ。
 - プレフィックス - バケット内の Amazon S3 フォルダ。
 - オブジェクト - Amazon S3 バケットでホストされているすべてのアイテム。

開始する前に

このセクションでは、aws s3 コマンドを使用する前の注意事項をいくつか示します。

大容量オブジェクトのアップロード

aws s3 コマンドを使用して大きなオブジェクトを Amazon S3 バケットにアップロードする場合、AWS CLI は自動的にマルチパートアップロードを実行します。これらの aws s3 コマンドを使用した場合、失敗したアップロードを再開することはできません。

マルチパートアップロードがタイムアウトによって失敗した場合、またはで手動でキャンセルした場合 AWS CLI、はアップロード AWS CLI を停止し、作成されたファイルをクリーンアップします。この処理には数分かかることもあります。

マルチパートアップロードまたはクリーンアッププロセスが kill コマンドまたはシステム障害によってキャンセルされた場合、作成されたファイルは Amazon S3 バケットに残ります。マルチパートアップロードをクリーンアップするには、[s3api abort-multipart-upload](#) コマンドを使用します。

マルチパートコピーのファイルプロパティとタグ

aws s3 名前空間内のコマンド AWS CLI のバージョン 1 を使用して、ある Amazon S3 バケットの場所から別の Amazon S3 バケットの場所にファイルをコピーし、そのオペレーションで[マルチパートコピー](#)を使用する場合、ソースオブジェクトからのファイルプロパティはコピー先オブジェクトにコピーされません。

デフォルトでは、マルチパートコピーを実行するs3名前空間 AWS CLI のバージョン 2 コマンドは、ソースから宛先コピーにすべてのタグとプロパティのセットを転送します。content-type、content-language、content-encoding、content-disposition、cache-control、expires、および metadata です。

これにより、AWS CLI バージョン 1 を使用した場合、Amazon S3 エンドポイントへの追加の AWS API 呼び出しが行われなかった可能性があります。これには、HeadObject、GetObjectTagging、および PutObjectTagging が含まれます。

AWS CLI バージョン 2 コマンドでこのデフォルトの動作を変更する必要がある場合は、`--copy-props` パラメータを使用して次のいずれかのオプションを指定します。

- `default` - デフォルト値。コピーに、ソースオブジェクトにアタッチされたすべてのタグと、マルチパートコピー以外に使用される `--metadata-directive` パラメータの次のプロパティが含まれることを指定します。content-type、content-language、content-encoding、content-disposition、cache-control、expires、metadata。
- `metadata-directive` - `--metadata-directive` マルチパートコピー以外に使用されるパラメータによって包含されるプロパティのみがコピーに含まれることを指定します。タグはコピーされません。
- `none` - コピーにソースオブジェクトのプロパティを含みません。

バケットを作成する

`s3 mb` コマンドを使用して、バケットを作成します。バケット名はグローバルに一意 (Amazon S3 全体で一意) で、DNS 準拠している必要があります。

バケット名には、小文字の英文字、数字、ハイフン、およびピリオドを使用することができます。バケット名の先頭と末尾は文字または数値とし、ハイフンまたは別のピリオドの横にピリオドを使用することはできません。

[Syntax] (構文)

```
$ aws s3 mb <target> [--options]
```

s3 mb の例

次の例では、`s3://amzn-s3-demo-bucket` バケットを作成します。

```
$ aws s3 mb s3://amzn-s3-demo-bucket
```

バケットとオブジェクトの一覧表示

バケット、フォルダ、オブジェクトのいずれかを一覧表示するには、[s3 ls](#) コマンドを使用します。ターゲットまたはオプションを指定せずにコマンドを使用すると、すべてのバケットが一覧表示されます。

[Syntax] (構文)

```
$ aws s3 ls <target> [--options]
```

このコマンドで使用するいくつかの一般的なオプションと例については、「[s3 コマンドでよく使用されるオプション](#)」を参照してください。使用可能なオプションの完全なリストについては、AWS CLI コマンドリファレンスの [s3 ls](#) を参照してください。

s3 ls の例

次の例では、すべての Amazon S3 バケットを一覧表示します。

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucketamzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

次のコマンドでは、バケット内のすべてのオブジェクトとプレフィックスを一覧表示します。この出力例では、プレフィックス `example/` 内に `MyFile1.txt` という名前のファイルが 1 つあります。

```
$ aws s3 ls s3://amzn-s3-demo-bucket
                                PRE example/
2018-12-04 19:05:48              3 MyFile1.txt
```

コマンドに含めることによって、出力を特定のプレフィックスに絞り込むことができます。次のコマンドは、`example/` のオブジェクトを一覧表示します。`bucket-name/example/` (つまり、`bucket-name` プレフィックスでフィルタリング `example/`)。

```
$ aws s3 ls s3://amzn-s3-demo-bucket/example/
2018-12-06 18:59:32              3 MyFile1.txt
```

バケットの削除

バケットを削除するには、[s3 rb](#) コマンドを使用します。

[Syntax] (構文)

```
$ aws s3 rb <target> [--options]
```

s3 rb の例

次の例では、`s3://amzn-s3-demo-bucket` バケットを削除します。

```
$ aws s3 rb s3://amzn-s3-demo-bucket
```

デフォルトでは、オペレーションが成功するにはバケットが空である必要があります。空でないバケットを削除するには、`--force` オプションを含める必要があります。以前に削除されたが保持されているオブジェクトを含む、バージョンングされたバケットを使用している場合、このコマンドでバケットを削除することはできません。すべての内容を削除しておく必要があります。

次の例では、バケット内のすべてのオブジェクトとプレフィックスを削除し、さらにバケット自体を削除します。

```
$ aws s3 rb s3://amzn-s3-demo-bucket --force
```

オブジェクトの削除

バケット内またはローカルディレクトリ内のオブジェクトを削除するには、[s3 rm](#) コマンドを使用します。

[Syntax] (構文)

```
$ aws s3 rm <target> [--options]
```

このコマンドで使用するいくつかの一般的なオプションと例については、「[s3 コマンドでよく使用されるオプション](#)」を参照してください。オプションの完全なリストについては、AWS CLI コマンドリファレンスの「[s3 rm](#)」を参照してください。

s3 rm の例

以下の例では、`filename.txt` から `s3://amzn-s3-demo-bucket/example` が削除されます。

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example/filename.txt
```

以下の例では、`s3://amzn-s3-demo-bucket/example` オプションを使用して `--recursive` からすべてのオブジェクトが削除されます。

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example --recursive
```

オブジェクトの移動

バケットまたはローカルディレクトリからオブジェクトを移動するには、[s3 mv](#) コマンドを使用します。`s3 mv` コマンドは、ソースオブジェクトまたはファイルを指定された宛先にコピーし、ソースオブジェクトまたはファイルを削除します。

[Syntax] (構文)

```
$ aws s3 mv <source> <target> [--options]
```

このコマンドで使用するいくつかの一般的なオプションと例については、「[s3 コマンドでよく使用されるオプション](#)」を参照してください。使用可能なオプションの完全なリストについては、AWS CLI コマンドリファレンスの [s3 mv](#) を参照してください。

Warning

Amazon S3 の送信元ARNsまたは送信先 で任意のタイプのアクセスポイントまたはアクセスポイントエイリアスを使用している場合はURIs、送信元と送信先 Amazon S3 が異なる基盤となるバケットにURIs解決されるように特に注意する必要があります。ソースバケットと宛先バケットが同じ場合、ソースファイルまたはオブジェクトをそれ自体に移動できるため、ソースファイルまたはオブジェクトが誤って削除される可能性があります。送信元バケットと送信先バケットが同じでないことを確認するには、`--validate-same-s3-paths` パラメータを使用するか、環境変数 [AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS](#) を に設定します `true`。

s3 mv の例

次の例では、すべてのオブジェクトを `s3://amzn-s3-demo-bucket/example` から `s3://amzn-s3-demo-bucket/` に移動します。

```
$ aws s3 mv s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

次の例では、`s3 mv` コマンドを使用して、ローカルファイルを現在の作業ディレクトリから Amazon S3 バケットに移動します。

```
$ aws s3 mv filename.txt s3://amzn-s3-demo-bucket
```

次の例では、ファイルを Amazon S3 バケットから現在の作業ディレクトリに移動します。./ では、現在の作業ディレクトリを指定します。

```
$ aws s3 mv s3://amzn-s3-demo-bucket/filename.txt ./
```

オブジェクトのコピー

バケットまたはローカルディレクトリからオブジェクトをコピーするには、[s3 cp](#) コマンドを使用します。

[Syntax] (構文)

```
$ aws s3 cp <source> <target> [--options]
```

標準入力 (stdin) または標準出力 (stdout) へのファイルストリーミングには `dash` パラメータを使用できます。

Warning

を使用している場合 PowerShell、シェルは のエンコーディングを変更CRLFしたり、パイプされた入力または出力CRLFに を追加したり、出力をリダイレクトしたりすることがあります。

`s3 cp` コマンドでは、次の構文を使用して、stdin から指定先のバケットにファイルストリームをアップロードします。

[Syntax] (構文)

```
$ aws s3 cp - <target> [--options]
```

`s3 cp` コマンドは、次の構文を使用して stdout 用の Amazon S3 ファイルストリームをダウンロードします。

[Syntax] (構文)

```
$ aws s3 cp <target> [--options] -
```

このコマンドで使用するいくつかの一般的なオプションと例については、「[s3 コマンドでよく使用されるオプション](#)」を参照してください。オプションの完全なリストについては、AWS CLI コマンドリファレンスの [s3 cp](#) を参照してください。

s3 cp の例

次の例では、すべてのオブジェクトを `s3://amzn-s3-demo-bucket/example` から `s3://amzn-s3-demo-bucket/` にコピーします。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

次の例では、`s3 cp` コマンドを使用して、現在の作業ディレクトリから Amazon S3 バケットにローカルファイルをコピーします。

```
$ aws s3 cp filename.txt s3://amzn-s3-demo-bucket
```

次の例では、ファイルを Amazon S3 バケットから現在の作業ディレクトリにコピーします (./ では現在の作業ディレクトリを指定します)。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt ./
```

次の例では、`echo` を使用して「hello world」というテキストを `s3://bucket-name/filename.txt` ファイルにストリームします。

```
$ echo "hello world" | aws s3 cp - s3://amzn-s3-demo-bucket/filename.txt
```

次の例では、`s3://amzn-s3-demo-bucket/filename.txt` ファイルを `stdout` にストリーミングし、その内容をコンソールに出力します。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt -  
hello world
```

次の例では、`s3://bucket-name/pre` の内容を `stdout` にストリーミングし、`bzip2` コマンドを使用してファイルを圧縮します。この圧縮したファイルを `key.bz2` という名前で `s3://bucket-name` にアップロードします。

```
$ aws s3 cp s3://amzn-s3-demo-bucket/pre - | bzip2 --best | aws s3 cp - s3://amzn-s3-demo-bucket/key.bz2
```

オブジェクトの同期

[s3 sync](#) コマンドはバケットとディレクトリの内容、または 2 つのバケットの内容を同期します。通常、s3 sync は欠落しているか古くなったファイルやオブジェクトを、ソースとターゲットの間でコピーします。ただし、`--delete` オプションを指定して、ソースに存在しないファイルまたはオブジェクトをターゲットから削除することもできます。

[Syntax] (構文)

```
$ aws s3 sync <source> <target> [--options]
```

このコマンドで使用するいくつかの一般的なオプションと例については、「[s3 コマンドでよく使用されるオプション](#)」を参照してください。オプションの完全なリストについては、AWS CLI コマンドリファレンスの [s3 sync](#) を参照してください。

s3 sync の例

次の例では、amzn-s3-demo-bucket という名前のバケット内のパスという名前の Amazon S3 プレフィックスの内容を現在の作業ディレクトリと同期します。

s3 sync は、同期先で同じ名前を持つファイルと、サイズや変更時間が異なるファイルを更新します。出力には、同期中に実行された特定のオペレーションが表示されます。このオペレーションでは、サブディレクトリ MySubdirectory とその内容を s3://amzn-s3-demo-bucket/path/MySubdirectory と再帰的に同期することに注意してください。

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path
upload: MySubdirectory\MyFile3.txt to s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
upload: MyFile2.txt to s3://amzn-s3-demo-bucket/path/MyFile2.txt
upload: MyFile1.txt to s3://amzn-s3-demo-bucket/path/MyFile1.txt
```

次の例 (前の例の拡張) では、`--delete` オプションの使用方法を示します。

```
// Delete local file
$ rm ./MyFile1.txt
```



```
// Attempt sync without --delete option - nothing happens
$ aws s3 sync . s3://amzn-s3-demo-bucket/path

// Sync with deletion - object is deleted from bucket
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete
delete: s3://amzn-s3-demo-bucket/path/MyFile1.txt

// Delete object from bucket
$ aws s3 rm s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
delete: s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt

// Sync with deletion - local file is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MySubdirectory\MyFile3.txt

// Sync with Infrequent Access storage class
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --storage-class STANDARD_IA
```

--delete オプションを使用する場合、--exclude オプションと --include オプションによって、s3 sync オペレーション中に削除するファイルやオブジェクトをフィルタリングできます。この場合、パラメータ文字列で、ターゲットディレクトリまたはバケットに関連して、削除から除外するか、削除に含めるファイルを指定する必要があります。例を以下に示します。

```
Assume local directory and s3://amzn-s3-demo-bucket/path currently in sync and each
contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt
...

// Sync with delete, excluding files that match a pattern. MyFile88.txt is deleted,
while remote MyFile1.txt is not.
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete --exclude "path/MyFile?.txt"
delete: s3://amzn-s3-demo-bucket/path/MyFile88.txt
...

// Sync with delete, excluding MyFile2.rtf - local file is NOT deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete --exclude "./MyFile2.rtf"
download: s3://amzn-s3-demo-bucket/path/MyFile1.txt to MyFile1.txt
...

// Sync with delete, local copy of MyFile2.rtf is deleted
```

```
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MyFile2.rtf
```

s3 コマンドでよく使用されるオプション

このトピックで説明するコマンドでは、次のオプションがよく使用されます。コマンドで使用できるオプションの完全なリストについては、[AWS CLI バージョン 2 リファレンスガイド](#) の特定のコマンドを参照してください。

acl

s3 sync と s3 cp では、--acl オプションを使用できます。これにより、Amazon S3 にコピーされたファイルのアクセス権を設定できます。--acl オプションは、private、public-read、および public-read-write の値を受け入れます。詳細については、Amazon S3ユーザーガイド」の「[CannedACL](#)」を参照してください。

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --acl public-read
```

exclude

s3 cp、s3 mv、s3 sync、s3 rm のいずれかのコマンドを使用する場合、--exclude オプションまたは --include オプションを使用して結果をフィルタリングできます。--exclude オプションは、コマンドからオブジェクトのみを除外するようにルールを設定します。各オプションは、指定した順に適用されます。これを次の例で示します。

```
Local directory contains 3 files:
```

```
MyFile1.txt
MyFile2.rtf
MyFile88.txt
```

```
// Exclude all .txt files, resulting in only MyFile2.rtf being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt"
```

```
// Exclude all .txt files but include all files with the "MyFile*.txt" format,
resulting in, MyFile1.txt, MyFile2.rtf, MyFile88.txt being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
"MyFile*.txt"
```

```
// Exclude all .txt files, but include all files with the "MyFile*.txt" format,
but exclude all files with the "MyFile?.txt" format resulting in, MyFile2.rtf and
MyFile88.txt being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include "MyFile*.txt" --exclude "MyFile?.txt"
```

include

s3 cp、s3 mv、s3 sync、s3 rm のいずれかのコマンドを使用する場合、--exclude オプションまたは --include オプションを使用して結果をフィルタリングできます。--include オプションは、指定したオブジェクトのみをコマンドに含めるようにルールを設定します。各オプションは、指定した順に適用されます。これを次の例で示します。

```
Local directory contains 3 files:
```

```
MyFile1.txt
```

```
MyFile2.rtf
```

```
MyFile88.txt
```

```
// Include all .txt files, resulting in MyFile1.txt and MyFile88.txt being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt"
```

```
// Include all .txt files but exclude all files with the "MyFile*.txt" format,  
resulting in no files being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude "MyFile*.txt"
```

```
// Include all .txt files, but exclude all files with the "MyFile*.txt" format, but  
include all files with the "MyFile?.txt" format resulting in MyFile1.txt being  
copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude "MyFile*.txt" --include "MyFile?.txt"
```

許可

s3 cp、s3 mv、s3 sync の各コマンドには、指定したユーザーやグループにオブジェクトへのアクセス許可を付与できる --grants オプションが含まれています。次の構文を使用して、--grants オプションをアクセス許可リストに設定します。Permission、Grantee_Type、Grantee_ID を独自の値に置き換えます。

[Syntax] (構文)

```
--grants Permission=Grantee_Type=Grantee_ID  
       [Permission=Grantee_Type=Grantee_ID ...]
```

各値には以下の要素が含まれます。

- **Permission** – 付与されたアクセス許可を指定します。read、readacl、writeacl、full のいずれかに設定できます。
- **Grantee_Type** – 被付与者を識別する方法を指定します。uri、emailaddress、id のいずれかに設定できます。
- **Grantee_ID** – に基づいて被付与者を指定します。 **Grantee_Type**.
 - uri – グループの URI。詳細については、「[被付与者とは](#)」を参照してください。
 - emailaddress – アカウントの E メールアドレス。
 - id – アカウントの正規 ID。

Amazon S3 アクセスコントロールの詳細については、「[アクセスコントロール](#)」を参照してください。

次の例では、バケットにオブジェクトをコピーします。これは、オブジェクトへの read アクセス許可を全員に付与し、full アクセス許可 (read、readacl、writeacl) を user@example.com に関連付けられたアカウントに付与します。

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://  
acs.amazonaws.com/groups/global/AllUsers full=emailaddress=user@example.com
```

Amazon S3 にアップロードするオブジェクトについて、デフォルト以外のストレージクラス (REDUCED_REDUNDANCY または STANDARD_IA) を指定することもできます。そのためには、--storage-class オプションを使用します。

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --storage-class REDUCED_REDUNDANCY
```

recursive

このオプションを使用すると、指定のディレクトリ内またはプレフィックス内のすべてのファイルやオブジェクトに対してコマンドが実行されます。次の例では、s3://amzn-s3-demo-bucket/path とそのすべての内容を削除します。

```
$ aws s3 rm s3://amzn-s3-demo-bucket/path --recursive
```

リソース

AWS CLI リファレンス :

- [aws s3](#)
- [aws s3 cp](#)
- [aws s3 mb](#)
- [aws s3 mv](#)
- [aws s3 ls](#)
- [aws s3 rb](#)
- [aws s3 rm](#)
- [aws s3 sync](#)

サービスリファレンス:

- [Amazon S3 ユーザーガイドの Amazon S3 バケットの使用](#) Amazon S3
- [Amazon S3 ユーザーガイドの Amazon S3 オブジェクトの使用](#) Amazon S3
- [Amazon S3 ユーザーガイドのプレフィックスと区切り文字を使用してキーを階層的に一覧表示する](#)
- [Amazon S3 ユーザーガイドの AWS SDK for .NET \(低レベル\) を使用して、S3 バケットへのマルチパートアップロードを中止する](#) Amazon S3

での API-Level (s3api) コマンドの使用 AWS CLI

APIレベルのコマンド (s3apiコマンドセットに含まれる) は、Amazon Simple Storage Service (Amazon S3) への直接アクセスを提供しAPIs、高レベルのs3コマンドで公開されていない一部のオペレーションを有効にします。これらのコマンドは、サービスの機能への APIレベルのアクセスを提供する他の AWS のサービスと同等です。s3 コマンドの詳細については、「[での高レベル \(s3\) コマンドの使用 AWS CLI](#)」を参照してください。

このトピックでは、Amazon S3 にマッピングされた下位レベルのコマンドを使用する方法を示す例を示しますAPIs。さらに、各 S3 API コマンドの例は、[AWS CLI バージョン 2 リファレンスガイド](#) の s3apiセクションにあります。

トピック

- [前提条件](#)
- [カスタムを適用する ACL](#)
- [ログ記録ポリシーの設定](#)

• [リソース](#)

前提条件

s3api コマンドを実行するには、次のことが必要です。

- AWS CLIをインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- 使用するプロファイルには、例によって実行される AWS オペレーションを許可するアクセス許可が必要です。
- Amazon S3 の以下の用語を解説します。
 - バケット - トップレベルの Amazon S3 フォルダ。
 - プレフィックス - バケット内の Amazon S3 フォルダ。
 - オブジェクト - Amazon S3 バケットでホストされているすべてのアイテム。

カスタムを適用する ACL

高レベルのコマンドでは、`--acl` オプションを使用して、定義済みのアクセスコントロールリスト (ACLs) を Amazon S3 オブジェクトに適用できます。ただし、そのコマンドを使用してバケット全体の を設定することはできませんACLs。ただし、これを行うには[put-bucket-acl](#)、APIレベルの コマンドを使用します。

次の例は、2 人の AWS ユーザー (`user1@example.com` と `user2@example.com`) にフルコントロールを付与し、すべてのユーザーに読み取りアクセス許可を付与する方法を示しています。「everyone」の識別子は、パラメータとしてURI渡す特殊な から取得されます。

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-full-control  
'emailaddress="user1@example.com",emailaddress="user2@example.com"' --grant-read  
'uri="http://acs.amazonaws.com/groups/global/AllUsers"'
```

の構築方法の詳細についてはACLs、「Amazon Simple Storage Service APIリファレンス」の[PUT「バケット acl」](#)を参照してください。CLIなどの のs3apiACLコマンドはput-bucket-acl、同じ[短縮引数表記](#)を使用します。

ログ記録ポリシーの設定

API コマンドは、バケットログ記録ポリシーput-bucket-loggingを設定します。

次の例では、AWS ユーザー `user@example.com` にログファイルに対する完全な制御が付与され、すべてのユーザーがログファイルへの読み取りアクセス権を持ちます。put-bucket-acl コマンドは、Amazon S3 ログ配信システム (で指定URI) に、バケットへのログの読み取りと書き込みに必要なアクセス許可を付与するためにも必要であることに注意してください。

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-read-acp 'URI="http://acs.amazonaws.com/groups/s3/LogDelivery"' --grant-write 'URI="http://acs.amazonaws.com/groups/s3/LogDelivery"'
$ aws s3api put-bucket-logging --bucket amzn-s3-demo-bucket --bucket-logging-status file://logging.json
```

前のコマンドで logging.json ファイルの内容は、次のとおりです。

```
{
  "LoggingEnabled": {
    "TargetBucket": "amzn-s3-demo-bucket",
    "TargetPrefix": "amzn-s3-demo-bucketLogs/",
    "TargetGrants": [
      {
        "Grantee": {
          "Type": "AmazonCustomerByEmail",
          "EmailAddress": "user@example.com"
        },
        "Permission": "FULL_CONTROL"
      },
      {
        "Grantee": {
          "Type": "Group",
          "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
        },
        "Permission": "READ"
      }
    ]
  }
}
```

リソース

AWS CLI リファレンス :

- [aws s3api](#)
- [aws s3api put-bucket-acl](#)

- [aws s3api put-bucket-logging](#)

サービスリファレンス:

- [Amazon S3 ユーザーガイドの Amazon S3 バケットの使用](#) Amazon S3
- [Amazon S3 ユーザーガイドの Amazon S3 オブジェクトの使用](#) Amazon S3
- [Amazon S3 ユーザーガイドのプレフィックスと区切り文字を使用してキーを階層的に一覧表示する](#)
- [Amazon S3 ユーザーガイドの AWS SDK for .NET \(低レベル\) を使用して、S3 バケットへのマルチパートアップロードを中止する](#) Amazon S3

の Amazon S3 バケットライフサイクルのスクリプト例 AWS CLI

このトピックでは、AWS Command Line Interface (AWS CLI) を使用した Amazon S3 バケットライフサイクルオペレーション用の bash スクリプトの例を使用します。このスクリプトの例では、一連の [aws s3api](#) コマンドを使用しています。シェルスクリプトは、コマンドラインインターフェイスで実行するように設計されたプログラムです。

トピック

- [開始する前に](#)
- [この例について](#)
- [ファイル](#)
- [リファレンス](#)

開始する前に

以下の例を実行する前に、次のことを完了する必要があります。

- AWS CLI をインストールして設定します。詳細については、「[the section called “インストール/更新”](#)」および「[の認証認証情報とアクセス認証情報 AWS CLI](#)」を参照してください。
- 使用するプロファイルには、例によって実行される AWS オペレーションを許可するアクセス許可が必要です。

- AWS ベストプラクティスとして、このコードの最小権限、またはタスクの実行に必要なアクセス許可のみを付与します。詳細については、IAM「ユーザーガイド」の「[最小権限を付与する](#)」を参照してください。
- このコードは、すべての AWS リージョンでテストされているわけではありません。一部の AWS サービスは、特定のリージョンでのみ利用できます。詳細については、AWS 全般のリファレンスガイドの「[サービスエンドポイントとクォータ](#)」を参照してください。
- このコードを実行すると、AWS アカウントに料金が発生する可能性があります。作業が終了したら、このスクリプトによって作成されたすべてのリソースを確実に削除するのはお客様の責任になります。

Amazon S3 サービスでは、次の用語を使用します。

- バケット - トップレベルの Amazon S3 フォルダ。
- プレフィックス - バケット内の Amazon S3 フォルダ。
- オブジェクト - Amazon S3 バケットでホストされているアイテム。

この例について

この例では、シェルスクリプトファイルの一連の関数を使用して、基本的な Amazon S3 オペレーションを操作する方法を示します。関数は、`bucket-operations.sh` という名前のシェルスクリプトファイルにあります。これらの関数を別のファイルで呼び出すことができます。各スクリプトファイルには、各関数について説明するコメントが含まれています。

各ステップの中間結果を表示するには、`-i` パラメータを指定してスクリプトを実行します。Amazon S3 コンソールを使用して、バケットまたはそのコンテンツの現在のステータスを表示できます。プロンプトで Enter キーを押すと、スクリプトは次のステップに進むだけです。

完全な例とダウンロード可能なスクリプトファイルについては、[のAWS コード例リポジトリのAmazon S3 バケットライフサイクルオペレーション](#)を参照してくださいGitHub。

ファイル

サンプルには以下のファイルが含まれています。

`bucket-operations.sh`

このメインスクリプトファイルは、別のファイルから取得できます。このファイルには、次のタスクを実行する関数が含まれています。

- バケットを作成し、それが存在することを検証する
- ローカルコンピュータからバケットにファイルをコピーする
- あるバケットの場所から別のバケットの場所にファイルをコピーする
- バケットの内容の一覧表示
- バケットからファイルを削除する
- バケットの削除

で [bucket-operations.sh](#) のコードを表示しますGitHub。

test-bucket-operations.sh

シェルスクリプトファイル `test-bucket-operations.sh` は、`bucket-operations.sh` ファイルを取得し、各関数を呼び出すことによって関数を呼び出す方法を示しています。関数を呼び出すと、テストスクリプトは作成したすべてのリソースを削除します。

[test-bucket-operations.sh](#) で のコードを表示しますGitHub。

awsdocs-general.sh

スクリプトファイル `awsdocs-general.sh` には、AWS CLIの高度なコード例で使用される汎用関数が格納されています。

[awsdocs-general.sh](#) で のコードを表示しますGitHub。

リファレンス

AWS CLI リファレンス :

- [aws s3api](#)
- [aws s3api create-bucket](#)
- [aws s3api copy-object](#)
- [aws s3api delete-bucket](#)
- [aws s3api delete-object](#)
- [aws s3api head-bucket](#)
- [aws s3api list-objects](#)
- [aws s3api put-object](#)

その他のリファレンス:

- [Amazon S3 ユーザーガイドの Amazon S3 バケットの使用](#) Amazon S3
- [Amazon S3 ユーザーガイドの Amazon S3 オブジェクトの使用](#) Amazon S3
- および AWS CLI コード例を表示して提供する AWS SDKには、の[AWS コード例リポジトリ](#)を参照してくださいGitHub。

SNS での Amazon へのアクセス AWS CLI

AWS Command Line Interface (SNS) を使用して、Amazon Simple Notification Service (Amazon) の機能にアクセスできますAWS CLI。Amazon の AWS CLI コマンドを一覧表示するにはSNS、次のコマンドを使用します。

```
aws sns help
```

コマンドを実行する前に、デフォルトの認証情報を設定します。詳細については、「[の設定 AWS CLI](#)」を参照してください。

このトピックでは、Amazon の一般的なタスクを実行する AWS CLI コマンドの例を示しますSNS。

トピック

- [トピックの作成](#)
- [トピックへのサブスクライブ](#)
- [トピックへの発行](#)
- [トピックからサブスクリプションを解除する](#)
- [トピックの削除](#)

トピックの作成

トピックを作成するには、[sns create-topic](#) コマンドを使用して、トピックに割り当てる名前を指定します。

```
$ aws sns create-topic --name my-topic
{
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

レスポンスの TopicArn を書き留めます。これは、後でメッセージを発行するために使用します。

トピックへのサブスクライブ

トピックをサブスクライブするには、[sns subscribe](#) コマンドを使用します。

次の例では、email プロトコルと notification-endpoint の E メールアドレスを指定します。

```
$ aws sns subscribe --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
protocol email --notification-endpoint saanvi@example.com
{
  "SubscriptionArn": "pending confirmation"
}
```

AWS は、subscribe コマンドで指定したアドレスに E メールで確認メッセージをすぐに送信します。E メールメッセージには、次のようなテキストが含まれています。

```
You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:123456789012:my-topic
To confirm this subscription, click or visit the following link (If this was in error
no action is necessary):
Confirm subscription
```

受信者がサブスクリプションを確認リンクをクリックすると、受信者のブラウザに次のような情報を含んだ通知メッセージが表示されます。

```
Subscription confirmed!

You have subscribed saanvi@example.com to the topic:my-topic.

Your subscription's id is:
arn:aws:sns:us-west-2:123456789012:my-topic:1328f057-de93-4c15-512e-8bb22EXAMPLE

If it was not your intention to subscribe, click here to unsubscribe.
```

トピックへの発行

トピックのすべての受信者にメッセージを送信するには、[sns publish](#) コマンドを使用します。

次の例は、「Hello World!」というメッセージを指定されたトピックのサブスクライバー全員に送信します。

```
$ aws sns publish --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --  
message "Hello World!"  
{  
  "MessageId": "4e41661d-5eec-5ddf-8dab-2c867EXAMPLE"  
}
```

この例では、AWS は「Hello World!」というテキストで E メールメッセージを送信します。saanvi@example.com に送信します。

トピックからサブスクリプションを解除する

トピックからサブスクライブを解除し、そのトピックに公開されたメッセージの受信を停止するには、[sns unsubscribe](#) コマンドを使用して、サブスクライブを解除するトピックARNの を指定します。

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-  
topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

サブスクライブが正常に解除されたことを確認するには、[sns list-subscriptions](#) コマンドを使用して、ガリストに表示されARNなくなったことを確認します。

```
$ aws sns list-subscriptions
```

トピックの削除

トピックを削除するには、[sns delete-topic](#) コマンドを実行します。

```
$ aws sns delete-topic --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic
```

トピック AWS が正常に削除されたことを確認するには、[sns list-topics](#) コマンドを使用して、トピックガリストに表示されなくなったことを確認します。

```
$ aws sns list-topics
```

AWS CLI コマンドの例

このトピックのコード例は、AWS Command Line Interface で を使用する方法を示しています AWS。

「基本」は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出し方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

「シナリオ」は、1つのサービス内から、または他のAWSのサービスと組み合わせて複数の関数を呼び出し、特定のタスクを実行する方法を示すコード例です。

サービス

- [ACM を使用した例 AWS CLI](#)
- [API を使用したゲートウェイの例 AWS CLI](#)
- [API を使用したゲートウェイHTTPと WebSocket API例 AWS CLI](#)
- [API を使用したゲートウェイ管理APIの例 AWS CLI](#)
- [を使用した App Mesh の例 AWS CLI](#)
- [を使用した App Runner の例 AWS CLI](#)
- [AWS AppConfig を使用した例 AWS CLI](#)
- [を使用した Application Auto Scaling の例 AWS CLI](#)
- [を使用した Application Discovery Service の例 AWS CLI](#)
- [AppRegistry を使用した例 AWS CLI](#)
- [を使用した Athena の例 AWS CLI](#)
- [を使用した Auto Scaling の例 AWS CLI](#)
- [を使用した Auto Scaling Plans の例 AWS CLI](#)
- [AWS Backup を使用した例 AWS CLI](#)
- [AWS Batch を使用した例 AWS CLI](#)
- [AWS Budgets を使用した例 AWS CLI](#)
- [を使用した Amazon Chime の例 AWS CLI](#)
- [を使用した Cloud Control APIの例 AWS CLI](#)
- [AWS Cloud Map を使用した例 AWS CLI](#)
- [AWS Cloud9 を使用した例 AWS CLI](#)
- [AWS CloudFormation を使用した例 AWS CLI](#)
- [CloudFront を使用した例 AWS CLI](#)

- [を使用した Amazon CloudSearch の例 AWS CLI](#)
- [CloudTrail を使用した例 AWS CLI](#)
- [CloudWatch を使用した例 AWS CLI](#)
- [CloudWatch を使用して例をログに記録します AWS CLI](#)
- [CloudWatch を使用したネットワークモニタリングの例 AWS CLI](#)
- [CodeArtifact を使用した例 AWS CLI](#)
- [CodeBuild を使用した例 AWS CLI](#)
- [CodeCommit を使用した例 AWS CLI](#)
- [CodeDeploy を使用した例 AWS CLI](#)
- [CodeGuru を使用したレビュー担当者の例 AWS CLI](#)
- [CodePipeline を使用した例 AWS CLI](#)
- [AWS CodeStar を使用した通知の例 AWS CLI](#)
- [CodeConnections を使用した例 AWS CLI](#)
- [を使用した Amazon Cognito Identity の例 AWS CLI](#)
- [を使用した Amazon Cognito Identity Provider の例 AWS CLI](#)
- [を使用した Amazon Comprehend の例 AWS CLI](#)
- [を使用した Amazon Comprehend Medical の例 AWS CLI](#)
- [AWS Config を使用した例 AWS CLI](#)
- [を使用した Amazon Connect の例 AWS CLI](#)
- [AWS Cost and Usage Report を使用した例 AWS CLI](#)
- [を使用した Cost Explorer Service の例 AWS CLI](#)
- [を使用した Firehose の例 AWS CLI](#)
- [を使用した Amazon Data Lifecycle Manager の例 AWS CLI](#)
- [AWS Data Pipeline を使用した例 AWS CLI](#)
- [DataSync を使用した例 AWS CLI](#)
- [DAX を使用した例 AWS CLI](#)
- [を使用した検出例 AWS CLI](#)
- [を使用した Device Farm の例 AWS CLI](#)
- [AWS Direct Connect を使用した例 AWS CLI](#)
- [AWS Directory Service を使用した例 AWS CLI](#)

- [AWS DMS を使用した例 AWS CLI](#)
- [を使用した Amazon DocumentDB の例 AWS CLI](#)
- [を使用した DynamoDB の例 AWS CLI](#)
- [を使用した DynamoDB Streams の例 AWS CLI](#)
- [を使用した Amazon EC2 の例 AWS CLI](#)
- [を使用した Amazon EC2 Instance Connect の例 AWS CLI](#)
- [を使用した Amazon ECR の例 AWS CLI](#)
- [を使用した Amazon ECR Public の例 AWS CLI](#)
- [を使用した Amazon ECS の例 AWS CLI](#)
- [を使用した Amazon EFS の例 AWS CLI](#)
- [を使用した Amazon EKS の例 AWS CLI](#)
- [を使用した Elastic Beanstalk の例 AWS CLI](#)
- [Elastic Load Balancing - を使用したバージョン 1 の例 AWS CLI](#)
- [Elastic Load Balancing - を使用したバージョン 2 の例 AWS CLI](#)
- [を使用した Elastic Transcoder の例 AWS CLI](#)
- [ElastiCache を使用した例 AWS CLI](#)
- [MediaStore を使用した例 AWS CLI](#)
- [を使用した Amazon EMR の例 AWS CLI](#)
- [を使用した Amazon EMR on EKS の例 AWS CLI](#)
- [EventBridge を使用した例 AWS CLI](#)
- [を使用した Firewall Manager の例 AWS CLI](#)
- [AWS FIS を使用した例 AWS CLI](#)
- [を使用した Amazon GameLift の例 AWS CLI](#)
- [を使用した Global Accelerator の例 AWS CLI](#)
- [AWS Glue を使用した例 AWS CLI](#)
- [GuardDuty を使用した例 AWS CLI](#)
- [AWS Health を使用した例 AWS CLI](#)
- [HealthImaging を使用した例 AWS CLI](#)
- [HealthLake を使用した例 AWS CLI](#)
- [HealthOmics を使用した例 AWS CLI](#)

- [IAM を使用した例 AWS CLI](#)
- [IAM を使用した Access Analyzer の例 AWS CLI](#)
- [を使用した Image Builder の例 AWS CLI](#)
- [を使用した Incident Manager の例 AWS CLI](#)
- [を使用した Incident Manager Contacts の例 AWS CLI](#)
- [を使用した Amazon Inspector の例 AWS CLI](#)
- [AWS IoT を使用した例 AWS CLI](#)
- [AWS IoT 1-Click を使用したデバイスの例 AWS CLI](#)
- [AWS IoT 1-Click を使用したプロジェクトの例 AWS CLI](#)
- [AWS IoT Analytics を使用した例 AWS CLI](#)
- [を使用した Device Advisor の例 AWS CLI](#)
- [AWS IoT data を使用した例 AWS CLI](#)
- [AWS IoT Events を使用した例 AWS CLI](#)
- [AWS IoT Events-Data を使用した例 AWS CLI](#)
- [AWS IoT Greengrass を使用した例 AWS CLI](#)
- [AWS IoT Greengrass V2 を使用した例 AWS CLI](#)
- [AWS IoT Jobs SDK release を使用した例 AWS CLI](#)
- [AWS IoT SiteWise を使用した例 AWS CLI](#)
- [AWS IoT Things Graph を使用した例 AWS CLI](#)
- [AWS IoT Wireless を使用した例 AWS CLI](#)
- [を使用した Amazon IVS の例 AWS CLI](#)
- [を使用した Amazon IVS Chat の例 AWS CLI](#)
- [を使用した Amazon IVS リアルタイムストリーミングの例 AWS CLI](#)
- [を使用した Amazon Kendra の例 AWS CLI](#)
- [を使用した Kinesis の例 AWS CLI](#)
- [AWS KMS を使用した例 AWS CLI](#)
- [を使用した Lake Formation の例 AWS CLI](#)
- [を使用した Lambda の例 AWS CLI](#)
- [を使用した License Manager の例 AWS CLI](#)
- [を使用した Lightsail の例 AWS CLI](#)

- [を使用した Macie の例 AWS CLI](#)
- [を使用した Amazon Managed Grafana の例 AWS CLI](#)
- [MediaConnect を使用した例 AWS CLI](#)
- [MediaConvert を使用した例 AWS CLI](#)
- [MediaLive を使用した例 AWS CLI](#)
- [MediaPackage を使用した例 AWS CLI](#)
- [MediaPackage VOD を使用した例 AWS CLI](#)
- [MediaStore を使用したデータプレーンの例 AWS CLI](#)
- [MediaTailor を使用した例 AWS CLI](#)
- [を使用した MemoryDB の例 AWS CLI](#)
- [を使用した Amazon MSK の例 AWS CLI](#)
- [を使用した Network Manager の例 AWS CLI](#)
- [を使用した Nimble Studio の例 AWS CLI](#)
- [OpenSearch を使用したサービス例 AWS CLI](#)
- [AWS OpsWorks を使用した例 AWS CLI](#)
- [AWS OpsWorks CM を使用した例 AWS CLI](#)
- [を使用した組織の例 AWS CLI](#)
- [AWS Outposts を使用した例 AWS CLI](#)
- [AWS Payment Cryptography を使用した例 AWS CLI](#)
- [AWS Payment Cryptography を使用したデータプレーンの例 AWS CLI](#)
- [を使用した Amazon Pinpoint の例 AWS CLI](#)
- [を使用した Amazon Polly の例 AWS CLI](#)
- [AWS の料金表 を使用した例 AWS CLI](#)
- [AWS Private CA を使用した例 AWS CLI](#)
- [AWS Proton を使用した例 AWS CLI](#)
- [QLDB を使用した例 AWS CLI](#)
- [を使用した Amazon RDS の例 AWS CLI](#)
- [を使用した Amazon RDS Data Service の例 AWS CLI](#)
- [を使用した Amazon RDS Performance Insights の例 AWS CLI](#)
- [を使用した Amazon Redshift の例 AWS CLI](#)

- [を使用した Amazon Rekognition の例 AWS CLI](#)
- [AWS RAM を使用した例 AWS CLI](#)
- [を使用した Resource Explorer の例 AWS CLI](#)
- [を使用したリソースグループの例 AWS CLI](#)
- [を使用したリソースグループのタグ付けAPIの例 AWS CLI](#)
- [AWS RoboMaker を使用した例 AWS CLI](#)
- [を使用した Route 53 の例 AWS CLI](#)
- [を使用した Route 53 ドメイン登録の例 AWS CLI](#)
- [を使用した Route 53 プロファイルの例 AWS CLI](#)
- [を使用した Route 53 Resolver の例 AWS CLI](#)
- [を使用した Amazon S3 の例 AWS CLI](#)
- [を使用した Amazon S3 Control の例 AWS CLI](#)
- [を使用した S3 Glacier の例 AWS CLI](#)
- [を使用した Secrets Manager の例 AWS CLI](#)
- [を使用した Security Hub の例 AWS CLI](#)
- [を使用した Security Lake の例 AWS CLI](#)
- [AWS Serverless Application Repository を使用した例 AWS CLI](#)
- [を使用した Service Catalog の例 AWS CLI](#)
- [を使用した Service Quotas の例 AWS CLI](#)
- [を使用した Amazon SESの例 AWS CLI](#)
- [を使用したシールドの例 AWS CLI](#)
- [を使用した署名者の例 AWS CLI](#)
- [を使用した Snowball の例 AWS CLI](#)
- [を使用した Amazon SNS の例 AWS CLI](#)
- [を使用した Amazon SQS の例 AWS CLI](#)
- [を使用した Storage Gateway の例 AWS CLI](#)
- [AWS STS を使用した例 AWS CLI](#)
- [AWS Support を使用した例 AWS CLI](#)
- [を使用した Amazon SWF の例 AWS CLI](#)
- [を使用した Systems Manager の例 AWS CLI](#)

- [を使用した Amazon Textract の例 AWS CLI](#)
- [を使用した Amazon Transcribe の例 AWS CLI](#)
- [を使用した Amazon Translate の例 AWS CLI](#)
- [Trusted Advisor を使用した例 AWS CLI](#)
- [を使用した検証済みアクセス許可の例 AWS CLI](#)
- [VPC を使用した格子の例 AWS CLI](#)
- [AWS WAF Classic を使用した例 AWS CLI](#)
- [AWS WAF Classic Regional を使用した例 AWS CLI](#)
- [AWS WAFV2 を使用した例 AWS CLI](#)
- [を使用した Amazon WorkDocs の例 AWS CLI](#)
- [を使用した Amazon WorkMail の例 AWS CLI](#)
- [を使用した Amazon WorkMail メッセージフローの例 AWS CLI](#)
- [WorkSpaces を使用した例 AWS CLI](#)
- [を使用した X-Ray の例 AWS CLI](#)

ACM を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますACM。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags-to-certificate

次のコード例は、add-tags-to-certificate を使用する方法を示しています。

AWS CLI

既存のACM証明書にタグを追加するには

次の `add-tags-to-certificate` コマンドは、指定された証明書に 2 つのタグを追加します。複数のタグは空白で区切ります。

```
aws acm add-tags-to-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AddTagsToCertificate](#)」の「」を参照してください。

`delete-certificate`

次のコード例は、`delete-certificate` を使用する方法を示しています。

AWS CLI

アカウントからACM証明書を削除するには

次の`delete-certificate`コマンドは、指定された `arn` を持つ証明書を削除しますARN。

```
aws acm delete-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteCertificate](#)」の「」を参照してください。

`describe-certificate`

次の例は、`describe-certificate` を使用する方法を説明しています。

AWS CLI

ACM証明書に含まれるフィールドを取得するには

次の`describe-certificate`コマンドは、指定された `arn` を持つ証明書のすべてのフィールドを取得しますARN。

```
aws acm describe-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

以下のような出力が表示されます。

```
{
  "Certificate": {
    "CertificateArn":
"arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
    "CreatedAt": 1446835267.0,
    "DomainName": "www.example.com",
    "DomainValidationOptions": [
      {
        "DomainName": "www.example.com",
        "ValidationDomain": "www.example.com",
        "ValidationEmails": [
          "hostmaster@example.com",
          "admin@example.com",
          "owner@example.com.whoisprivacyservice.org",
          "tech@example.com.whoisprivacyservice.org",
          "admin@example.com.whoisprivacyservice.org",
          "postmaster@example.com",
          "webmaster@example.com",
          "administrator@example.com"
        ]
      },
      {
        "DomainName": "www.example.net",
        "ValidationDomain": "www.example.net",
        "ValidationEmails": [
          "postmaster@example.net",
          "admin@example.net",
          "owner@example.net.whoisprivacyservice.org",
          "tech@example.net.whoisprivacyservice.org",
          "admin@example.net.whoisprivacyservice.org",
          "hostmaster@example.net",
          "administrator@example.net",
          "webmaster@example.net"
        ]
      }
    ],
    "InUseBy": [],
    "IssuedAt": 1446835815.0,
```

```
"Issuer": "Amazon",
"KeyAlgorithm": "RSA-2048",
"NotAfter": 1478433600.0,
"NotBefore": 1446768000.0,
"Serial": "0f:ac:b0:a3:8d:ea:65:52:2d:7d:01:3a:39:36:db:d6",
"SignatureAlgorithm": "SHA256WITHRSA",
>Status": "ISSUED",
"Subject": "CN=www.example.com",
"SubjectAlternativeNames": [
  "www.example.com",
  "www.example.net"
]
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCertificate](#)」の「」を参照してください。

export-certificate

次の例は、`export-certificate` を使用する方法を説明しています。

AWS CLI

プライベート CA によって発行されたプライベート証明書をエクスポートするには。

次の`export-certificate` コマンドは、プライベート証明書、証明書チェーン、プライベートキーをディスプレイにエクスポートします。

```
aws acm export-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --passphrase file://path-to-passphrase-file
```

証明書、チェーン、プライベートキーをローカルファイルにエクスポートするには、次のコマンドを使用します。

```
aws acm export-certificate --certificate-arn arn:aws:acm:region:sccount:certificate/12345678-1234-1234-1234-123456789012 --passphrase file://path-to-passphrase-file > c:\temp\export.txt
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ExportCertificate](#)」の「」を参照してください。

get-certificate

次のコード例は、get-certificate を使用する方法を示しています。

AWS CLI

ACM証明書を取得するには

次のget-certificateコマンドは、指定された ARN と証明書チェーンの証明書を取得します。

```
aws acm get-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

以下のような出力が表示されます。

```
{
  "Certificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
  "CertificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
```



```

21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCcAfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAd
BgkqhkiG9w0BCQEWEG5vb25lQGftYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAdBgkqhkiG9w0BCQEWEG5vb25lQGft
YXpvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCcAfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAd
BgkqhkiG9w0BCQEWEG5vb25lQGftYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmZAdBgkqhkiG9w0BCQEWEG5vb25lQGft
YXpvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCertificate](#)」の「」を参照してください。

import-certificate

次の例は、import-certificate を使用する方法を説明しています。

AWS CLI

証明書を にインポートするにはACM。

次のimport-certificateコマンドは、証明書を にインポートしますACM。ファイル名を使用するファイル名で置き換えます。

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportCertificate](#)」の「」を参照してください。

list-certificates

次の例は、list-certificates を使用する方法を説明しています。

AWS CLI

AWS アカウントのACM証明書を一覧表示するには

次のlist-certificatesコマンドは、アカウント内の証明書ARNsの を一覧表示します。

```
aws acm list-certificates
```

このコマンドを処理すると、次のような出力が生成されます。

```
{
  "CertificateSummaryList": [
    {
      "CertificateArn":
"arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012",
      "DomainName": "www.example.com"
    },
    {
      "CertificateArn": "arn:aws:acm:region:account:certificate/aaaaaaaa-bbbb-
cccc-dddd-eeeeeeeeeeee",

```

```
        "DomainName": "www.example.net"
      }
    ]
  }
```

`list-certificates` を呼び出すたびに表示される証明書数を指定できます。例えば、証明書が 4 つあって一度に 2 つまでを表示する場合は、次の例のように `max-items` 引数を 2 に設定します。

```
aws acm list-certificates --max-items 2
```

2 つの証明書 ARNs と `NextToken` 値が表示されます。

```
"CertificateSummaryList": [
  {
    "CertificateArn": "arn:aws:acm:region:account: \
      certificate/12345678-1234-1234-1234-123456789012",
    "DomainName": "www.example.com"
  },
  {
    "CertificateArn": "arn:aws:acm:region:account: \
      certificate/aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "DomainName": "www.example.net"
  }
],
"NextToken": "9f4d9f69-275a-41fe-b58e-2b837bd9ba48"
```

アカウント内の次の 2 つの証明書を表示するには、次の呼び出しで `NextToken` 値を設定します。

```
aws acm list-certificates --max-items 2 --next-token 9f4d9f69-275a-41fe-
b58e-2b837bd9ba48
```

`certificate-statuses` 引数を使用して出力をフィルタリングできます。次のコマンドは、`PENDING_VALIDATION` ステータスの証明書を表示します。

```
aws acm list-certificates --certificate-statuses PENDING_VALIDATION
```

また、`includes` 引数を使用して出力をフィルタリングすることもできます。次のコマンドは、以下のプロパティでフィルタリングされた証明書を表示します。表示される証明書:

- Specify that the RSA algorithm and a 2048 bit key are used to generate key pairs.
- Contain a Key Usage extension that specifies that the certificates can be used to create digital signatures.
- Contain an Extended Key Usage extension that specifies that the certificates can be used for code signing.

```
aws acm list-certificates --max-items 10 --includes
extendedKeyUsage=CODE_SIGNING,keyUsage=DIGITAL_SIGNATURE,keyTypes=RSA_2048
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListCertificates](#)」の「」を参照してください。

list-tags-for-certificate

次の例は、list-tags-for-certificate を使用する方法を説明しています。

AWS CLI

ACM証明書に適用されるタグを一覧表示するには

次の list-tags-for-certificate コマンドは、アカウント内の証明書に適用されたタグを一覧表示します。

```
aws acm list-tags-for-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

このコマンドを処理すると、次のような出力が生成されます。

```
{
  "Tags": [
    {
      "Value": "Website",
      "Key": "Purpose"
    },
    {
      "Value": "Alice",
      "Key": "Admin"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForCertificate](#)」の「」を参照してください。

remove-tags-from-certificate

次の例は、remove-tags-from-certificate を使用する方法を説明しています。

AWS CLI

ACM証明書からタグを削除するには

次の remove-tags-from-certificate コマンドは、指定された証明書から 2 つのタグを削除します。複数のタグは空白で区切ります。

```
aws acm remove-tags-from-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice Key=Purpose,Value=Website
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveTagsFromCertificate](#)」の「」を参照してください。

request-certificate

次の例は、request-certificate を使用する方法を説明しています。

AWS CLI

新しいACM証明書をリクエストするには

次のrequest-certificateコマンドは、DNS検証を使用して www.example.com ドメインの新しい証明書をリクエストします。

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS
```

冪等性トークンを入力して、request-certificate への呼び出しを区別できます。

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --idempotency-token 91adc45q
```

サブジェクトの代替名を 1 つ以上入力して、複数の Apex ドメインを保護する証明書をリクエストできます。

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names www.example.net
```

ウェブサイトへのアクセスに使用できる代替名を入力することもできます。

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names www.example.com
```

アスタリスク (*) をワイルドカードとして使用して、同じドメイン内の複数のサブドメインの証明書を作成できます。

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names *.example.com
```

複数の代替名を入力することもできます。

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
subject-alternative-names b.example.com c.example.com d.example.com
```

検証に E メールを使用している場合は、ドメイン検証オプションを入力して、検証 Eメールの送信先ドメインを指定できます。

```
aws acm request-certificate --domain-name example.com --validation-  
method EMAIL --subject-alternative-names www.example.com --domain-validation-  
options DomainName=example.com,ValidationDomain=example.com
```

次のコマンドは、新しい証明書をリクエストしたときに、証明書の透明性ロギングをオプトアウトします。

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --  
options CertificateTransparencyLoggingPreference=DISABLED --idempotency-token 184627
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RequestCertificate](#)」の「」を参照してください。

resend-validation-email

次の例は、resend-validation-email を使用する方法を説明しています。

AWS CLI

ACM証明書リクエストの検証 E メールを再送信するには

次の resend-validation-email コマンドは、Amazon 認証機関に検証 E メールを適切なアドレスに送信するように指示します。

```
aws acm resend-validation-email --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --domain www.example.com --validation-domain example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResendValidationEmail](#)」の「」を参照してください。

update-certificate-options

次の例は、update-certificate-options を使用する方法を説明しています。

AWS CLI

証明書オプションを更新するには

次のupdate-certificate-optionsコマンドは、証明書の透明性ログ記録をオプトアウトします。

```
aws acm update-certificate-options --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --options CertificateTransparencyLoggingPreference=DISABLED
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCertificateOptions](#)」の「」を参照してください。

API を使用したゲートウェイの例 AWS CLI

次のコード例は、APIGateway AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-api-key

次の例は、create-api-key を使用する方法を説明しています。

AWS CLI

既存の API と Stage で有効になっている API キーを作成するには

コマンド:

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateApiKey](#)」の「」を参照してください。

create-authorizer

次の例は、create-authorizer を使用する方法を説明しています。

AWS CLI

例 1: のトークンベースの API Gateway Custom Authorizer を作成するには API

次のcreate-authorizer例では、トークンベースのオーソライザーを作成します。

```
aws apigateway create-authorizer \  
  --rest-api-id 1234123412 \  
  \
```



```

--name 'First-Token-Custom-Authorizer' \
--type TOKEN \
--authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \
--identity-source 'method.request.header.Authorization' \
--authorizer-result-ttl-in-seconds 300

```

出力:

```

{
  "authType": "custom",
  "name": "First-Token-Custom-Authorizer",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",
  "authorizerResultTtlInSeconds": 300,
  "identitySource": "method.request.header.Authorization",
  "type": "TOKEN",
  "id": "z40xj0"
}

```

例 2: の Cognito ユーザープールベースの API ゲートウェイ カスタム オーソライザーを作成するには API

次の `create-authorizer` 例では、Cognito ユーザープールベースの API ゲートウェイ カスタム オーソライザーを作成します。

```

aws apigateway create-authorizer \
--rest-api-id 1234123412 \
--name 'First-Cognito-Custom-Authorizer' \
--type COGNITO_USER_POOLS \
--provider-arns 'arn:aws:cognito-idp:us-east-1:123412341234:userpool/us-
east-1_aWcZeQbuD' \
--identity-source 'method.request.header.Authorization'

```

出力:

```

{
  "authType": "cognito_user_pools",
  "identitySource": "method.request.header.Authorization",
  "name": "First-Cognito-Custom-Authorizer",
  "providerARNs": [
    "arn:aws:cognito-idp:us-east-1:342398297714:userpool/us-east-1_qWbZzQhzE"
  ]
}

```

```

    ],
    "type": "COGNITO_USER_POOLS",
    "id": "5yid1t"
  }

```

例 3: のリクエストベースの API Gateway Custom Authorizer を作成するには API

次のcreate-authorizer例では、リクエストベースのオーソライザーを作成します。

```

aws apigateway create-authorizer \
  --rest-api-id 1234123412 \
  --name 'First_Request_Custom_Authorizer' \
  --type REQUEST \
  --authorizer-uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/  
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations' \
  --identity-source 'method.request.header.Authorization,context.accountId' \
  --authorizer-result-ttl-in-seconds 300

```

出力:

```

{
  "id": "z40xj0",
  "name": "First_Request_Custom_Authorizer",
  "type": "REQUEST",
  "authType": "custom",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/  
arn:aws:lambda:us-west-2:123412341234:function:customAuthFunction/invocations",
  "identitySource": "method.request.header.Authorization,context.accountId",
  "authorizerResultTtlInSeconds": 300
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAuthorizer](#)」の「」を参照してください。

create-base-path-mapping

次のコード例は、create-base-path-mapping を使用する方法を示しています。

AWS CLI

カスタムドメイン名のベースパスマッピングを作成するには

コマンド:

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-api-id 1234123412 --stage prod --base-path v1
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBasePathMapping](#)」の「」を参照してください。

create-deployment

次の例は、create-deployment を使用する方法を説明しています。

AWS CLI

の設定済みリソースを新しいステージAPIにデプロイするには

コマンド:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-description 'Development Stage' --description 'First deployment to the dev stage'
```

の設定済みリソースを既存のステージAPIにデプロイするには

コマンド:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Second deployment to the dev stage'
```

に設定されたリソースをステージ変数を使用して既存のステージAPIにデプロイするには

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description '開発ステージへの 3 回目のデプロイ' --variables key='value',otherKey='otherValue'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDeployment](#)」の「」を参照してください。

create-domain-name

次の例は、create-domain-name を使用する方法を説明しています。

AWS CLI

カスタムドメイン名を作成するには

コマンド:

```
aws apigateway create-domain-name --domain-name 'my.domain.tld' --  
certificate-name 'my.domain.tld cert' --certificate-arn 'arn:aws:acm:us-  
east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3'
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDomainName](#)」の「」を参照してください。

create-model

次のコード例は、create-model を使用する方法を示しています。

AWS CLI

のモデルを作成するには API

コマンド:

```
aws apigateway create-model --rest-api-id 1234123412 --name 'firstModel' --  
description 'The First Model' --content-type 'application/json' --schema  
'{ "$schema": "http://json-schema.org/draft-04/schema#", "title": "firstModel",  
"type": "object", "properties": { "firstProperty" : { "type": "object",  
"properties": { "key": { "type": "string" } } } } }'
```

出力:

```
{  
  "contentType": "application/json",  
  "description": "The First Model",  
  "name": "firstModel",  
  "id": "2rzg01",  
  "schema": "{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title  
\": \"firstModel\", \"type\": \"object\", \"properties\": { \"firstProperty  
\": { \"type\": \"object\", \"properties\": { \"key\": { \"type\": \"string  
\" } } } } }"}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateModel](#)」の「」を参照してください。

create-resource

次のコード例は、create-resource を使用する方法を示しています。

AWS CLI

でリソースを作成するには API

コマンド:

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-part 'new-resource'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResource](#)」の「」を参照してください。

create-rest-api

次の例は、create-rest-api を使用する方法を説明しています。

AWS CLI

を作成するには API

コマンド:

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first API'
```

既存の API から重複を作成するには API

コマンド:

```
aws apigateway create-rest-api --name 'Copy of My First API' --description 'This is a copy of my first API' --clone-from 1234123412
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRestApi](#)」の「」を参照してください。

create-stage

次のコード例は、create-stage を使用する方法を示しています。

AWS CLI

に既存のデプロイAPIを含むステージを作成するには

コマンド:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --  
description 'Development stage' --deployment-id a1b2c3
```

に既存のデプロイとカスタムステージ変数APIを含むステージを作成するには

コマンド:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev'  
--description 'Development stage' --deployment-id a1b2c3 --variables  
key='value',otherKey='otherValue'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStage](#)」の「」を参照してください。

create-usage-plan-key

次のコード例は、create-usage-plan-key を使用する方法を示しています。

AWS CLI

既存のAPIキーを使用状況プランに関連付ける

コマンド:

```
aws apigateway create-usage-plan-key --usage-plan-id a1b2c3 --key-type "API_KEY" --  
key-id 4vq3yryqm5
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUsagePlanKey](#)」の「」を参照してください。

create-usage-plan

次のコード例は、create-usage-plan を使用する方法を示しています。

AWS CLI

月の初めにリセットされるスロットルとクォータの制限を使用して使用プランを作成するには

コマンド:

```
aws apigateway create-usage-plan --name "New Usage Plan" --description "A new usage plan" --throttle burstLimit=10,rateLimit=5 --quota limit=500,offset=0,period=MONTH
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateUsagePlan](#)」の「」を参照してください。

delete-api-key

次のコード例は、delete-api-key を使用する方法を示しています。

AWS CLI

APIキーを削除するには

コマンド:

```
aws apigateway delete-api-key --api-key 8bk1k8b11k3sB38D9B310enyWT8c09B30lkq0b1k
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteApiKey](#)」の「」を参照してください。

delete-authorizer

次の例は、delete-authorizer を使用する方法を説明しています。

AWS CLI

でカスタムオーソライザーを削除するには API

コマンド:

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAuthorizer](#)」の「」を参照してください。

delete-base-path-mapping

次の例は、delete-base-path-mapping を使用する方法を説明しています。

AWS CLI

カスタムドメイン名のベースパスマッピングを削除するには

コマンド:

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path 'dev'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBasePathMapping](#)」の「」を参照してください。

delete-client-certificate

次の例は、delete-client-certificate を使用する方法を説明しています。

AWS CLI

クライアント証明書を削除するには

コマンド:

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteClientCertificate](#)」の「」を参照してください。

delete-deployment

次のコード例は、delete-deployment を使用する方法を示しています。

AWS CLI

でデプロイを削除するには API

コマンド:

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDeployment](#)」の「」を参照してください。

delete-domain-name

次の例は、delete-domain-name を使用する方法を説明しています。

AWS CLI

カスタムドメイン名を削除するには

コマンド:

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDomainName](#)」の「」を参照してください。

delete-integration-response

次の例は、delete-integration-response を使用する方法を説明しています。

AWS CLI

内の特定のリソース、メソッド、ステータスコードの統合レスポンスを削除するには API

コマンド:

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIntegrationResponse](#)」の「」を参照してください。

delete-integration

次の例は、delete-integration を使用する方法を説明しています。

AWS CLI

で特定のリソースとメソッドの統合を削除するには API

コマンド:

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --  
http-method GET
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIntegration](#)」の「」を参照してください。

delete-method-response

次の例は、delete-method-response を使用する方法を説明しています。

AWS CLI

で特定のリソース、メソッド、ステータスコードのメソッドレスポンスを削除するには API

コマンド:

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3  
--http-method GET --status-code 200
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMethodResponse](#)」の「」を参照してください。

delete-method

次の例は、delete-method を使用する方法を説明しています。

AWS CLI

で特定のリソースのメソッドを削除するには API

コマンド:

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMethod](#)」の「」を参照してください。

delete-model

次のコード例は、delete-model を使用する方法を示しています。

AWS CLI

特定の でモデルを削除するには API

コマンド:

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteModel](#)」の「」を参照してください。

delete-resource

次のコード例は、delete-resource を使用する方法を示しています。

AWS CLI

でリソースを削除するには API

コマンド:

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResource](#)」の「」を参照してください。

delete-rest-api

次のコード例は、delete-rest-api を使用する方法を示しています。

AWS CLI

を削除するには API

コマンド:

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRestApi](#)」の「」を参照してください。

delete-stage

次のコード例は、delete-stage を使用する方法を示しています。

AWS CLI

でステージを削除するには API

コマンド:

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStage](#)」の「」を参照してください。

delete-usage-plan-key

次のコード例は、delete-usage-plan-key を使用する方法を示しています。

AWS CLI

使用プランからAPIキーを削除するには

コマンド:

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUsagePlanKey](#)」の「」を参照してください。

delete-usage-plan

次の例は、delete-usage-plan を使用する方法を説明しています。

AWS CLI

使用状況プランを削除するには

コマンド:

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUsagePlan](#)」の「」を参照してください。

flush-stage-authorizers-cache

次の例は、flush-stage-authorizers-cache を使用する方法を説明しています。

AWS CLI

ステージのすべてのオーソライザーキャッシュエントリをフラッシュするには

コマンド:

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-name dev
```

- API 詳細については、AWS CLI 「コマンドリファレンス[FlushStageAuthorizersCache](#)」の「」を参照してください。

flush-stage-cache

次の例は、flush-stage-cache を使用する方法を説明しています。

AWS CLI

APIのステージのキャッシュをフラッシュするには

コマンド:

```
aws apigateway flush-stage-cache --rest-api-id 1234123412 --stage-name dev
```

- API 詳細については、AWS CLI 「コマンドリファレンス[FlushStageCache](#)」の「」を参照してください。

generate-client-certificate

次のコード例は、generate-client-certificate を使用する方法を示しています。

AWS CLI

クライアント側のSSL証明書を作成するには

コマンド:

```
aws apigateway generate-client-certificate --description 'My First Client Certificate'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateClientCertificate](#)」の「」を参照してください。

get-account

次のコード例は、get-account を使用する方法を示しています。

AWS CLI

API Gateway アカウント設定を取得するには

コマンド:

```
aws apigateway get-account
```

出力:

```
{
  "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogsRole",
  "throttleSettings": {
    "rateLimit": 500.0,
```

```
    "burstLimit": 1000
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAccount](#)」の「」を参照してください。

get-api-key

次のコード例は、get-api-key を使用する方法を示しています。

AWS CLI

特定のAPIキーに関する情報を取得するには

コマンド:

```
aws apigateway get-api-key --api-key 8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k
```

出力:

```
{
  "description": "My first key",
  "enabled": true,
  "stageKeys": [
    "a1b2c3d4e5/dev",
    "e5d4c3b2a1/dev"
  ],
  "lastUpdatedDate": 1456184515,
  "createdDate": 1456184452,
  "id": "8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k",
  "name": "My key"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApiKey](#)」の「」を参照してください。

get-api-keys

次のコード例は、get-api-keys を使用する方法を示しています。

AWS CLI

APIキーのリストを取得するには

コマンド:

```
aws apigateway get-api-keys
```

出力:

```
{
  "items": [
    {
      "description": "My first key",
      "enabled": true,
      "stageKeys": [
        "a1b2c3d4e5/dev",
        "e5d4c3b2a1/dev"
      ],
      "lastUpdatedDate": 1456184515,
      "createdDate": 1456184452,
      "id": "8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k",
      "name": "My key"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApiKeys](#)」の「」を参照してください。

get-authorizer

次のコード例は、get-authorizer を使用する方法を示しています。

AWS CLI

オーソライザーごとのAPIAPIゲートウェイ設定を取得するには

コマンド:

```
aws apigateway get-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3
```


出力:

```
{
  "authorizerResultTtlInSeconds": 300,
  "name": "MyAuthorizer",
  "type": "TOKEN",
  "identitySource": "method.request.header.Authorization",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:authorizer_function/invocations",
  "id": "gfi4n3"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAuthorizer](#)」の「」を参照してください。

get-authorizers

次の例は、get-authorizers を使用する方法を説明しています。

AWS CLI

のオーソライザーのリストを取得するには REST API

コマンド:

```
aws apigateway get-authorizers --rest-api-id 1234123412
```

出力:

```
{
  "items": [
    {
      "name": "MyAuthorizer",
      "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123412341234:function:My_Authorizer_Function/
invocations",
      "authorizerResultTtlInSeconds": 300,
      "identitySource": "method.request.header.Authorization",
      "type": "TOKEN",
      "id": "gfi4n3"
    }
  ]
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAuthorizers](#)」の「」を参照してください。

get-base-path-mapping

次のコード例は、get-base-path-mapping を使用する方法を示しています。

AWS CLI

カスタムドメイン名のベースパスマッピングを取得するには

コマンド:

```
aws apigateway get-base-path-mapping --domain-name subdomain.domain.tld --base-path v1
```

出力:

```
{
  "basePath": "v1",
  "restApiId": "1234w4321e",
  "stage": "api"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBasePathMapping](#)」の「」を参照してください。

get-base-path-mappings

次の例は、get-base-path-mappings を使用する方法を説明しています。

AWS CLI

カスタムドメイン名の基本パスマッピングを取得するには

コマンド:

```
aws apigateway get-base-path-mappings --domain-name subdomain.domain.tld
```

出力:

```
{
  "items": [
    {
      "basePath": "(none)",
      "restApiId": "1234w4321e",
      "stage": "dev"
    },
    {
      "basePath": "v1",
      "restApiId": "1234w4321e",
      "stage": "api"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBasePathMappings](#)」の「」を参照してください。

get-client-certificate

次のコード例は、get-client-certificate を使用する方法を示しています。

AWS CLI

クライアント証明書を取得するには

コマンド:

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetClientCertificate](#)」の「」を参照してください。

get-client-certificates

次のコード例は、get-client-certificates を使用する方法を示しています。

AWS CLI

クライアント証明書のリストを取得するには

コマンド:

```
aws apigateway get-client-certificates
```

出力:

```
{
  "items": [
    {
      "pemEncodedCertificate": "-----BEGIN CERTIFICATE----- <certificate
content> -----END CERTIFICATE-----",
      "clientCertificateId": "a1b2c3",
      "expirationDate": 1483556561,
      "description": "My Client Certificate",
      "createdDate": 1452020561
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetClientCertificates](#)」の「」を参照してください。

get-deployment

次の例は、get-deployment を使用する方法を説明しています。

AWS CLI

デプロイに関する情報を取得するには

コマンド:

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

出力:

```
{
  "description": "myDeployment",
  "id": "ztt4m2",
  "createdDate": 1455218022
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeployment](#)」の「」を参照してください。

get-deployments

次の例は、get-deployments を使用する方法を説明しています。

AWS CLI

のデプロイのリストを取得するには REST API

コマンド:

```
aws apigateway get-deployments --rest-api-id 1234123412
```

出力:

```
{
  "items": [
    {
      "createdDate": 1453797217,
      "id": "0a2b4c",
      "description": "Deployed my API for the first time"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeployments](#)」の「」を参照してください。

get-domain-name

次のコード例は、get-domain-name を使用する方法を示しています。

AWS CLI

カスタムドメイン名に関する情報を取得するには

コマンド:

```
aws apigateway get-domain-name --domain-name api.domain.tld
```

出力:

```
{
  "domainName": "api.domain.tld",
  "distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
  "certificateName": "uploadedCertificate",
  "certificateUploadDate": 1462565487
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDomainName](#)」の「」を参照してください。

get-domain-names

次の例は、get-domain-names を使用する方法を説明しています。

AWS CLI

カスタムドメイン名のリストを取得するには

コマンド:

```
aws apigateway get-domain-names
```

出力:

```
{
  "items": [
    {
      "distributionDomainName": "d9511k3109bkd.cloudfront.net",
      "certificateUploadDate": 1452812505,
      "certificateName": "my_custom_domain-certificate",
      "domainName": "subdomain.domain.tld"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDomainNames](#)」の「」を参照してください。

get-export

次の例は、get-export を使用する方法を説明しています。

AWS CLI

ステージの JSON Swagger テンプレートを取得するには

コマンド:

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

ステージの JSON Swagger テンプレート + API Gateway 拡張を取得するには

コマンド:

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

ステージの JSON Swagger テンプレート + Postman Extensions を取得するには

コマンド:

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetExport](#)」の「」を参照してください。

get-integration-response

次のコード例は、get-integration-response を使用する方法を示しています。

AWS CLI

RESTAPIのリソースで定義されたHTTPメソッドの統合レスポンス設定を取得するには

コマンド:

```
aws apigateway get-integration-response --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET --status-code 200
```

出力:

```
{
  "statusCode": "200",
  "responseTemplates": {
    "application/json": null
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetIntegrationResponse](#)」の「」を参照してください。

get-integration

次のコード例は、get-integration を使用する方法を示しています。

AWS CLI

RESTAPIのリソースで定義されたHTTPメソッドの統合設定を取得するには

コマンド:

```
aws apigateway get-integration --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

出力:

```
{
  "httpMethod": "POST",
  "integrationResponses": {
    "200": {
      "responseTemplates": {
        "application/json": null
      },
      "statusCode": "200"
    }
  },
  "cacheKeyParameters": [],
  "type": "AWS",
  "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
  "cacheNamespace": "y9h6rt"
}
```



```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIntegration](#)」の「」を参照してください。

get-method-response

次のコード例は、get-method-response を使用する方法を示しています。

AWS CLI

のリソースで定義されたメソッドのHTTPメソッドレスポンスRESTAPIリソース設定を取得するには

コマンド:

```
aws apigateway get-method-response --rest-api-id 1234123412 --resource-id y9h6rt --  
http-method GET --status-code 200
```

出力:

```
{  
  "responseModels": {  
    "application/json": "Empty"  
  },  
  "statusCode": "200"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMethodResponse](#)」の「」を参照してください。

get-method

次のコード例は、get-method を使用する方法を示しています。

AWS CLI

のリソースで定義されたメソッドのHTTPメソッドRESTAPIリソース設定を取得するには

コマンド:

```
aws apigateway get-method --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

出力:

```
{
  "apiKeyRequired": false,
  "httpMethod": "GET",
  "methodIntegration": {
    "integrationResponses": {
      "200": {
        "responseTemplates": {
          "application/json": null
        },
        "statusCode": "200"
      }
    },
    "cacheKeyParameters": [],
    "uri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:My_Function/invocations",
    "httpMethod": "POST",
    "cacheNamespace": "y9h6rt",
    "type": "AWS"
  },
  "requestParameters": {},
  "methodResponses": {
    "200": {
      "responseModels": {
        "application/json": "Empty"
      },
      "statusCode": "200"
    }
  },
  "authorizationType": "NONE"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMethod](#)」の「」を参照してください。

get-model-template

次の例は、get-model-template を使用する方法を説明しています。

AWS CLI

で定義されたモデルのマッピングテンプレートを取得するには REST API

コマンド:

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

出力:

```
{
  "value": "#set($inputRoot = $input.path('$'))\n{ }"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetModelTemplate](#)」の「」を参照してください。

get-model

次のコード例は、get-model を使用する方法を示しています。

AWS CLI

で定義されたモデルの設定を取得するには REST API

コマンド:

```
aws apigateway get-model --rest-api-id 1234123412 --model-name Empty
```

出力:

```
{
  "contentType": "application/json",
  "description": "This is a default empty schema model",
  "name": "Empty",
  "id": "etd5w5",
  "schema": "{\n  \"\n  \"$schema\": \"http://json-schema.org/draft-04/schema#\",
  \"title\": \"Empty Schema\",
  \"type\": \"object\"\n}"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetModel](#)」の「」を参照してください。

get-models

次の例は、get-models を使用する方法を説明しています。

AWS CLI

のモデルのリストを取得するには REST API

コマンド:

```
aws apigateway get-models --rest-api-id 1234123412
```

出力:

```
{
  "items": [
    {
      "description": "This is a default error schema model",
      "schema": "{\n  \"\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#\n  \",\n  \"title\" : \"Error Schema\",\n  \"type\" : \"object\",\n  \"properties\" :\n  {\n    \"message\" : { \"type\" : \"string\" }\n  }\n  }",
      "contentType": "application/json",
      "id": "7tpbze",
      "name": "Error"
    },
    {
      "description": "This is a default empty schema model",
      "schema": "{\n  \"\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#\n  \",\n  \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n  }",
      "contentType": "application/json",
      "id": "etd5w5",
      "name": "Empty"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetModels](#)」の「」を参照してください。

get-resource

次のコード例は、get-resource を使用する方法を示しています。

AWS CLI

リソースに関する情報を取得するには

コマンド:

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

出力:

```
{
  "path": "/path",
  "pathPart": "path",
  "id": "zwo0y3",
  "parentId": "uyokt6ij2g"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResource](#)」の「」を参照してください。

get-resources

次のコード例は、get-resources を使用する方法を示しています。

AWS CLI

のリソースのリストを取得するには REST API

コマンド:

```
aws apigateway get-resources --rest-api-id 1234123412
```

出力:

```
{
  "items": [
    {
```

```
    "path": "/resource/subresource",
    "resourceMethods": {
      "POST": {}
    },
    "id": "024ace",
    "pathPart": "subresource",
    "parentId": "ai5b02"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResources](#)」の「」を参照してください。

get-rest-api

次のコード例は、get-rest-api を使用する方法を示しています。

AWS CLI

に関する情報を取得するには API

コマンド:

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

出力:

```
{
  "name": "myAPI",
  "id": "o1y243m4f5",
  "createdDate": 1453416433
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRestApi](#)」の「」を参照してください。

get-rest-apis

次の例は、get-rest-apis を使用する方法を説明しています。

AWS CLI

のリストを取得するには REST APIs

コマンド:

```
aws apigateway get-rest-apis
```

出力:

```
{
  "items": [
    {
      "createdDate": 1438884790,
      "id": "12s44z21rb",
      "name": "My First API"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRestApis](#)」の「」を参照してください。

get-sdk

次の例は、get-sdk を使用する方法を説明しています。

AWS CLI

RESTAPIステージの Android SDK を取得するには

コマンド:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-type android
--parameters
groupId='com.mycompany',invokerPackage='com.mycompany.clientsdk',artifactId='Mycompany-
client',artifactVersion='1.0.0' /path/to/android_sdk.zip
```

出力:

```
{
```

```
"contentType": "application/octet-stream",
"contentDisposition": "attachment; filename=\"android_2016-02-22_23-52Z.zip\""
}
```

RESTAPIステージIOSSDKの を取得するには

コマンド:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type objectivec --parameters classPrefix='myprefix' /path/to/iOS_sdk.zip
```

出力:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"objectivec_2016-02-22_23-52Z.zip
\""
}
```

SDKRESTAPIステージの Javascript を取得するには

コマンド:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type javascript /path/to/javascript_sdk.zip
```

出力:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"javascript_2016-02-22_23-52Z.zip
\""
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSdk](#)」の「」を参照してください。

get-stage

次の例は、get-stage を使用する方法を説明しています。

AWS CLI

APIのステージに関する情報を取得するには

コマンド:

```
aws apigateway get-stage --rest-api-id 1234123412 --stage-name dev
```

出力:

```
{
  "stageName": "dev",
  "cacheClusterSize": "0.5",
  "cacheClusterEnabled": false,
  "cacheClusterStatus": "NOT_AVAILABLE",
  "deploymentId": "rbh1fj",
  "lastUpdatedDate": 1466802961,
  "createdDate": 1460682074,
  "methodSettings": {
    "*/*": {
      "cacheTtlInSeconds": 300,
      "loggingLevel": "INFO",
      "dataTraceEnabled": false,
      "metricsEnabled": true,
      "unauthorizedCacheControlHeaderStrategy":
"SUCCEED_WITH_RESPONSE_HEADER",
      "throttlingRateLimit": 500.0,
      "cacheDataEncrypted": false,
      "cachingEnabled": false,
      "throttlingBurstLimit": 1000,
      "requireAuthorizationForCacheControl": true
    },
    "~1resource/GET": {
      "cacheTtlInSeconds": 300,
      "loggingLevel": "INFO",
      "dataTraceEnabled": false,
      "metricsEnabled": true,
      "unauthorizedCacheControlHeaderStrategy":
"SUCCEED_WITH_RESPONSE_HEADER",
      "throttlingRateLimit": 500.0,
      "cacheDataEncrypted": false,
      "cachingEnabled": false,
      "throttlingBurstLimit": 1000,
```

```
        "requireAuthorizationForCacheControl": true
    }
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStage](#)」の「」を参照してください。

get-stages

次の例は、get-stages を使用する方法を説明しています。

AWS CLI

のステージのリストを取得するには REST API

コマンド:

```
aws apigateway get-stages --rest-api-id 1234123412
```

出力:

```
{
  "item": [
    {
      "stageName": "dev",
      "cacheClusterSize": "0.5",
      "cacheClusterEnabled": true,
      "cacheClusterStatus": "AVAILABLE",
      "deploymentId": "123h64",
      "lastUpdatedDate": 1456185138,
      "createdDate": 1453589092,
      "methodSettings": {
        "~1resource~1subresource/POST": {
          "cacheTtlInSeconds": 300,
          "loggingLevel": "INFO",
          "dataTraceEnabled": true,
          "metricsEnabled": true,
          "throttlingRateLimit": 500.0,
          "cacheDataEncrypted": false,
          "cachingEnabled": false,
          "throttlingBurstLimit": 1000
        }
      }
    }
  ]
}
```

```
}  
  }  
} ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStages](#)」の「」を参照してください。

get-usage-plan-key

次のコード例は、get-usage-plan-key を使用する方法を示しています。

AWS CLI

使用状況プランに関連付けられたAPIキーの詳細を取得するには

コマンド:

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUsagePlanKey](#)」の「」を参照してください。

get-usage-plan-keys

次のコード例は、get-usage-plan-keys を使用する方法を示しています。

AWS CLI

使用状況プランに関連付けられたAPIキーのリストを取得するには

コマンド:

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUsagePlanKeys](#)」の「」を参照してください。

get-usage-plan

次のコード例は、get-usage-plan を使用する方法を示しています。

AWS CLI

使用状況プランの詳細を取得するには

コマンド:

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUsagePlan](#)」の「」を参照してください。

get-usage-plans

次の例は、get-usage-plans を使用する方法を説明しています。

AWS CLI

すべての使用状況プランの詳細を取得するには

コマンド:

```
aws apigateway get-usage-plans
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUsagePlans](#)」の「」を参照してください。

get-usage

次のコード例は、get-usage を使用する方法を示しています。

AWS CLI

Usage Plan の使用の詳細を取得するには

コマンド:

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-date "2016-08-17"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUsage](#)」の「」を参照してください。

import-rest-api

次の例は、import-rest-api を使用する方法を説明しています。

AWS CLI

Swagger テンプレートをインポートして を作成するには API

コマンド:

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportRestApi](#)」の「」を参照してください。

put-integration-response

次のコード例は、put-integration-response を使用する方法を示しています。

AWS CLI

マッピングテンプレートを定義して、統合レスポンスをデフォルトレスポンスとして作成するには

コマンド:

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200 --selection-pattern "" --response-templates '{"application/json": "{\"json\": \"template\"}"}'
```

regex が 400 でヘッダー値が静的に定義された統合レスポンスを作成するには

コマンド:

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 400 --selection-pattern 400 --response-parameters '{"method.response.header.custom-header": "''''custom-value''''"}'
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutIntegrationResponse](#)」の「」を参照してください。

put-integration

次の例は、put-integration を使用する方法を説明しています。

AWS CLI

MOCK統合リクエストを作成するには

コマンド:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type MOCK --request-templates '{ "application/json": "{\\"statusCode\\": 200}" }'
```

HTTP統合リクエストを作成するには

コマンド:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type HTTP --integration-http-method GET --uri 'https://domain.tld/path'
```

Lambda Function エンドポイントと AWS の統合リクエストを作成するには

コマンド:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type AWS --integration-http-method POST --uri 'arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:function_name/invocations'
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutIntegration](#)」の「」を参照してください。

put-method-response

次の例は、put-method-response を使用する方法を説明しています。

AWS CLI

カスタムメソッドレスポンスヘッダーを使用して、指定したステータスコードでメソッドレスポンスを作成するには

コマンド:

```
aws apigateway put-method-response --rest-api-id 1234123412 --  
resource-id a1b2c3 --http-method GET --status-code 400 --response-  
parameters "method.response.header.custom-header=false"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutMethodResponse](#)」の「」を参照してください。

put-method

次の例は、put-method を使用する方法を説明しています。

AWS CLI

認証なし、APIキーなし、カスタムメソッドリクエストヘッダーAPIを持つでリソースのメソッドを作成するには

コマンド:

```
aws apigateway put-method --rest-api-id 1234123412 --resource-id a1b2c3 --  
http-method PUT --authorization-type "NONE" --no-api-key-required --request-  
parameters "method.request.header.custom-header=false"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutMethod](#)」の「」を参照してください。

put-rest-api

次の例は、put-rest-api を使用する方法を説明しています。

AWS CLI

Swagger テンプレートAPIを使用して既存のを上書きするには

コマンド:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body  
'fileb:///path/to/API_Swagger_template.json'
```

Swagger テンプレートを既存の にマージするには API

コマンド:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///  
path/to/API_Swagger_template.json'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRestApi](#)」の「」を参照してください。

test-invoke-authorizer

次のコード例は、test-invoke-authorizer を使用する方法を示しています。

AWS CLI

必要なヘッダーと値を含むカスタムオーソライザーへのリクエストをテストするには

コマンド:

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-  
id 5yid1t --headers Authorization='Value'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[TestInvokeAuthorizer](#)」の「」を参照してください。

test-invoke-method

次の例は、test-invoke-method を使用する方法を説明しています。

AWS CLI

GETリクエストAPIを実行して でルートリソースをテストするには

コマンド:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id av15sg8fw8  
--http-method GET --path-with-query-string  '/'
```


パスパラメータ値を指定APIしてGETリクエストを実行して、 でサブリソースをテストするには
コマンド:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --  
http-method GET --path-with-query-string '/pets/1'
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[TestInvokeMethod](#)」の「」を参照してください。

update-account

次のコード例は、update-account を使用する方法を示しています。

AWS CLI

CloudWatch Logs へのログ記録ARNのIAMロールを変更するには

コマンド:

```
aws apigateway update-account --patch-operations op='replace',path='/  
cloudwatchRoleArn',value='arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogs'
```

出力:

```
{  
  "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/  
APIGatewayToCloudWatchLogs",  
  "throttleSettings": {  
    "rateLimit": 1000.0,  
    "burstLimit": 2000  
  }  
}
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[UpdateAccount](#)」の「」を参照してください。

update-api-key

次の例は、update-api-key を使用する方法を説明しています。

AWS CLI

APIキーの名前を変更するには

コマンド:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --patch-operations op='replace',path='/name',value='newName'
```

出力:

```
{
  "description": "currentDescription",
  "enabled": true,
  "stageKeys": [
    "41t2j324r5/dev"
  ],
  "lastUpdatedDate": 1470086052,
  "createdDate": 1445460347,
  "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
  "name": "newName"
}
```

APIキーを無効にするには

コマンド:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --patch-operations op='replace',path='/enabled',value='false'
```

出力:

```
{
  "description": "currentDescription",
  "enabled": false,
  "stageKeys": [
    "41t2j324r5/dev"
  ],
  "lastUpdatedDate": 1470086052,
  "createdDate": 1445460347,
  "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
  "name": "newName"
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateApiKey](#)」の「」を参照してください。

update-authorizer

次の例は、update-authorizer を使用する方法を説明しています。

AWS CLI

Custom Authorizer の名前を変更するには

コマンド:

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --patch-operations op='replace',path='/name',value='testAuthorizer'
```

出力:

```
{
  "authType": "custom",
  "name": "testAuthorizer",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:customAuthorizer/invocations",
  "authorizerResultTtlInSeconds": 300,
  "identitySource": "method.request.header.Authorization",
  "type": "TOKEN",
  "id": "gfi4n3"
}
```

カスタムオーソライザーによって呼び出される Lambda 関数を変更するには

コマンド:

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --patch-operations op='replace',path='/authorizerUri',value='arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-west-2:123412341234:function:newAuthorizer/invocations'
```

出力:

```
{
  "authType": "custom",
  "name": "testAuthorizer",
  "authorizerUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123412341234:function:newAuthorizer/invocations",
  "authorizerResultTtlInSeconds": 300,
  "identitySource": "method.request.header.Authorization",
  "type": "TOKEN",
  "id": "gfi4n3"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAuthorizer](#)」の「」を参照してください。

update-base-path-mapping

次の例は、update-base-path-mapping を使用する方法を説明しています。

AWS CLI

カスタムドメイン名のベースパスマッピングを変更するには

コマンド:

```
aws apigateway update-base-path-mapping --domain-name api.domain.tld --base-
path prod --patch-operations op='replace',path='/basePath',value='v1'
```

出力:

```
{
  "basePath": "v1",
  "restApiId": "1234123412",
  "stage": "api"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateBasePathMapping](#)」の「」を参照してください。

update-client-certificate

次のコード例は、update-client-certificate を使用する方法を示しています。

AWS CLI

クライアント証明書の説明を更新するには

コマンド:

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-operations op='replace',path='/description',value='My new description'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateClientCertificate](#)」の「」を参照してください。

update-deployment

次の例は、update-deployment を使用する方法を説明しています。

AWS CLI

デプロイの説明を変更するには

コマンド:

```
aws apigateway update-deployment --rest-api-id 1234123412 --deployment-id ztt4m2 --patch-operations op='replace',path='/description',value='newDescription'
```

出力:

```
{
  "description": "newDescription",
  "id": "ztt4m2",
  "createdDate": 1455218022
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDeployment](#)」の「」を参照してください。

update-domain-name

次のコード例は、update-domain-name を使用する方法を示しています。

AWS CLI

カスタムドメイン名の証明書名を変更するには

次のupdate-domain-name例では、カスタムドメインの証明書名を変更します。

```
aws apigateway update-domain-name \  
  --domain-name api.domain.tld \  
  --patch-operations op='replace',path='/certificateArn',value='arn:aws:acm:us-  
west-2:111122223333:certificate/CERTEXAMPLE123EXAMPLE'
```

出力:

```
{  
  "domainName": "api.domain.tld",  
  "distributionDomainName": "d123456789012.cloudfront.net",  
  "certificateArn": "arn:aws:acm:us-west-2:111122223333:certificate/  
CERTEXAMPLE123EXAMPLE",  
  "certificateUploadDate": 1462565487  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「ゲートウェイAPIでのカスタムドメイン名の設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateDomainName](#)」の「」を参照してください。

update-integration-response

次のコード例は、update-integration-response を使用する方法を示しています。

AWS CLI

統合レスポンスヘッダーを '*' の静的マッピングに変更するには

コマンド:

```
aws apigateway update-integration-response --rest-api-id 1234123412 --  
resource-id 3gapai --http-method GET --status-code 200 --patch-operations  
  op='replace',path='/responseParameters/method.response.header.Access-Control-Allow-  
Origin',value='''*'''
```

出力:

```
{
  "statusCode": "200",
  "responseParameters": {
    "method.response.header.Access-Control-Allow-Origin": "*"
  }
}
```

統合レスポンスヘッダーを削除するには

コマンド:

```
aws apigateway update-integration-response --rest-api-id 1234123412 --resource-id 3gapai --http-method GET --status-code 200 --patch-operations op='remove',path='/responseParameters/method.response.header.Access-Control-Allow-Origin'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIntegrationResponse](#)」の「」を参照してください。

update-integration

次のコード例は、update-integration を使用する方法を示しています。

AWS CLI

Input Passthrough で設定された「コンテンツタイプ: application/json」マッピングテンプレートを追加するには

コマンド:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='add',path='/requestTemplates/application~1json'"
```

カスタムテンプレートで設定された「コンテンツタイプ: application/json」マッピングテンプレートを更新 (置き換え) するには

コマンド:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='replace',path='/requestTemplates/  
application~1json',value='{\"example\": \"json\"}'"
```

Input Passthrough で「Content-Type: application/json」に関連付けられたカスタムテンプレートを更新 (置き換え) するには

コマンド:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='replace',path='requestTemplates/application~1json'"
```

「コンテンツタイプ: application/json」 マッピングテンプレートを削除するには

コマンド:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='remove',path='/requestTemplates/application~1json'"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIntegration](#)」の「」を参照してください。

update-method-response

次の例は、update-method-response を使用する方法を説明しています。

AWS CLI

メソッドで 200 レスポンスの新しいメソッドレスポンスヘッダーを作成し、不要として定義するには (デフォルト)

コマンド:


```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200 --patch-operations op="add",path="/responseParameters/method.response.header.custom-header",value="false"
```

メソッドで 200 レスポンスのレスポンスモデルを削除するには

コマンド:

```
aws apigateway update-method-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200 --patch-operations op="remove",path="/responseModels/application~1json"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMethodResponse](#)」の「」を参照してください。

update-method

次の例は、update-method を使用する方法を説明しています。

AWS CLI

例 1: APIキーを必要とするようにメソッドを変更するには

次のupdate-method例では、APIキーを必要とするようにメソッドを変更します。

```
aws apigateway update-method \  
  --rest-api-id 1234123412 \  
  --resource-id a1b2c3 \  
  --http-method GET \  
  --patch-operations op="replace",path="/apiKeyRequired",value="true"
```

出力:

```
{  
  "httpMethod": "GET",  
  "authorizationType": "NONE",  
  "apiKeyRequired": true,  
  "methodResponses": {  
    "200": {  
      "statusCode": "200",
```

```

        "responseModels": {}
    }
},
"methodIntegration": {
    "type": "AWS",
    "httpMethod": "POST",
    "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
    "passthroughBehavior": "WHEN_NO_MATCH",
    "contentHandling": "CONVERT_TO_TEXT",
    "timeoutInMillis": 29000,
    "cacheNamespace": "h7i8j9",
    "cacheKeyParameters": [],
    "integrationResponses": {
        "200": {
            "statusCode": "200",
            "responseTemplates": {}
        }
    }
}
}
}

```

例 2: IAM認証を必要とするようにメソッドを変更するには

次のupdate-method例では、IAM認証を必要とするようにメソッドを変更します。

```

aws apigateway update-method \
  --rest-api-id 1234123412 \
  --resource-id a1b2c3 \
  --http-method GET \
  --patch-operations op="replace",path="/authorizationType",value="AWS_IAM"

```

出力:

```

{
  "httpMethod": "GET",
  "authorizationType": "AWS_IAM",
  "apiKeyRequired": false,
  "methodResponses": {
    "200": {
      "statusCode": "200",
      "responseModels": {}
    }
  }
}

```

```

    },
    "methodIntegration": {
      "type": "AWS",
      "httpMethod": "POST",
      "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
      "passthroughBehavior": "WHEN_NO_MATCH",
      "contentHandling": "CONVERT_TO_TEXT",
      "timeoutInMillis": 29000,
      "cacheNamespace": "h7i8j9",
      "cacheKeyParameters": [],
      "integrationResponses": {
        "200": {
          "statusCode": "200",
          "responseTemplates": {}
        }
      }
    }
  }
}

```

例 3: Lambda 認証を要求するようにメソッドを変更するには

次の update-method 例では、メソッドを必要な Lambda 認証に変更します。

```

aws apigateway update-method --rest-api-id 1234123412 \
  --resource-id a1b2c3 \
  --http-method GET \
  --patch-operations op="replace",path="/authorizationType",value="CUSTOM"
op="replace",path="/authorizerId",value="e4f5g6"

```

出力:

```

{
  "httpMethod": "GET",
  "authorizationType": "CUSTOM",
  "authorizerId" : "e4f5g6",
  "apiKeyRequired": false,
  "methodResponses": {
    "200": {
      "statusCode": "200",
      "responseModels": {}
    }
  }
},

```

```

    "methodIntegration": {
      "type": "AWS",
      "httpMethod": "POST",
      "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-east-1:123456789111:function:hello-world/invocations",
      "passthroughBehavior": "WHEN_NO_MATCH",
      "contentHandling": "CONVERT_TO_TEXT",
      "timeoutInMillis": 29000,
      "cacheNamespace": "h7i8j9",
      "cacheKeyParameters": [],
      "integrationResponses": {
        "200": {
          "statusCode": "200",
          "responseTemplates": {}
        }
      }
    }
  }
}

```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「Gateway を使用した使用プランの作成、設定、テストCLIRESTAPI」](#)と[API「Gateway RESTAPIでのへのアクセスの制御と管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMethod](#)」の「」を参照してください。

update-model

次のコード例は、update-model を使用する方法を示しています。

AWS CLI

でモデルの説明を変更するには API

コマンド:

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/description,value='New Description'
```

でモデルのスキーマを変更するには API

コマンド:

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/schema,value='{" $schema": "http://json-schema.org/draft-04/schema#", "title": "Empty Schema", "type": "object" }'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateModel](#)」の「」を参照してください。

update-resource

次のコード例は、update-resource を使用する方法を示しています。

AWS CLI

リソースを移動し、内の別の親リソースに配置するには API

コマンド:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/parentId,value='3c2b1a'
```

出力:

```
{
  "path": "/resource",
  "pathPart": "resource",
  "id": "1a2b3c",
  "parentId": "3c2b1a"
}
```

でリソース (pathPart) の名前を変更するには API

コマンド:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/pathPart,value=newresourceName
```

出力:

```
{
  "path": "/newresourceName",
}
```

```
"pathPart": "newresourceName",
"id": "1a2b3c",
"parentId": "3c2b1a"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResource](#)」の「」を参照してください。

update-rest-api

次の例は、update-rest-api を使用する方法を説明しています。

AWS CLI

の名前を変更するには API

コマンド:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
op=replace,path=/name,value='New Name'
```

の説明を変更するには API

コマンド:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
op=replace,path=/description,value='New Description'
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRestApi](#)」の「」を参照してください。

update-stage

次の例は、update-stage を使用する方法を説明しています。

AWS CLI

例 1: リソースとメソッドのステージ設定を上書きするには

次のupdate-stage例では、ステージ設定を上書きし、特定のリソースとメソッドの完全なリクエストレスポンスログ記録をオフにします。

```
aws apigateway update-stage \  
  --rest-api-id 1234123412 \  
  --stage-name 'dev' \  
  --patch-operations op=replace,path=~1resourceName/GET/Logging/  
dataTrace,value=false
```

出力:

```
{  
  "deploymentId": "5ubd17",  
  "stageName": "dev",  
  "cacheClusterEnabled": false,  
  "cacheClusterStatus": "NOT_AVAILABLE",  
  "methodSettings": {  
    "~1resourceName/GET": {  
      "metricsEnabled": false,  
      "dataTraceEnabled": false,  
      "throttlingBurstLimit": 5000,  
      "throttlingRateLimit": 10000.0,  
      "cachingEnabled": false,  
      "cacheTtlInSeconds": 300,  
      "cacheDataEncrypted": false,  
      "requireAuthorizationForCacheControl": true,  
      "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"  
    }  
  },  
  "tracingEnabled": false,  
  "createdDate": "2022-07-18T10:11:18-07:00",  
  "lastUpdatedDate": "2022-07-18T10:19:04-07:00"  
}
```

詳細については、「[Amazon API Gateway デベロッパーガイド REST](#)」の「[のステージの設定 API](#)」を参照してください。

例 2: ステージのすべてのリソースとメソッドのAPIステージ設定を更新するには

次のupdate-stage例では、APIステージのすべてのリソースとメソッドの完全なリクエスト/レスポンスログ記録を有効にします。

```
aws apigateway update-stage \  
  --rest-api-id 1234123412 \  
  --stage-name 'dev' \  
  --patch-operations op=replace,path=~1resourceName/GET/Logging/  
dataTrace,value=true
```

```
--patch-operations 'op=replace,path=/*/*/logging/dataTrace,value=true'
```

出力:

```
{
  "deploymentId": "5b1d17",
  "stageName": "dev",
  "cacheClusterEnabled": false,
  "cacheClusterStatus": "NOT_AVAILABLE",
  "methodSettings": {
    "/*/*": {
      "metricsEnabled": false,
      "dataTraceEnabled": true,
      "throttlingBurstLimit": 5000,
      "throttlingRateLimit": 10000.0,
      "cachingEnabled": false,
      "cacheTtlInSeconds": 300,
      "cacheDataEncrypted": false,
      "requireAuthorizationForCacheControl": true,
      "unauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER"
    }
  },
  "tracingEnabled": false,
  "createdDate": "2022-07-18T10:11:18-07:00",
  "lastUpdatedDate": "2022-07-18T10:31:04-07:00"
}
```

詳細については、「[Amazon API Gateway デベロッパーガイド REST](#)」の「[のステージの設定 API](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス UpdateStage](#)」の「」を参照してください。

update-usage-plan

次のコード例は、update-usage-plan を使用する方法を示しています。

AWS CLI

使用状況プランで定義された期間を変更するには

コマンド:


```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/quota/period",value="MONTH"
```

Usage Plan で定義されているクォータ制限を変更するには

コマンド:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/quota/limit",value="500"
```

使用プランで定義されているスロットルレート制限を変更するには

コマンド:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/rateLimit",value="10"
```

使用プランで定義されているスロットルバースト制限を変更するには

コマンド:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/burstLimit",value="20"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUsagePlan](#)」の「」を参照してください。

update-usage

次のコード例は、update-usage を使用する方法を示しています。

AWS CLI

使用状況プランで定義されている現在の期間のAPIキーのクォータを一時的に変更するには

コマンド:

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-  
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/  
remaining",value="50"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUsage](#)」の「」を参照してください。

API を使用したゲートウェイHTTPと WebSocket API例 AWS CLI

次のコード例は、APIGateway と AWS Command Line Interface で を使用してアクションを実行 HTTPし、一般的なシナリオを実装する方法を示しています WebSocket API。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-api-mapping

次の例は、create-api-mapping を使用する方法を説明しています。

AWS CLI

のAPIマッピングを作成するには API

次のcreate-api-mapping例では、 のtestステージをregional.example.comカスタムドメイン名の/myApiパスAPIにマッピングします。

```
aws apigatewayv2 create-api-mapping \  
  --domain-name regional.example.com \  
  --api-mapping-key myApi \  
  --api-id a1b2c3d4 \  
  --stage test
```

出力:

```
{
```

```
"ApiId": "a1b2c3d4",
"ApiMappingId": "0qzs2sy7bh",
"ApiMappingKey": "myApi"
"Stage": "test"
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の [API「Gateway でのリージョンカスタムドメイン名の設定」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [CreateApiMapping](#)」の「」を参照してください。

create-api

次の例は、create-api を使用する方法を説明しています。

AWS CLI

を作成するには HTTP API

次のcreate-api例では、クイック作成HTTPAPIを使用してを作成します。クイック作成を使用して、AWS Lambda またはHTTP統合、デフォルトのキャッチオールルート、および変更を自動的にデプロイするように設定されたデフォルトのステージAPIでを作成できます。次のコマンドは、クイック作成を使用して、Lambda 関数と統合HTTPAPIするを作成します。

```
aws apigatewayv2 create-api \
  --name my-http-api \
  --protocol-type HTTP \
  --target arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function
```

出力:

```
{
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
  "ApiId": "a1b2c3d4",
  "ApiKeySelectionExpression": "$request.header.x-api-key",
  "CreateDate": "2020-04-08T19:05:45+00:00",
  "Name": "my-http-api",
  "ProtocolType": "HTTP",
  "RouteSelectionExpression": "$request.method $request.path"
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の [「APIGateway HTTPAPIでの開発」](#) を参照してください。

を作成するには WebSocket API

次のcreate-api例では、WebSocket API指定された名前でを作成します。

```
aws apigatewayv2 create-api \  
  --name "myWebSocketApi" \  
  --protocol-type WEBSOCKET \  
  --route-selection-expression '$request.body.action'
```

出力:

```
{  
  "ApiKeySelectionExpression": "$request.header.x-api-key",  
  "Name": "myWebSocketApi",  
  "CreateDate": "2018-11-15T06:23:51Z",  
  "ProtocolType": "WEBSOCKET",  
  "RouteSelectionExpression": "'$request.body.action'",  
  "ApiId": "aabbccdee"  
}
```

詳細については、「Amazon Gateway [デベロッパーガイド](#)」の [WebSocket APIAPI「ゲートウェイで」](#) を作成する」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateApi](#)」の「」を参照してください。

create-authorizer

次の例は、create-authorizer を使用する方法を説明しています。

AWS CLI

のオーJWTソライザーを作成するには HTTP API

次のcreate-authorizer例では、Amazon Cognito を ID プロバイダーとして使用するJWTオーソライザーを作成します。

```
aws apigatewayv2 create-authorizer \  

```

```
--name my-jwt-authorizer \  
--api-id a1b2c3d4 \  
--authorizer-type JWT \  
--identity-source '$request.header.Authorization' \  
--jwt-configuration Audience=123456abc,Issuer=https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123
```

出力:

```
{  
  "AuthorizerId": "a1b2c3",  
  "AuthorizerType": "JWT",  
  "IdentitySource": [  
    "$request.header.Authorization"  
  ],  
  "JwtConfiguration": {  
    "Audience": [  
      "123456abc"  
    ],  
    "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"  
  },  
  "Name": "my-jwt-authorizer"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[JWT「オーソライザー HTTPAPIsによる へのアクセスの制御」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateAuthorizer](#)」の「」を参照してください。

create-deployment

次の例は、create-deployment を使用する方法を説明しています。

AWS CLI

のデプロイを作成するには API

次のcreate-deployment例では、 のデプロイを作成しAPI、そのデプロイを のdevステージに関連付けますAPI。

```
aws apigatewayv2 create-deployment \  

```

```
--api-id a1b2c3d4 \  
--stage-name dev
```

出力:

```
{  
  "AutoDeployed": false,  
  "CreateDate": "2020-04-06T23:38:08Z",  
  "DeploymentId": "531z91",  
  "DeploymentStatus": "DEPLOYED"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「デプロイ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDeployment](#)」の「」を参照してください。

create-domain-name

次のコード例は、create-domain-name を使用する方法を示しています。

AWS CLI

カスタムドメイン名を作成するには

次のcreate-domain-name例では、 のリージョン別カスタムドメイン名を作成しますAPI。

```
aws apigatewayv2 create-domain-name \  
  --domain-name regional.example.com \  
  --domain-name-configurations CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

出力:

```
{  
  "ApiMappingSelectionExpression": "$request.basepath",  
  "DomainName": "regional.example.com",  
  "DomainNameConfigurations": [  
    {  
      "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
```

```

    "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
    "EndpointType": "REGIONAL",
    "HostedZoneId": "123456789111",
    "SecurityPolicy": "TLS_1_2",
    "DomainNameStatus": "AVAILABLE"
  }
]
}

```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「Gateway でのリージョンカスタムドメイン名の設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateDomainName](#)」の「」を参照してください。

create-integration

次の例は、create-integration を使用する方法を説明しています。

AWS CLI

統合を作成するには WebSocket API

次のcreate-integration例では、 のモック統合を作成します WebSocket API。

```

aws apigatewayv2 create-integration \
  --api-id aabbccdee \
  --passthrough-behavior WHEN_NO_MATCH \
  --timeout-in-millis 29000 \
  --connection-type INTERNET \
  --integration-type MOCK

```

出力:

```

{
  "ConnectionType": "INTERNET",
  "IntegrationId": "0abcdef",
  "IntegrationResponseSelectionExpression": "${integration.response.statuscode}",
  "IntegrationType": "MOCK",
  "PassthroughBehavior": "WHEN_NO_MATCH",
  "PayloadFormatVersion": "1.0",

```

```
"TimeoutInMillis": 29000
}
```

詳細については、[WebSocket API「Amazon Gateway デベロッパーガイド」のAPI「ゲートウェイで統合リクエストを設定する」](#)を参照してください。 API

HTTPAPI統合を作成するには

次のcreate-integration例では、 の AWS Lambda HTTP 統合を作成しますAPI。

```
aws apigatewayv2 create-integration \  
  --api-id a1b2c3d4 \  
  --integration-type AWS_PROXY \  
  --integration-uri arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --payload-format-version 2.0
```

出力:

```
{  
  "ConnectionType": "INTERNET",  
  "IntegrationId": "0abcdef",  
  "IntegrationMethod": "POST",  
  "IntegrationType": "AWS_PROXY",  
  "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "PayloadFormatVersion": "2.0",  
  "TimeoutInMillis": 30000  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[HTTP「の統合の設定APIs」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateIntegration](#)」の「」を参照してください。

create-route

次のコード例は、create-route を使用する方法を示しています。

AWS CLI

WebSocket または の \$default ルートを作成するには HTTP API

次のcreate-route例では、WebSocket または HTTP の\$defaultルートを作成しますAPI。

```
aws apigatewayv2 create-route \  
  --api-id aabbccdde \  
  --route-key '$default'
```

出力:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteKey": "$default",  
  "RouteId": "1122334"  
}
```

詳細については、Amazon API Gateway デベロッパーガイドの「[のルートの使用 WebSocket APIs](#)」を参照してください。

のルートを作成するには HTTP API

次のcreate-route例では、POSTリクエストを受け入れる `signup` という名前のルートを作成します。

```
aws apigatewayv2 create-route \  
  --api-id aabbccdde \  
  --route-key 'POST /signup'
```

出力:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteKey": "POST /signup",  
  "RouteId": "1122334"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のルートの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRoute](#)」の「」を参照してください。

create-stage

次の例は、create-stage を使用する方法を説明しています。

AWS CLI

ステージを作成するには

次のcreate-stage例では、の dev という名前のステージを作成しますAPI。

```
aws apigatewayv2 create-stage \  
  --api-id a1b2c3d4 \  
  --stage-name dev
```

出力:

```
{  
  "CreateDate": "2020-04-06T23:23:46Z",  
  "DefaultRouteSettings": {  
    "DetailedMetricsEnabled": false  
  },  
  "LastUpdatedDate": "2020-04-06T23:23:46Z",  
  "RouteSettings": {},  
  "StageName": "dev",  
  "StageVariables": {},  
  "Tags": {}  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[のステージの操作 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateStage](#)」の「」を参照してください。

create-vpc-link

次の例は、create-vpc-link を使用する方法を説明しています。

AWS CLI

のVPCリンクを作成するには HTTP API

次のcreate-vpc-link例では、HTTP のVPCリンクを作成しますAPIs。

```
aws apigatewayv2 create-vpc-link \  
  --name MyVpcLink \  
  --subnet-ids subnet-aaaa subnet-bbbb \  
  --security-group-ids sg1234 sg5678
```

出力:

```
{  
  "CreateDate": "2020-04-07T00:11:46Z",  
  "Name": "MyVpcLink",  
  "SecurityGroupIds": [  
    "sg1234",  
    "sg5678"  
  ],  
  "SubnetIds": [  
    "subnet-aaaa",  
    "subnet-bbbb"  
  ],  
  "Tags": {},  
  "VpcLinkId": "abcd123",  
  "VpcLinkStatus": "PENDING",  
  "VpcLinkStatusMessage": "VPC link is provisioning ENIs",  
  "VpcLinkVersion": "V2"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[のVPCリンクの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateVpcLink](#)」の「」を参照してください。

delete-access-log-settings

次のコード例は、delete-access-log-settings を使用する方法を示しています。

AWS CLI

のアクセスログ記録を無効にするには API

次のdelete-access-log-settings例では、の\$defaultステージのアクセスログ設定を削除しますAPI。ステージのアクセスログ記録を無効にするには、そのアクセスログ設定を削除します。

```
aws apigatewayv2 delete-access-log-settings \  
  --api-id a1b2c3d4 \  
  --stage-name '$default'
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[ログ記録の設定 API](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAccessLogSettings](#)」の「」を参照してください。

delete-api-mapping

次のコード例は、delete-api-mapping を使用する方法を示しています。

AWS CLI

APIマッピングを削除するには

次のdelete-api-mapping例では、api.example.comカスタムドメイン名のAPIマッピングを削除します。

```
aws apigatewayv2 delete-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド」の [API「Gateway でのリージョンカスタムドメイン名の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteApiMapping](#)」の「」を参照してください。

delete-api

次の例は、delete-api を使用する方法を説明しています。

AWS CLI

を削除するには API

次のdelete-api例では、 を削除しますAPI。

```
aws apigatewayv2 delete-api \  
  --api-id a1b2c3d4
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド」の「[の使用HTTPAPIs](#)」と「[の使用 WebSocket APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApi](#)」の「」を参照してください。

delete-authorizer

次の例は、delete-authorizer を使用する方法を説明しています。

AWS CLI

オーソライザーを削除するには

次のdelete-authorizer例では、オーソライザーを削除します。

```
aws apigatewayv2 delete-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド」の[JWT「オーソライザー HTTPAPIsによる へのアクセスの制御」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAuthorizer](#)」の「」を参照してください。

delete-cors-configuration

次の例は、delete-cors-configuration を使用する方法を説明しています。

AWS CLI

CORSの設定を削除するには HTTP API

次のdelete-cors-configuration例では、CORSの設定を削除HTTPAPIしてCORSを無効にします。

```
aws apigatewayv2 delete-cors-configuration \  
  --api-id a1b2c3d4
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイドHTTP」の「[CORSの設定API](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCorsConfiguration](#)」の「」を参照してください。

delete-deployment

次のコード例は、delete-deployment を使用する方法を示しています。

AWS CLI

デプロイを削除するには

次のdelete-deployment例では、のデプロイを削除しますAPI。

```
aws apigatewayv2 delete-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id a1b2c3
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「デプロイ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDeployment](#)」の「」を参照してください。

delete-domain-name

次の例は、delete-domain-name を使用する方法を説明しています。

AWS CLI

カスタムドメイン名を削除するには

次のdelete-domain-name例では、カスタムドメイン名を削除します。

```
aws apigatewayv2 delete-domain-name \  
  --domain-name api.example.com
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「Gatewayでのリージョンカスタムドメイン名の設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteDomainName](#)」の「」を参照してください。

delete-integration

次のコード例は、delete-integration を使用する方法を示しています。

AWS CLI

統合を削除するには

次のdelete-integration例では、API統合を削除します。

```
aws apigatewayv2 delete-integration \  
  --api-id a1b2c3d4 \  
  --integration-id a1b2c3
```

このコマンドでは何も出力されません。

詳細については、「Amazon Gateway デベロッパーガイド」の「[の統合の設定HTTPAPIs](#)」および「[の統合の設定 WebSocket API](#)」を参照してください。 API

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteIntegration](#)」の「」を参照してください。

delete-route-settings

次のコード例は、delete-route-settings を使用する方法を示しています。

AWS CLI

ルート設定を削除するには

次のdelete-route-settings例では、指定されたルートのルート設定を削除します。

```
aws apigatewayv2 delete-route-settings \  
  --api-id a1b2c3d4 \  
  --stage-name dev \  
  --route-key 'GET /pets'
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のルートの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRouteSettings](#)」の「」を参照してください。

delete-route

次の例は、delete-route を使用する方法を説明しています。

AWS CLI

ルートを削除するには

次のdelete-route例では、APIルートを削除します。

```
aws apigatewayv2 delete-route \  
  --api-id a1b2c3d4 \  
  --route-id a1b2c3
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のルートの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRoute](#)」の「」を参照してください。

delete-stage

次の例は、delete-stage を使用する方法を説明しています。

AWS CLI

ステージを削除するには

次のdelete-stage例では、のtestステージを削除しますAPI。

```
aws apigatewayv2 delete-stage \  
  --api-id a1b2c3d4 \  
  --stage-name test
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のステージの操作 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス DeleteStage](#)」の「」を参照してください。

delete-vpc-link

次の例は、delete-vpc-link を使用する方法を説明しています。

AWS CLI

のVPCリンクを削除するには HTTP API

次のdelete-vpc-link例では、VPCリンクを削除します。

```
aws apigatewayv2 delete-vpc-link \  
  --vpc-link-id abcd123
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のVPCリンクの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス DeleteVpcLink](#)」の「」を参照してください。

export-api

次の例は、`export-api` を使用する方法を説明しています。

AWS CLI

の OpenAPI 定義をエクスポートするには HTTP API

次の`export-api`例では、`prod` という名前の API ステージの OpenAPI 3.0 定義 `prod` を `prod` という名前の YAML ファイルにエクスポートします `stage-definition.yaml`。エクスポートされた定義ファイルには、デフォルトで API ゲートウェイ 拡張子が含まれています。

```
aws apigatewayv2 export-api \  
  --api-id a1b2c3d4 \  
  --output-type YAML \  
  --specification OAS30 \  
  --stage-name prod \  
  stage-definition.yaml
```

このコマンドでは何も出力されません。

詳細については、「Amazon [API Gateway デベロッパーガイド](#)」の「[ゲートウェイ HTTP API からのエクスポート](#)」を参照してください。 API

- API 詳細については、AWS CLI 「[コマンドリファレンス ExportApi](#)」の「」を参照してください。

get-api-mapping

次の例は、`get-api-mapping` を使用する方法を説明しています。

AWS CLI

カスタムドメイン名の API マッピングに関する情報を取得するには

次の`get-api-mapping`例では、`api.example.com` カスタムドメイン名の API マッピングに関する情報を表示します。

```
aws apigatewayv2 get-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

出力:

```
{
  "ApiId": "a1b2c3d4",
  "ApiMappingId": "a1b2c3d5",
  "ApiMappingKey": "myTestApi"
  "Stage": "test"
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の「[APIGatewayでのリージョンカスタムドメイン名の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApiMapping](#)」の「」を参照してください。

get-api-mappings

次の例は、get-api-mappings を使用する方法を説明しています。

AWS CLI

カスタムドメイン名のAPIマッピングを取得するには

次のget-api-mappings例では、api.example.comカスタムドメイン名のすべてのAPIマッピングのリストを表示します。

```
aws apigatewayv2 get-api-mappings \
  --domain-name api.example.com
```

出力:

```
{
  "Items": [
    {
      "ApiId": "a1b2c3d4",
      "ApiMappingId": "a1b2c3d5",
      "ApiMappingKey": "myTestApi"
      "Stage": "test"
    },
    {
      "ApiId": "a5b6c7d8",
      "ApiMappingId": "a1b2c3d6",
      "ApiMappingKey": "myDevApi"
      "Stage": "dev"
    }
  ]
}
```

```
    },  
  ]  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の「[APIGateway でのリージョンカスタムドメイン名の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApiMappings](#)」の「」を参照してください。

get-api

次のコード例は、get-api を使用する方法を示しています。

AWS CLI

に関する情報を取得するには API

次のget-api例では、に関する情報を表示しますAPI。

```
aws apigatewayv2 get-api \  
  --api-id a1b2c3d4
```

出力:

```
{  
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",  
  "ApiId": "a1b2c3d4",  
  "ApiKeySelectionExpression": "$request.header.x-api-key",  
  "CreateDate": "2020-03-28T00:32:37Z",  
  "Name": "my-api",  
  "ProtocolType": "HTTP",  
  "RouteSelectionExpression": "$request.method $request.path",  
  "Tags": {  
    "department": "finance"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApi](#)」の「」を参照してください。

get-apis

次の例は、get-apis を使用する方法を説明しています。

AWS CLI

のリストを取得するには APIs

次のget-apis例では、現在のユーザーのをすべて一覧表示APIsします。

```
aws apigatewayv2 get-apis
```

出力:

```
{
  "Items": [
    {
      "ApiEndpoint": "wss://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
      "ApiId": "a1b2c3d4",
      "ApiKeySelectionExpression": "$request.header.x-api-key",
      "CreateDate": "2020-04-07T20:21:59Z",
      "Name": "my-websocket-api",
      "ProtocolType": "WEBSOCKET",
      "RouteSelectionExpression": "$request.body.message",
      "Tags": {}
    },
    {
      "ApiEndpoint": "https://a1b2c3d5.execute-api.us-west-2.amazonaws.com",
      "ApiId": "a1b2c3d5",
      "ApiKeySelectionExpression": "$request.header.x-api-key",
      "CreateDate": "2020-04-07T20:23:50Z",
      "Name": "my-http-api",
      "ProtocolType": "HTTP",
      "RouteSelectionExpression": "$request.method $request.path",
      "Tags": {}
    }
  ]
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の「[の使用HTTPAPIs](#)と [の使用WebSocket APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApis](#)」の「」を参照してください。

get-authorizer

次の例は、get-authorizer を使用する方法を説明しています。

AWS CLI

オーソライザーに関する情報を取得するには

次のget-authorizer例では、オーソライザーに関する情報を表示します。

```
aws apigatewayv2 get-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3
```

出力:

```
{  
  "AuthorizerId": "a1b2c3",  
  "AuthorizerType": "JWT",  
  "IdentitySource": [  
    "$request.header.Authorization"  
  ],  
  "JwtConfiguration": {  
    "Audience": [  
      "123456abc"  
    ],  
    "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"  
  },  
  "Name": "my-jwt-authorizer"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[JWT「オーソライザー HTTP APIsによる へのアクセスの制御」](#)を参照してください。

- API 詳細については、「コマンドリファレンス[GetAuthorizer](#)」の「」を参照してください。

AWS CLI

get-authorizers

次のコード例は、get-authorizers を使用する方法を示しています。

AWS CLI

のオーソライザーのリストを取得するには API

次のget-authorizers例では、 のすべてのオーソライザーのリストを表示しますAPI。

```
aws apigatewayv2 get-authorizers \  
  --api-id a1b2c3d4
```

出力:

```
{  
  "Items": [  
    {  
      "AuthorizerId": "a1b2c3",  
      "AuthorizerType": "JWT",  
      "IdentitySource": [  
        "$request.header.Authorization"  
      ],  
      "JwtConfiguration": {  
        "Audience": [  
          "123456abc"  
        ],  
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"  
      },  
      "Name": "my-jwt-authorizer"  
    },  
    {  
      "AuthorizerId": "a1b2c4",  
      "AuthorizerType": "JWT",  
      "IdentitySource": [  
        "$request.header.Authorization"  
      ],  
      "JwtConfiguration": {  
        "Audience": [  
          "6789abcde"  
        ],  
        "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc234"  
      }  
    }  
  ]  
}
```

```
    },  
    "Name": "new-jwt-authorizer"  
  }  
]  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[JWT「オーソライザー HTTPAPIsによるへのアクセスの制御」](#)を参照してください。

- API 詳細については、「コマンドリファレンス[GetAuthorizers](#)」の「」を参照してください。
AWS CLI

get-deployment

次のコード例は、get-deployment を使用する方法を示しています。

AWS CLI

デプロイに関する情報を取得するには

次のget-deployment例では、デプロイに関する情報を表示します。

```
aws apigatewayv2 get-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id abcdef
```

出力:

```
{  
  "AutoDeployed": true,  
  "CreateDate": "2020-04-07T23:58:40Z",  
  "DeploymentId": "abcdef",  
  "DeploymentStatus": "DEPLOYED",  
  "Description": "Automatic deployment triggered by changes to the Api  
configuration"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「デプロイ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeployment](#)」の「」を参照してください。

get-deployments

次の例は、get-deployments を使用する方法を説明しています。

AWS CLI

デプロイのリストを取得するには

次のget-deployments例では、 のすべてのAPIデプロイのリストを表示します。

```
aws apigatewayv2 get-deployments \  
  --api-id a1b2c3d4
```

出力:

```
{  
  "Items": [  
    {  
      "AutoDeployed": true,  
      "CreateDate": "2020-04-07T23:58:40Z",  
      "DeploymentId": "abcdef",  
      "DeploymentStatus": "DEPLOYED",  
      "Description": "Automatic deployment triggered by changes to the Api  
configuration"  
    },  
    {  
      "AutoDeployed": true,  
      "CreateDate": "2020-04-06T00:33:00Z",  
      "DeploymentId": "bcdefg",  
      "DeploymentStatus": "DEPLOYED",  
      "Description": "Automatic deployment triggered by changes to the Api  
configuration"  
    }  
  ]  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「デプロイ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeployments](#)」の「」を参照してください。

get-domain-name

次の例は、get-domain-name を使用する方法を説明しています。

AWS CLI

カスタムドメイン名に関する情報を取得するには

次のget-domain-name例では、カスタムドメイン名に関する情報を表示します。

```
aws apigatewayv2 get-domain-name \  
  --domain-name api.example.com
```

出力:

```
{  
  "ApiMappingSelectionExpression": "$request.basepath",  
  "DomainName": "api.example.com",  
  "DomainNameConfigurations": [  
    {  
      "ApiGatewayDomainName": "d-1234.execute-api.us-west-2.amazonaws.com",  
      "CertificateArn": "arn:aws:acm:us-  
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",  
      "EndpointType": "REGIONAL",  
      "HostedZoneId": "123456789111",  
      "SecurityPolicy": "TLS_1_2",  
      "DomainNameStatus": "AVAILABLE"  
    }  
  ],  
  "Tags": {}  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「Gatewayでのリージョンカスタムドメイン名の設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetDomainName](#)」の「」を参照してください。

get-domain-names

次のコード例は、get-domain-names を使用する方法を示しています。

AWS CLI

カスタムドメイン名のリストを取得するには

次の`get-domain-names`例では、現在のユーザーのすべてのカスタムドメイン名のリストを表示します。

```
aws apigatewayv2 get-domain-names
```

出力:

```
{
  "Items": [
    {
      "ApiMappingSelectionExpression": "$request.basepath",
      "DomainName": "api.example.com",
      "DomainNameConfigurations": [
        {
          "ApiGatewayDomainName": "d-1234.execute-api.us-
west-2.amazonaws.com",
          "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
          "EndpointType": "REGIONAL",
          "HostedZoneId": "123456789111",
          "SecurityPolicy": "TLS_1_2",
          "DomainNameStatus": "AVAILABLE"
        }
      ]
    },
    {
      "ApiMappingSelectionExpression": "$request.basepath",
      "DomainName": "newApi.example.com",
      "DomainNameConfigurations": [
        {
          "ApiGatewayDomainName": "d-5678.execute-api.us-
west-2.amazonaws.com",
          "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
          "EndpointType": "REGIONAL",
          "HostedZoneId": "123456789222",
          "SecurityPolicy": "TLS_1_2",
          "DomainNameStatus": "AVAILABLE"
        }
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「Gatewayでのリージョンカスタムドメイン名の設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetDomainNames](#)」の「」を参照してください。

get-integration

次のコード例は、get-integration を使用する方法を示しています。

AWS CLI

統合に関する情報を取得するには

次のget-integration例では、統合に関する情報を表示します。

```
aws apigatewayv2 get-integration \  
  --api-id a1b2c3d4 \  
  --integration-id a1b2c3
```

出力:

```
{  
  "ApiGatewayManaged": true,  
  "ConnectionType": "INTERNET",  
  "IntegrationId": "a1b2c3",  
  "IntegrationMethod": "POST",  
  "IntegrationType": "AWS_PROXY",  
  "IntegrationUri": "arn:aws:lambda:us-west-2:12356789012:function:hello12",  
  "PayloadFormatVersion": "2.0",  
  "TimeoutInMillis": 30000  
}
```

詳細については、「[Amazon Gateway デベロッパーガイド](#)」の「[の統合の設定HTTPAPIs](#)」および「[の統合の設定 WebSocket API](#)」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIntegration](#)」の「」を参照してください。

get-integrations

次のコード例は、get-integrations を使用する方法を示しています。

AWS CLI

統合のリストを取得するには

次のget-integrations例では、 のすべてのAPI統合のリストを表示します。

```
aws apigatewayv2 get-integrations \  
  --api-id a1b2c3d4
```

出力:

```
{  
  "Items": [  
    {  
      "ApiGatewayManaged": true,  
      "ConnectionType": "INTERNET",  
      "IntegrationId": "a1b2c3",  
      "IntegrationMethod": "POST",  
      "IntegrationType": "AWS_PROXY",  
      "IntegrationUri": "arn:aws:lambda:us-west-2:123456789012:function:my-  
function",  
      "PayloadFormatVersion": "2.0",  
      "TimeoutInMillis": 30000  
    },  
    {  
      "ConnectionType": "INTERNET",  
      "IntegrationId": "a1b2c4",  
      "IntegrationMethod": "ANY",  
      "IntegrationType": "HTTP_PROXY",  
      "IntegrationUri": "https://www.example.com",  
      "PayloadFormatVersion": "1.0",  
      "TimeoutInMillis": 30000  
    }  
  ]  
}
```

詳細については、「[Amazon Gateway デベロッパーガイド](#)」の「[の統合の設定HTTPAPIs](#)」および「[の統合の設定 WebSocket API](#)」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIntegrations](#)」の「」を参照してください。

get-route

次の例は、get-route を使用する方法を説明しています。

AWS CLI

ルートに関する情報を取得するには

次のget-route例では、ルートに関する情報を表示します。

```
aws apigatewayv2 get-route \  
  --api-id a1b2c3d4 \  
  --route-id 72jz1wk
```

出力:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteId": "72jz1wk",  
  "RouteKey": "ANY /pets",  
  "Target": "integrations/a1b2c3"  
}
```

詳細については、「[Amazon API Gateway デベロッパーガイドHTTP](#)」の「[のルートの使用APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRoute](#)」の「」を参照してください。

get-routes

次のコード例は、get-routes を使用する方法を示しています。

AWS CLI

ルートの一覧を取得するには

次のget-routes例では、 のすべてのAPIルートの一覧を表示します。

```
aws apigatewayv2 get-routes \  
  --api-id a1b2c3d4
```

出力:

```
{  
  "Items": [  
    {  
      "ApiKeyRequired": false,  
      "AuthorizationType": "NONE",  
      "RouteId": "72jz1wk",  
      "RouteKey": "ANY /admin",  
      "Target": "integrations/a1b2c3"  
    },  
    {  
      "ApiGatewayManaged": true,  
      "ApiKeyRequired": false,  
      "AuthorizationType": "NONE",  
      "RouteId": "go65gqi",  
      "RouteKey": "$default",  
      "Target": "integrations/a1b2c4"  
    }  
  ]  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のルートの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetRoutes](#)」の「」を参照してください。

get-stage

次のコード例は、get-stage を使用する方法を示しています。

AWS CLI

ステージに関する情報を取得するには

次のget-stage例では、のprodステージに関する情報を表示しますAPI。

```
aws apigatewayv2 get-stage \  
  --api-id a1b2c3d4 \  
  --stage-name prod
```

出力:

```
{  
  "CreateDate": "2020-04-08T00:36:05Z",  
  "DefaultRouteSettings": {  
    "DetailedMetricsEnabled": false  
  },  
  "DeploymentId": "x1zwyv",  
  "LastUpdatedDate": "2020-04-08T00:36:13Z",  
  "RouteSettings": {},  
  "StageName": "prod",  
  "StageVariables": {  
    "function": "my-prod-function"  
  },  
  "Tags": {}  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のステージの操作 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStage](#)」の「」を参照してください。

get-stages

次の例は、get-stages を使用する方法を説明しています。

AWS CLI

ステージのリストを取得するには

次のget-stages例では、のすべてのステージを一覧表示APIします。


```
aws apigatewayv2 get-stages \  
  --api-id a1b2c3d4
```

出力:

```
{  
  "Items": [  
    {  
      "ApiGatewayManaged": true,  
      "AutoDeploy": true,  
      "CreateDate": "2020-04-08T00:08:44Z",  
      "DefaultRouteSettings": {  
        "DetailedMetricsEnabled": false  
      },  
      "DeploymentId": "dty748",  
      "LastDeploymentStatusMessage": "Successfully deployed stage with  
deployment ID 'dty748'",  
      "LastUpdatedDate": "2020-04-08T00:09:49Z",  
      "RouteSettings": {},  
      "StageName": "$default",  
      "StageVariables": {},  
      "Tags": {}  
    },  
    {  
      "AutoDeploy": true,  
      "CreateDate": "2020-04-08T00:35:06Z",  
      "DefaultRouteSettings": {  
        "DetailedMetricsEnabled": false  
      },  
      "LastUpdatedDate": "2020-04-08T00:35:48Z",  
      "RouteSettings": {},  
      "StageName": "dev",  
      "StageVariables": {  
        "function": "my-dev-function"  
      },  
      "Tags": {}  
    },  
    {  
      "CreateDate": "2020-04-08T00:36:05Z",  
      "DefaultRouteSettings": {  
        "DetailedMetricsEnabled": false  
      },  
      "DeploymentId": "x1zwyv",
```

```

        "LastUpdatedDate": "2020-04-08T00:36:13Z",
        "RouteSettings": {},
        "StageName": "prod",
        "StageVariables": {
            "function": "my-prod-function"
        },
        "Tags": {}
    }
]
}

```

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[のステージの操作 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetStages](#)」の「」を参照してください。

get-tags

次のコード例は、get-tags を使用する方法を示しています。

AWS CLI

リソースのタグのリストを取得するには

次のget-tags例では、のすべてのタグを一覧表示APIします。

```
aws apigatewayv2 get-tags \
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4
```

出力:

```
{
  "Tags": {
    "owner": "dev-team",
    "environment": "prod"
  }
}
```

詳細については、「Amazon [API Gateway デベロッパーガイド](#)」の「[ゲートウェイリソースのタグ付け](#)」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTags](#)」の「」を参照してください。

get-vpc-link

次の例は、get-vpc-link を使用する方法を説明しています。

AWS CLI

VPC リンクに関する情報を取得するには

次のget-vpc-link例では、VPCリンクに関する情報を表示します。

```
aws apigatewayv2 get-vpc-link \  
--vpc-link-id abcd123
```

出力:

```
{  
  "CreateDate": "2020-04-07T00:27:47Z",  
  "Name": "MyVpcLink",  
  "SecurityGroupIds": [  
    "sg1234",  
    "sg5678"  
  ],  
  "SubnetIds": [  
    "subnet-aaaa",  
    "subnet-bbbb"  
  ],  
  "Tags": {},  
  "VpcLinkId": "abcd123",  
  "VpcLinkStatus": "AVAILABLE",  
  "VpcLinkStatusMessage": "VPC link is ready to route traffic",  
  "VpcLinkVersion": "V2"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のVPCリンクの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVpcLink](#)」の「」を参照してください。

get-vpc-links

次の例は、get-vpc-links を使用する方法を説明しています。

AWS CLI

VPC リンクのリストを取得するには

次のget-vpc-links例では、現在のユーザーのすべてのVPCリンクのリストを表示します。

```
aws apigatewayv2 get-vpc-links
```

出力:

```
{
  "Items": [
    {
      "CreateDate": "2020-04-07T00:27:47Z",
      "Name": "MyVpcLink",
      "SecurityGroupIds": [
        "sg1234",
        "sg5678"
      ],
      "SubnetIds": [
        "subnet-aaaa",
        "subnet-bbbb"
      ],
      "Tags": {},
      "VpcLinkId": "abcd123",
      "VpcLinkStatus": "AVAILABLE",
      "VpcLinkStatusMessage": "VPC link is ready to route traffic",
      "VpcLinkVersion": "V2"
    }
  ],
  {
    "CreateDate": "2020-04-07T00:27:47Z",
    "Name": "MyOtherVpcLink",
    "SecurityGroupIds": [
      "sg1234",
      "sg5678"
    ],
    "SubnetIds": [
      "subnet-aaaa",
      "subnet-bbbb"
    ]
  }
}
```

```
    ],
    "Tags": {},
    "VpcLinkId": "abcd456",
    "VpcLinkStatus": "AVAILABLE",
    "VpcLinkStatusMessage": "VPC link is ready to route traffic",
    "VpcLinkVersion": "V2"
  }
]
}
```

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[のVPCリンクの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetVpcLinks](#)」の「」を参照してください。

import-api

次のコード例は、import-api を使用する方法を示しています。

AWS CLI

をインポートするには HTTP API

次のimport-api例では、という名前の OpenAPI 3.0 定義ファイルHTTPAPIから を作成しますapi-definition.yaml。

```
aws apigatewayv2 import-api \
  --body file://api-definition.yaml
```

api-definition.yaml の内容:

```
openapi: 3.0.1
info:
  title: My Lambda API
  version: v1.0
paths:
  /hello:
    x-amazon-apigateway-any-method:
      x-amazon-apigateway-integration:
        payloadFormatVersion: 2.0
        type: aws_proxy
```

```
httpMethod: POST
uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:123456789012:function:hello/invocations
connectionType: INTERNET
```

出力:

```
{
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
  "ApiId": "a1b2c3d4",
  "ApiKeySelectionExpression": "$request.header.x-api-key",
  "CreateDate": "2020-04-08T17:19:38+00:00",
  "Name": "My Lambda API",
  "ProtocolType": "HTTP",
  "RouteSelectionExpression": "$request.method $request.path",
  "Tags": {},
  "Version": "v1.0"
}
```

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[のオープンAPI定義の使用APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ImportApi](#)」の「」を参照してください。

reimport-api

次の例は、reimport-api を使用する方法を説明しています。

AWS CLI

を再インポートするには HTTP API

次のreimport-api例では、で指定された OpenAPI 3.0 定義を使用するHTTPAPIのように既存のを更新しますapi-definition.yaml。

```
aws apigatewayv2 reimport-api \
  --body file://api-definition.yaml \
  --api-id a1b2c3d4
```

api-definition.yaml の内容:

```
openapi: 3.0.1
info:
  title: My Lambda API
  version: v1.0
paths:
  /hello:
    x-amazon-apigateway-any-method:
      x-amazon-apigateway-integration:
        payloadFormatVersion: 2.0
        type: aws_proxy
        httpMethod: POST
        uri: arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/
arn:aws:lambda:us-west-2:12356789012:function:hello/invocations
        connectionType: INTERNET
```

出力:

```
{
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
  "ApiId": "a1b2c3d4",
  "ApiKeySelectionExpression": "$request.header.x-api-key",
  "CreateDate": "2020-04-08T17:19:38+00:00",
  "Name": "My Lambda API",
  "ProtocolType": "HTTP",
  "RouteSelectionExpression": "$request.method $request.path",
  "Tags": {},
  "Version": "v1.0"
}
```

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[のオープンAPI定義の使用APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ReimportApi](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、キー名Departmentと の値を持つタグAccountingを指定されたに追加しますAPI。

```
aws apigatewayv2 tag-resource \  
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \  
  --tags Department=Accounting
```

このコマンドでは何も出力されません。

詳細については、「Amazon [API Gateway デベロッパーガイド](#)」の「[Gateway リソースのタグ付け](#)」を参照してください。 API

- API 詳細については、AWS CLI 「 [コマンドリファレンスTagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグを削除する

次のuntag-resource例では、キー名 Projectと を含むタグを指定された Ownerから削除しますAPI。

```
aws apigatewayv2 untag-resource \  
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \  
  --tag-keys Project Owner
```

このコマンドでは何も出力されません。

詳細については、「Amazon [API Gateway デベロッパーガイド](#)」の「[Gateway リソースのタグ付け](#)」を参照してください。 API

- API 詳細については、AWS CLI 「 [コマンドリファレンスUntagResource](#)」の「」を参照してください。

update-api-mapping

次のコード例は、update-api-mapping を使用する方法を示しています。

AWS CLI

APIマッピングを更新するには

次のupdate-api-mapping例では、カスタムドメイン名のAPIマッピングを変更します。その結果、指定された API および ステージのカスタムドメイン名 URL を使用するベースは になりま
ずhttps://api.example.com/dev。

```
aws apigatewayv2 update-api-mapping \  
  --api-id a1b2c3d4 \  
  --stage dev \  
  --domain-name api.example.com \  
  --api-mapping-id 0qzs2sy7bh \  
  --api-mapping-key dev
```

出力:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "0qzs2sy7bh",  
  "ApiMappingKey": "dev"  
  "Stage": "dev"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「Gatewayでのリージョンカスタムドメイン名の設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateApiMapping](#)」の「」を参照してください。

update-api

次の例は、update-api を使用する方法を説明しています。

AWS CLI

CORSで を有効にするには HTTP API

次のupdate-api例では、指定された APICORSの設定を更新して、からのリクエストを許可しますhttps://www.example.com。

```
aws apigatewayv2 update-api \  
  --api-id a1b2c3d4 \  
  --cors-configuration AllowOrigins=https://www.example.com
```

出力:

```
{  
  "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",  
  "ApiId": "a1b2c3d4",  
  "ApiKeySelectionExpression": "$request.header.x-api-key",  
  "CorsConfiguration": {  
    "AllowCredentials": false,  
    "AllowHeaders": [  
      "header1",  
      "header2"  
    ],  
    "AllowMethods": [  
      "GET",  
      "OPTIONS"  
    ],  
    "AllowOrigins": [  
      "https://www.example.com"  
    ]  
  },  
  "CreateDate": "2020-04-08T18:39:37+00:00",  
  "Name": "my-http-api",  
  "ProtocolType": "HTTP",  
  "RouteSelectionExpression": "$request.method $request.path",  
  "Tags": {},  
  "Version": "v1.0"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[CORSの設定 API](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateApi](#)」の「」を参照してください。

update-authorizer

次のコード例は、update-authorizer を使用する方法を示しています。

AWS CLI

オーソライザーを更新するには

次のupdate-authorizer例では、JWTオーソライザーの ID ソースを という名前のヘッダーに変更しますAuthorization。

```
aws apigatewayv2 update-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3 \  
  --identity-source '$request.header.Authorization'
```

出力:

```
{  
  "AuthorizerId": "a1b2c3",  
  "AuthorizerType": "JWT",  
  "IdentitySource": [  
    "$request.header.Authorization"  
  ],  
  "JwtConfiguration": {  
    "Audience": [  
      "123456abc"  
    ],  
    "Issuer": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123"  
  },  
  "Name": "my-jwt-authorizer"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[JWT「オーソライザー HTTPAPIsによる へのアクセスの制御」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAuthorizer](#)」の「」を参照してください。

update-deployment

次の例は、update-deployment を使用する方法を説明しています。

AWS CLI

デプロイの説明を変更するには

次のupdate-deployment例では、デプロイの説明を更新します。

```
aws apigatewayv2 update-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id abcdef \  
  --description 'Manual deployment to fix integration test failures.'
```

出力:

```
{  
  "AutoDeployed": false,  
  "CreateDate": "2020-02-05T16:21:48+00:00",  
  "DeploymentId": "abcdef",  
  "DeploymentStatus": "DEPLOYED",  
  "Description": "Manual deployment to fix integration test failures."  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の「[APIGateway HTTPAPIでの開発](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDeployment](#)」の「」を参照してください。

update-domain-name

次の例は、update-domain-name を使用する方法を説明しています。

AWS CLI

カスタムドメイン名を更新するには

次のupdate-domain-name例では、api.example.comカスタムドメイン名の新しいACM証明書を指定します。

```
aws apigatewayv2 update-domain-name \  
  --domain-name api.example.com \  
  --domain-name-configurations CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678
```

出力:

```
{
  "ApiMappingSelectionExpression": "$request.basepath",
  "DomainName": "regional.example.com",
  "DomainNameConfigurations": [
    {
      "ApiGatewayDomainName": "d-id.execute-api.us-west-2.amazonaws.com",
      "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/123456789012-1234-1234-1234-12345678",
      "EndpointType": "REGIONAL",
      "HostedZoneId": "123456789111",
      "SecurityPolicy": "TLS_1_2",
      "DomainNameStatus": "AVAILABLE"
    }
  ]
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[API「Gatewayでのリージョンカスタムドメイン名の設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateDomainName](#)」の「」を参照してください。

update-integration

次のコード例は、update-integration を使用する方法を示しています。

AWS CLI

Lambda 統合を更新するには

次のupdate-integration例では、既存の AWS Lambda 統合を更新して、指定された Lambda 関数を使用します。

```
aws apigatewayv2 update-integration \  
  --api-id a1b2c3d4 \  
  --integration-id a1b2c3 \  
  --integration-uri arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/functions/  
arn:aws:lambda:us-west-2:123456789012:function:my-new-function/invocations
```

出力:

```
{
```

```
"ConnectionType": "INTERNET",
"IntegrationId": "a1b2c3",
"IntegrationMethod": "POST",
"IntegrationType": "AWS_PROXY",
"IntegrationUri": "arn:aws:apigateway:us-west-2:lambda:path/2015-03-31/
functions/arn:aws:lambda:us-west-2:123456789012:function:my-new-function/
invocations",
"PayloadFormatVersion": "2.0",
"TimeoutInMillis": 5000
}
```

詳細については、[「Amazon Gateway デベロッパーガイド」の「の統合の設定HTTPAPIs」](#)および[「の統合の設定 WebSocket API」](#)を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIntegration](#)」の「」を参照してください。

update-route

次の例は、update-route を使用する方法を説明しています。

AWS CLI

例 1: ルートの統合を更新するには

次のupdate-route例では、指定されたルートの統合を更新します。

```
aws apigatewayv2 update-route \
  --api-id a1b2c3d4 \
  --route-id a1b2c3 \
  --target integrations/a1b2c6
```

出力:

```
{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",
  "RouteId": "a1b2c3",
  "RouteKey": "ANY /pets",
  "Target": "integrations/a1b2c6"
}
```

例 2: ルートにオーソライザーを追加するには

次のupdate-route例では、指定されたルートを更新してオーソライザーを使用します。

```
aws apigatewayv2 update-route \  
  --api-id a1b2c3d4 \  
  --route-id a1b2c3 \  
  --authorization-type JWT \  
  --authorizer-id a1b2c5 \  
  --authorization-scopes user.id user.email
```

出力:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationScopes": [  
    "user.id",  
    "user.email"  
  ],  
  "AuthorizationType": "JWT",  
  "AuthorizerId": "a1b2c5",  
  "OperationName": "GET HTTP",  
  "RequestParameters": {},  
  "RouteId": "a1b2c3",  
  "RouteKey": "GET /pets",  
  "Target": "integrations/a1b2c6"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の[JWT「オーソライザー HTTP APIsによる へのアクセスの制御」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRoute](#)」の「」を参照してください。

update-stage

次の例は、update-stage を使用する方法を説明しています。

AWS CLI

カスタムスロットリングを設定するには

次のupdate-stage例では、 の指定されたステージとルートにカスタムスロットリングを設定しますAPI。

```
aws apigatewayv2 update-stage \  
  --api-id a1b2c3d4 \  
  --stage-name dev \  
  --route-settings '{"GET /pets":  
{"ThrottlingBurstLimit":100,"ThrottlingRateLimit":2000}}'
```

出力:

```
{  
  "CreateDate": "2020-04-05T16:21:16+00:00",  
  "DefaultRouteSettings": {  
    "DetailedMetricsEnabled": false  
  },  
  "DeploymentId": "shktxb",  
  "LastUpdatedDate": "2020-04-08T22:23:17+00:00",  
  "RouteSettings": {  
    "GET /pets": {  
      "ThrottlingBurstLimit": 100,  
      "ThrottlingRateLimit": 2000.0  
    }  
  },  
  "StageName": "dev",  
  "StageVariables": {},  
  "Tags": {}  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド [HTTP](#)」の「[の保護API](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateStage](#)」の「」を参照してください。

update-vpc-link

次の例は、update-vpc-link を使用する方法を説明しています。

AWS CLI

VPC リンクを更新するには

次のupdate-vpc-link例では、VPCリンクの名前を更新します。VPCリンクを作成したら、そのセキュリティグループやサブネットを変更することはできません。

```
aws apigatewayv2 update-vpc-link \  
  --vpc-link-id abcd123 \  
  --name MyUpdatedVpcLink
```

出力:

```
{  
  "CreateDate": "2020-04-07T00:27:47Z",  
  "Name": "MyUpdatedVpcLink",  
  "SecurityGroupIds": [  
    "sg1234",  
    "sg5678"  
  ],  
  "SubnetIds": [  
    "subnet-aaaa",  
    "subnet-bbbb"  
  ],  
  "Tags": {},  
  "VpcLinkId": "abcd123",  
  "VpcLinkStatus": "AVAILABLE",  
  "VpcLinkStatusMessage": "VPC link is ready to route traffic",  
  "VpcLinkVersion": "V2"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド[HTTP](#)」の「[のVPCリンクの使用 APIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVpcLink](#)」の「」を参照してください。

API を使用したゲートウェイ管理APIの例 AWS CLI

次のコード例は、APIGateway Management AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますAPI。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-connection

次の例は、delete-connection を使用する方法を説明しています。

AWS CLI

WebSocket 接続を削除するには

次のdelete-connection例では、クライアントを指定された から切断します WebSocket API。

```
aws apigatewaymanagementapi delete-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド」の「[バックエンドサービスで @connections コマンドを使用する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConnection](#)」の「」を参照してください。

get-connection

次のコード例は、get-connection を使用する方法を示しています。

AWS CLI

WebSocket 接続に関する情報を取得するには

次のget-connection例では、指定された への接続について説明します WebSocket API。

```
aws apigatewaymanagementapi get-connection \  
--connection-id L0SM9c0FvHcCIhw= \  
--endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

出力:

```
{  
  "ConnectedAt": "2020-04-30T20:10:33.236Z",  
  "Identity": {  
    "SourceIp": "192.0.2.1"  
  },  
  "LastActiveAt": "2020-04-30T20:10:42.997Z"  
}
```

詳細については、「Amazon API Gateway デベロッパーガイド」の「[バックエンドサービスで @connections コマンドを使用する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConnection](#)」の「」を参照してください。

post-to-connection

次の例は、post-to-connection を使用する方法を説明しています。

AWS CLI

WebSocket 接続にデータを送信するには

次のpost-to-connection例では、指定された に接続されているクライアントにメッセージを送信します WebSocket API。

```
aws apigatewaymanagementapi post-to-connection \  
--connection-id L0SM9c0FvHcCIhw= \  
--data "Hello from API Gateway!" \  
--endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

このコマンドでは何も出力されません。

詳細については、「Amazon API Gateway デベロッパーガイド」の「[バックエンドサービスで @connections コマンドを使用する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PostToConnection](#)」の「」を参照してください。

を使用した App Mesh の例 AWS CLI

次のコード例は、App Mesh AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-mesh

次のコード例は、create-mesh を使用する方法を示しています。

AWS CLI

例 1: 新しいサービスマッシュを作成するには

次のcreate-mesh例では、サービスマッシュを作成します。

```
aws appmesh create-mesh \  
  --mesh-name app1
```

出力:

```
{  
  "mesh": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
```

```
        "createdAt":1563809909.282,  
        "lastUpdatedAt":1563809909.282,  
        "uid":"a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
        "version":1  
    },  
    "spec":{},  
    "status":{  
        "status":"ACTIVE"  
    }  
}  
}
```

例 2: 複数のタグで新しいサービスメッシュを作成するには

次のcreate-mesh例では、複数のタグを持つサービスメッシュを作成します。

```
aws appmesh create-mesh \  
  --mesh-name app2 \  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

出力:

```
{  
  "mesh":{  
    "meshName":"app2",  
    "metadata":{  
      "arn":"arn:aws:appmesh:us-east-1:123456789012:mesh/app2",  
      "createdAt":1563822121.877,  
      "lastUpdatedAt":1563822121.877,  
      "uid":"a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version":1  
    },  
    "spec":{},  
    "status":{  
      "status":"ACTIVE"  
    }  
  }  
}
```

詳細については、App [Mesh ユーザーガイド](#)の「[Service AWS Meshes](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateMesh](#)」の「」を参照してください。

create-route

次のコード例は、create-route を使用する方法を示しています。

AWS CLI

新しい gRPC ルートを作成するには

次のcreate-route例では、JSON入力ファイルを使用して gRPC ルートを作成します。gRPC 123 で始まるメタデータを持つトラフィックは、という名前の仮想ノードにルーティングされず serviceBgrpc。ルートのターゲットと通信しようとするときに特定の gRPC、HTTP、またはのTCP障害が発生した場合、ルートは 3 回再試行されます。再試行するたびに 15 秒の遅延が発生します。

```
aws appmesh create-route \  
  --cli-input-json file://create-route-grpc.json
```

create-route-grpc.json の内容:

```
{  
  "meshName" : "apps",  
  "routeName" : "grpcRoute",  
  "spec" : {  
    "grpcRoute" : {  
      "action" : {  
        "weightedTargets" : [  
          {  
            "virtualNode" : "serviceBgrpc",  
            "weight" : 100  
          }  
        ]  
      },  
      "match" : {  
        "metadata" : [  
          {  
            "invert" : false,  
            "match" : {  
              "prefix" : "123"  
            },  
            "name" : "myMetadata"  
          }  
        ],  
      }  
    }  
  }  
}
```

```

    "methodName" : "GetColor",
    "serviceName" : "com.amazonaws.services.ColorService"
  },
  "retryPolicy" : {
    "grpcRetryEvents" : [ "deadline-exceeded" ],
    "httpRetryEvents" : [ "server-error", "gateway-error" ],
    "maxRetries" : 3,
    "perRetryTimeout" : {
      "unit" : "s",
      "value" : 15
    },
    "tcpRetryEvents" : [ "connection-error" ]
  },
  "priority" : 100
},
"virtualRouterName" : "serviceBgrpc"
}

```

出力:

```

{
  "route": {
    "meshName": "apps",
    "metadata": {
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/serviceBgrpc/route/grpcRoute",
      "createdAt": 1572010806.008,
      "lastUpdatedAt": 1572010806.008,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "routeName": "grpcRoute",
    "spec": {
      "grpcRoute": {
        "action": {
          "weightedTargets": [
            {
              "virtualNode": "serviceBgrpc",
              "weight": 100
            }
          ]
        }
      }
    }
  },
}

```

```
    "match": {
      "metadata": [
        {
          "invert": false,
          "match": {
            "prefix": "123"
          },
          "name": "mymetadata"
        }
      ],
      "methodName": "GetColor",
      "serviceName": "com.amazonaws.services.ColorService"
    },
    "retryPolicy": {
      "grpcRetryEvents": [
        "deadline-exceeded"
      ],
      "httpRetryEvents": [
        "server-error",
        "gateway-error"
      ],
      "maxRetries": 3,
      "perRetryTimeout": {
        "unit": "s",
        "value": 15
      },
      "tcpRetryEvents": [
        "connection-error"
      ]
    }
  },
  "priority": 100
},
"status": {
  "status": "ACTIVE"
},
"virtualRouterName": "serviceBgrpc"
}
}
```

新しいルートHTTPまたは HTTP/2 ルートを作成するには


```

        "gateway-error"
      ],
      "maxRetries": 3,
      "perRetryTimeout": {
        "unit": "s",
        "value": 15
      },
      "tcpRetryEvents": [
        "connection-error"
      ]
    }
  },
  "priority": 200
},
"virtualRouterName": "serviceBhttp2"
}

```

出力:

```

{
  "route": {
    "meshName": "apps",
    "metadata": {
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/
serviceBhttp2/route/http2Route",
      "createdAt": 1572011008.352,
      "lastUpdatedAt": 1572011008.352,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "routeName": "http2Route",
    "spec": {
      "http2Route": {
        "action": {
          "weightedTargets": [
            {
              "virtualNode": "serviceBhttp2",
              "weight": 100
            }
          ]
        },
        "match": {
          "headers": [

```

```
        {
            "invert": false,
            "match": {
                "prefix": "123"
            },
            "name": "clientRequestId"
        }
    ],
    "method": "POST",
    "prefix": "/",
    "scheme": "http"
},
"retryPolicy": {
    "httpRetryEvents": [
        "server-error",
        "gateway-error"
    ],
    "maxRetries": 3,
    "perRetryTimeout": {
        "unit": "s",
        "value": 15
    },
    "tcpRetryEvents": [
        "connection-error"
    ]
}
},
"priority": 200
},
"status": {
    "status": "ACTIVE"
},
"virtualRouterName": "serviceBhttp2"
}
}
```

新しいTCPルートを作成するには

次のcreate-route例では、JSON入力ファイルを使用してTCPルートを作成します。トラフィックの75%はという名前の仮想ノードにルーティングされserviceBtcp、トラフィックの25%はserviceBv2tcpという名前の仮想ノードにルーティングされます。ターゲットごとに異なる重みを指定すると、アプリケーションの新しいバージョンのデプロイを効果的に行うことがで

きます。重みを調整して、最終的にすべてのトラフィックの 100% がアプリケーションの新しいバージョンを持つターゲットにルーティングされるようになります。

```
aws appmesh create-route \  
--cli-input-json file://create-route-tcp.json
```

create-route-tcp.json の内容 :

```
{  
  "meshName": "apps",  
  "routeName": "tcpRoute",  
  "spec": {  
    "priority": 300,  
    "tcpRoute": {  
      "action": {  
        "weightedTargets": [  
          {  
            "virtualNode": "serviceBtcp",  
            "weight": 75  
          },  
          {  
            "virtualNode": "serviceBv2tcp",  
            "weight": 25  
          }  
        ]  
      }  
    }  
  },  
  "virtualRouterName": "serviceBtcp"  
}
```

出力:

```
{  
  "route": {  
    "meshName": "apps",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/apps/virtualRouter/  
serviceBtcp/route/tcpRoute",  
      "createdAt": 1572011436.26,  
      "lastUpdatedAt": 1572011436.26,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    }  
  }  
}
```

```
    "version": 1
  },
  "routeName": "tcpRoute",
  "spec": {
    "priority": 300,
    "tcpRoute": {
      "action": {
        "weightedTargets": [
          {
            "virtualNode": "serviceBtcp",
            "weight": 75
          },
          {
            "virtualNode": "serviceBv2tcp",
            "weight": 25
          }
        ]
      }
    }
  },
  "status": {
    "status": "ACTIVE"
  },
  "virtualRouterName": "serviceBtcp"
}
}
```

詳細については、AWS「App Mesh ユーザーガイド」の「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateRoute](#)」の「」を参照してください。

create-virtual-gateway

次のコード例は、create-virtual-gateway を使用する方法を示しています。

AWS CLI

新しい仮想ゲートウェイを作成するには

次のcreate-virtual-gateway例では、JSON入力ファイルを使用して、ポート 9080 HTTPを使用するリスナーを持つ仮想ゲートウェイを作成します。

```
aws appmesh create-virtual-gateway \  
  --mesh-name meshName \  
  --virtual-gateway-name virtualGatewayName \  
  --cli-input-json file://create-virtual-gateway.json
```

create-virtual-gateway.json の内容:

```
{  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 9080,  
          "protocol": "http"  
        }  
      }  
    ]  
  }  
}
```

出力:

```
{  
  "virtualGateway": {  
    "meshName": "meshName",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-west-2:123456789012:mesh/meshName/  
virtualGateway/virtualGatewayName",  
      "createdAt": "2022-04-06T10:42:42.015000-05:00",  
      "lastUpdatedAt": "2022-04-06T10:42:42.015000-05:00",  
      "meshOwner": "123456789012",  
      "resourceOwner": "123456789012",  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    },  
    "spec": {  
      "listeners": [  
        {  
          "portMapping": {  
            "port": 9080,  
            "protocol": "http"  
          }  
        }  
      ]  
    }  
  }  
}
```

```
    }
  ]
},
"status": {
  "status": "ACTIVE"
},
"virtualGatewayName": "virtualGatewayName"
}
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVirtualGateway](#)」の「」を参照してください。

create-virtual-node

次のコード例は、create-virtual-node を使用する方法を示しています。

AWS CLI

例 1: 検出DNSに を使用する新しい仮想ノードを作成するには

次のcreate-virtual-node例では、JSON入力ファイルを使用して、 がサービス検出DNSに使用する仮想ノードを作成します。

```
aws appmesh create-virtual-node \  
  --cli-input-json file://create-virtual-node-dns.json
```

create-virtual-node-dns.json の内容:

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ],  
  },  
}
```

```
    "serviceDiscovery": {
      "dns": {
        "hostname": "serviceBv1.svc.cluster.local"
      }
    },
    "virtualNodeName": "vnServiceBv1"
  }
}
```

出力:

```
{
  "virtualNode": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/vnServiceBv1",
      "createdAt": 1563810019.874,
      "lastUpdatedAt": 1563810019.874,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ],
      "serviceDiscovery": {
        "dns": {
          "hostname": "serviceBv1.svc.cluster.local"
        }
      }
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualNodeName": "vnServiceBv1"
  }
}
```


例 2: 検出に AWS Cloud Map を使用する新しい仮想ノードを作成するには

次の `create-virtual-node` 例では、JSON 入力ファイルを使用して、サービス検出に AWS Cloud Map を使用する仮想ノードを作成します。

```
aws appmesh create-virtual-node \  
  --cli-input-json file://create-virtual-node-cloud-map.json
```

`create-virtual-node-cloud-map.json` の内容:

```
{  
  "meshName": "app1",  
  "spec": {  
    "backends": [  
      {  
        "virtualService": {  
          "virtualServiceName": "serviceA.svc.cluster.local"  
        }  
      }  
    ],  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ],  
    "serviceDiscovery": {  
      "awsCloudMap": {  
        "attributes": [  
          {  
            "key": "Environment",  
            "value": "Testing"  
          }  
        ],  
        "namespaceName": "namespace1",  
        "serviceName": "serviceA"  
      }  
    }  
  },  
  "virtualNodeName": "vnServiceA"  
}
```

```
}
```

出力:

```
{
  "virtualNode": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/vnServiceA",
      "createdAt": 1563810859.465,
      "lastUpdatedAt": 1563810859.465,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {
      "backends": [
        {
          "virtualService": {
            "virtualServiceName": "serviceA.svc.cluster.local"
          }
        }
      ],
      "listeners": [
        {
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ],
      "serviceDiscovery": {
        "awsCloudMap": {
          "attributes": [
            {
              "key": "Environment",
              "value": "Testing"
            }
          ],
          "namespaceName": "namespace1",
          "serviceName": "serviceA"
        }
      }
    }
  }
}
```

```
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualNodeName": "vnServiceA"
  }
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ノード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVirtualNode](#)」の「」を参照してください。

create-virtual-router

次のコード例は、create-virtual-router を使用する方法を示しています。

AWS CLI

新しい仮想ルーターを作成するには

次のcreate-virtual-router例では、JSON入力ファイルを使用して、ポート 80 HTTPを使用するリスナーを持つ仮想ルーターを作成します。

```
aws appmesh create-virtual-router \
  --cli-input-json file://create-virtual-router.json
```

create-virtual-router.json の内容:

```
{
  "meshName": "app1",
  "spec": {
    "listeners": [
      {
        "portMapping": {
          "port": 80,
          "protocol": "http"
        }
      }
    ]
  },
  "virtualRouterName": "vrServiceB"
}
```

```
}
```

出力:

```
{
  "virtualRouter": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
      "createdAt": 1563810546.59,
      "lastUpdatedAt": 1563810546.59,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ]
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualRouterName": "vrServiceB"
  }
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ルーター](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVirtualRouter](#)」の「」を参照してください。

create-virtual-service

次のコード例は、create-virtual-service を使用する方法を示しています。

AWS CLI

例 1: 仮想ノードプロバイダーを使用して新しい仮想サービスを作成するには

次のcreate-virtual-service例では、JSON入力ファイルを使用して、仮想ノードプロバイダーを使用して仮想サービスを作成します。

```
aws appmesh create-virtual-service \  
  --cli-input-json file://create-virtual-service-virtual-node.json
```

create-virtual-service-virtual-node.json の内容:

```
{  
  "meshName": "app1",  
  "spec": {  
    "provider": {  
      "virtualNode": {  
        "virtualNodeName": "vnServiceA"  
      }  
    }  
  },  
  "virtualServiceName": "serviceA.svc.cluster.local"  
}
```

出力:

```
{  
  "virtualService": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/  
serviceA.svc.cluster.local",  
      "createdAt": 1563810859.474,  
      "lastUpdatedAt": 1563810967.179,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 2  
    },  
    "spec": {  
      "provider": {  
        "virtualNode": {  
          "virtualNodeName": "vnServiceA"  
        }  
      }  
    }  
  }  
}
```

```

    }
  },
  "status": {
    "status": "ACTIVE"
  },
  "virtualServiceName": "serviceA.svc.cluster.local"
}
}

```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ノード](#)」を参照してください。

例 2: 仮想ルータープロバイダーを使用して新しい仮想サービスを作成するには

次のcreate-virtual-service例では、JSON入力ファイルを使用して、仮想ルータープロバイダーを使用して仮想サービスを作成します。

```

aws appmesh create-virtual-service \
  --cli-input-json file://create-virtual-service-virtual-router.json

```

create-virtual-service-virtual-router.json の内容:

```

{
  "meshName": "app1",
  "spec": {
    "provider": {
      "virtualRouter": {
        "virtualRouterName": "vrServiceB"
      }
    }
  },
  "virtualServiceName": "serviceB.svc.cluster.local"
}

```

出力:

```

{
  "virtualService": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/serviceB.svc.cluster.local",
      "createdAt": 1563908363.999,

```

```
        "lastUpdatedAt": 1563908363.999,
        "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "version": 1
    },
    "spec": {
        "provider": {
            "virtualRouter": {
                "virtualRouterName": "vrServiceB"
            }
        }
    },
    "status": {
        "status": "ACTIVE"
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
}
}
```

詳細については、AWS App Mesh ユーザーガイドの「Virtual Services<https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual_services.html>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateVirtualService](#)」の「」を参照してください。

delete-mesh

次の例は、delete-mesh を使用する方法を説明しています。

AWS CLI

サービスメッシュを削除するには

次のdelete-mesh例では、指定されたサービスメッシュを削除します。

```
aws appmesh delete-mesh \
  --mesh-name app1
```

出力:

```
{
  "mesh": {
    "meshName": "app1",
```

```
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
      "createdAt": 1563809909.282,
      "lastUpdatedAt": 1563824981.248,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 2
    },
    "spec": {
      "egressFilter": {
        "type": "ALLOW_ALL"
      }
    },
    "status": {
      "status": "DELETED"
    }
  }
}
```

詳細については、App [Mesh ユーザーガイドの「Service AWS Meshes」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteMesh](#)」の「」を参照してください。

delete-route

次の例は、delete-route を使用する方法を説明しています。

AWS CLI

ルートを削除するには

次のdelete-route例では、指定されたルートを削除します。

```
aws appmesh delete-route \
  --mesh-name app1 \
  --virtual-router-name vrServiceB \
  --route-name toVnServiceB-weighted
```

出力:

```
{
  "route": {
    "meshName": "app1",
```



```
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB-weighted",
      "createdAt": 1563811384.015,
      "lastUpdatedAt": 1563823915.936,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 3
    },
    "routeName": "toVnServiceB-weighted",
    "spec": {
      "httpRoute": {
        "action": {
          "weightedTargets": [
            {
              "virtualNode": "vnServiceBv1",
              "weight": 80
            },
            {
              "virtualNode": "vnServiceBv2",
              "weight": 20
            }
          ]
        },
        "match": {
          "prefix": "/"
        }
      }
    },
    "status": {
      "status": "DELETED"
    },
    "virtualRouterName": "vrServiceB"
  }
}
```

詳細については、AWS「App Mesh ユーザーガイド」の「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteRoute](#)」の「」を参照してください。

delete-virtual-node

次の例は、delete-virtual-node を使用する方法を説明しています。

AWS CLI

仮想ノードを削除するには

次のdelete-virtual-node例では、指定された仮想ノードを削除します。

```
aws appmesh delete-virtual-node \  
  --mesh-name app1 \  
  --virtual-node-name vnServiceBv2
```

出力:

```
{  
  "virtualNode": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/  
vnServiceBv2",  
      "createdAt": 1563810117.297,  
      "lastUpdatedAt": 1563824700.678,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 2  
    },  
    "spec": {  
      "backends": [],  
      "listeners": [  
        {  
          "portMapping": {  
            "port": 80,  
            "protocol": "http"  
          }  
        }  
      ],  
      "serviceDiscovery": {  
        "dns": {  
          "hostname": "serviceBv2.svc.cluster.local"  
        }  
      }  
    },  
    "status": {  
      "status": "DELETED"  
    },  
    "virtualNodeName": "vnServiceBv2"  
  }  
}
```

```
}  
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ノード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVirtualNode](#)」の「」を参照してください。

delete-virtual-router

次の例は、delete-virtual-router を使用する方法を説明しています。

AWS CLI

仮想ルーターを削除するには

次のdelete-virtual-router例では、指定された仮想ルーターを削除します。

```
aws appmesh delete-virtual-router \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB
```

出力:

```
{  
  "virtualRouter": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB",  
      "createdAt": 1563810546.59,  
      "lastUpdatedAt": 1563824253.467,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 3  
    },  
    "spec": {  
      "listeners": [  
        {  
          "portMapping": {  
            "port": 80,  
            "protocol": "http"  
          }  
        }  
      ]  
    }  
  }  
}
```

```
    }
  ]
},
"status": {
  "status": "DELETED"
},
"virtualRouterName": "vrServiceB"
}
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ルーター](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVirtualRouter](#)」の「」を参照してください。

delete-virtual-service

次のコード例は、delete-virtual-service を使用する方法を示しています。

AWS CLI

仮想サービスを削除するには

次のdelete-virtual-service例では、指定された仮想サービスを削除します。

```
aws appmesh delete-virtual-service \
  --mesh-name app1 \
  --virtual-service-name serviceB.svc.cluster.local
```

出力:

```
{
  "virtualService": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/
serviceB.svc.cluster.local",
      "createdAt": 1563908363.999,
      "lastUpdatedAt": 1563913940.866,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 3
    },
  },
}
```

```
    "spec": {},
    "status": {
      "status": "DELETED"
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
  }
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想サービス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVirtualService](#)」の「」を参照してください。

describe-mesh

次の例は、describe-mesh を使用する方法を説明しています。

AWS CLI

サービスメッシュを記述するには

次のdescribe-mesh例では、指定されたサービスメッシュに関する詳細を返します。

```
aws appmesh describe-mesh \
  --mesh-name app1
```

出力:

```
{
  "mesh": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
      "createdAt": 1563809909.282,
      "lastUpdatedAt": 1563809909.282,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {},
    "status": {
      "status": "ACTIVE"
    }
  }
}
```

```
}
```

詳細については、App [Mesh ユーザーガイド](#)の「[Service AWS Meshes](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeMesh](#)」の「」を参照してください。

describe-route

次の例は、describe-route を使用する方法を説明しています。

AWS CLI

ルートを記述するには

次のdescribe-route例では、指定されたルートに関する詳細を返します。

```
aws appmesh describe-route \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB \  
  --route-name toVnServiceB-weighted
```

出力:

```
{  
  "route": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB/route/toVnServiceB-weighted",  
      "createdAt": 1563811384.015,  
      "lastUpdatedAt": 1563811384.015,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    },  
    "routeName": "toVnServiceB-weighted",  
    "spec": {  
      "httpRoute": {  
        "action": {  
          "weightedTargets": [  
            {  
              "virtualNode": "vnServiceBv1",  
              "weight": 90  
            }  
          ]  
        }  
      }  
    }  
  }  
}
```

```
        },
        {
            "virtualNode": "vnServiceBv2",
            "weight": 10
        }
    ]
},
"match": {
    "prefix": "/"
}
}
},
"status": {
    "status": "ACTIVE"
},
"virtualRouterName": "vrServiceB"
}
}
```

詳細については、AWS「App Mesh ユーザーガイド」の「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeRoute](#)」の「」を参照してください。

describe-virtual-node

次の例は、describe-virtual-node を使用する方法を説明しています。

AWS CLI

仮想ノードを記述するには

次のdescribe-virtual-node例では、指定された仮想ノードに関する詳細を返します。

```
aws appmesh describe-virtual-node \
  --mesh-name app1 \
  --virtual-node-name vnServiceBv1
```

出力:

```
{
  "virtualNode": {
    "meshName": "app1",
```

```
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv1",
      "createdAt": 1563810019.874,
      "lastUpdatedAt": 1563810019.874,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 1
    },
    "spec": {
      "backends": [],
      "listeners": [
        {
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ],
      "serviceDiscovery": {
        "dns": {
          "hostname": "serviceBv1.svc.cluster.local"
        }
      }
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualNodeName": "vnServiceBv1"
  }
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ノード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVirtualNode](#)」の「」を参照してください。

describe-virtual-router

次のコード例は、describe-virtual-router を使用する方法を示しています。

AWS CLI

仮想ルーターを記述するには

次のdescribe-virtual-router例では、指定された仮想ルーターの詳細を返します。

```
aws appmesh describe-virtual-router \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB
```

出力:

```
{  
  "virtualRouter": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB",  
      "createdAt": 1563810546.59,  
      "lastUpdatedAt": 1563810546.59,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    },  
    "spec": {  
      "listeners": [  
        {  
          "portMapping": {  
            "port": 80,  
            "protocol": "http"  
          }  
        }  
      ]  
    },  
    "status": {  
      "status": "ACTIVE"  
    },  
    "virtualRouterName": "vrServiceB"  
  }  
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ルーター](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVirtualRouter](#)」の「」を参照してください。

describe-virtual-service

次の例は、describe-virtual-service を使用する方法を説明しています。

AWS CLI

仮想サービスを記述するには

次のdescribe-virtual-service例では、指定された仮想サービスの詳細を返します。

```
aws appmesh describe-virtual-service \  
  --mesh-name app1 \  
  --virtual-service-name serviceB.svc.cluster.local
```

出力:

```
{  
  "virtualService": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/  
serviceB.svc.cluster.local",  
      "createdAt": 1563908363.999,  
      "lastUpdatedAt": 1563908363.999,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 1  
    },  
    "spec": {  
      "provider": {  
        "virtualRouter": {  
          "virtualRouterName": "vrServiceB"  
        }  
      }  
    },  
    "status": {  
      "status": "ACTIVE"  
    },  
    "virtualServiceName": "serviceB.svc.cluster.local"  
  }  
}
```

詳細については、「App Mesh ユーザーガイド」の [「仮想サービス」](#) を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVirtualService](#)」の「」を参照してください。

list-meshes

次のコード例は、list-meshes を使用する方法を示しています。

AWS CLI

サービスマッシュを一覧表示するには

次のlist-meshes例では、現在の AWS リージョンのすべてのサービスマッシュを一覧表示します。

```
aws appmesh list-meshes
```

出力:

```
{
  "meshes": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
      "meshName": "app1"
    }
  ]
}
```

詳細については、App [Mesh ユーザーガイド](#)の「[Service AWS Meshes](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMeshes](#)」の「」を参照してください。

list-routes

次のコード例は、list-routes を使用する方法を示しています。

AWS CLI

ルートを一覧表示するには

次のlist-routes例では、指定された仮想ルーターのすべてのルートを一覧表示します。

```
aws appmesh list-routes \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB
```

出力:

```
{  
  "routes": [  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB/route/toVnServiceB",  
      "meshName": "app1",  
      "routeName": "toVnServiceB-weighted",  
      "virtualRouterName": "vrServiceB"  
    }  
  ]  
}
```

詳細については、AWS「App Mesh ユーザーガイド」の「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListRoutes](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたリソースに割り当てられたすべてのタグを一覧表示します。

```
aws appmesh list-tags-for-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1
```

出力:

```
{
```

```
"tags": [
  {
    "key": "key1",
    "value": "value1"
  },
  {
    "key": "key2",
    "value": "value2"
  },
  {
    "key": "key3",
    "value": "value3"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-virtual-nodes

次のコード例は、list-virtual-nodes を使用する方法を示しています。

AWS CLI

仮想ノードを一覧表示するには

次のlist-virtual-nodes例では、指定されたサービスマッシュ内のすべての仮想ノードを一覧表示します。

```
aws appmesh list-virtual-nodes \
  --mesh-name app1
```

出力:

```
{
  "virtualNodes": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/vnServiceBv1",
      "meshName": "app1",
      "virtualNodeName": "vnServiceBv1"
    }
  ]
}
```

```
    },
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/
vnServiceBv2",
      "meshName": "app1",
      "virtualNodeName": "vnServiceBv2"
    }
  ]
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ノード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVirtualNodes](#)」の「」を参照してください。

list-virtual-routers

次の例は、list-virtual-routers を使用する方法を説明しています。

AWS CLI

仮想ルーターを一覧表示するには

次のlist-virtual-routers例では、指定されたサービスメッシュ内のすべての仮想ルーターを一覧表示します。

```
aws appmesh list-virtual-routers \
  --mesh-name app1
```

出力:

```
{
  "virtualRouters": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
      "meshName": "app1",
      "virtualRouterName": "vrServiceB"
    }
  ]
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ルーター](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVirtualRouters](#)」の「」を参照してください。

list-virtual-services

次のコード例は、list-virtual-services を使用する方法を示しています。

AWS CLI

仮想サービスを一覧表示するには

次のlist-virtual-services例では、指定されたサービスメッシュ内のすべての仮想サービスを一覧表示します。

```
aws appmesh list-virtual-services \  
  --mesh-name app1
```

出力:

```
{  
  "virtualServices": [  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/  
serviceA.svc.cluster.local",  
      "meshName": "app1",  
      "virtualServiceName": "serviceA.svc.cluster.local"  
    },  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/  
serviceB.svc.cluster.local",  
      "meshName": "app1",  
      "virtualServiceName": "serviceB.svc.cluster.local"  
    }  
  ]  
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想サービス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVirtualServices](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、指定されたリソースvalue1に 値key1を持つ タグを追加します。

```
aws appmesh tag-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \  
  --tags key=key1,value=value1
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを解除するには

次のuntag-resource例では、指定されたリソースkey1から キーを持つタグを削除します。

```
aws appmesh untag-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \  
  --tag-keys key1
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-mesh

次の例は、update-mesh を使用する方法を説明しています。

AWS CLI

サービスメッシュを更新するには

次のupdate-mesh例では、JSON入力ファイルを使用してサービスメッシュを更新し、すべての外部エグレストラフィックを Envoy プロキシを介してそのまま転送できるようにします。

```
aws appmesh update-mesh \  
  --cli-input-json file://update-mesh.json
```

update-mesh.json の内容:

```
{  
  "meshName": "app1",  
  "spec": {  
    "egressFilter": {  
      "type": "ALLOW_ALL"  
    }  
  }  
}
```

出力:

```
{  
  "mesh": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",  
      "createdAt": 1563809909.282,  
      "lastUpdatedAt": 1563812829.687,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 2  
    },  
    "spec": {  
      "egressFilter": {  
        "type": "ALLOW_ALL"  
      }  
    },  
    "status": {  
      "status": "ACTIVE"  
    }  
  }  
}
```

```
}
```

詳細については、App [Mesh ユーザーガイドの「Service AWS Meshes」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMesh](#)」の「」を参照してください。

update-route

次のコード例は、update-route を使用する方法を示しています。

AWS CLI

ルートを更新するには

次のupdate-route例では、JSON入力ファイルを使用してルートの重みを更新します。

```
aws appmesh update-route \  
  --cli-input-json file://update-route-weighted.json
```

update-route-weighted.json の内容:

```
{  
  "meshName": "app1",  
  "routeName": "toVnServiceB-weighted",  
  "spec": {  
    "httpRoute": {  
      "action": {  
        "weightedTargets": [  
          {  
            "virtualNode": "vnServiceBv1",  
            "weight": 80  
          },  
          {  
            "virtualNode": "vnServiceBv2",  
            "weight": 20  
          }  
        ]  
      },  
      "match": {  
        "prefix": "/"  
      }  
    }  
  }  
}
```

```
  },  
  "virtualRouterName": "vrServiceB"  
}
```

出力:

```
{  
  "route": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB/route/toVnServiceB-weighted",  
      "createdAt": 1563811384.015,  
      "lastUpdatedAt": 1563819600.022,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 2  
    },  
    "routeName": "toVnServiceB-weighted",  
    "spec": {  
      "httpRoute": {  
        "action": {  
          "weightedTargets": [  
            {  
              "virtualNode": "vnServiceBv1",  
              "weight": 80  
            },  
            {  
              "virtualNode": "vnServiceBv2",  
              "weight": 20  
            }  
          ]  
        },  
        "match": {  
          "prefix": "/"  
        }  
      }  
    },  
    "status": {  
      "status": "ACTIVE"  
    },  
    "virtualRouterName": "vrServiceB"  
  }  
}
```

詳細については、AWS 「App Mesh ユーザーガイド」の「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRoute](#)」の「」を参照してください。

update-virtual-node

次の例は、update-virtual-node を使用する方法を説明しています。

AWS CLI

仮想ノードを更新するには

次のupdate-virtual-node例では、JSON入力ファイルを使用して、ヘルスチェックを仮想ノードに追加します。

```
aws appmesh update-virtual-node \  
  --cli-input-json file://update-virtual-node.json
```

update-virtual-node.json の内容:

```
{  
  "clientToken": "500",  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "healthCheck": {  
          "healthyThreshold": 5,  
          "intervalMillis": 10000,  
          "path": "/",  
          "port": 80,  
          "protocol": "http",  
          "timeoutMillis": 3000,  
          "unhealthyThreshold": 3  
        },  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ],  
  }  
}
```

```
    "serviceDiscovery": {
      "dns": {
        "hostname": "serviceBv1.svc.cluster.local"
      }
    },
    "virtualNodeName": "vnServiceBv1"
  }
}
```

出力:

```
{
  "virtualNode": {
    "meshName": "app1",
    "metadata": {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualNode/vnServiceBv1",
      "createdAt": 1563810019.874,
      "lastUpdatedAt": 1563819234.825,
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "version": 2
    },
    "spec": {
      "listeners": [
        {
          "healthCheck": {
            "healthyThreshold": 5,
            "intervalMillis": 10000,
            "path": "/",
            "port": 80,
            "protocol": "http",
            "timeoutMillis": 3000,
            "unhealthyThreshold": 3
          },
          "portMapping": {
            "port": 80,
            "protocol": "http"
          }
        }
      ],
      "serviceDiscovery": {
        "dns": {
          "hostname": "serviceBv1.svc.cluster.local"
        }
      }
    }
  }
}
```

```
    }
  },
  "status": {
    "status": "ACTIVE"
  },
  "virtualNodeName": "vnServiceBv1"
}
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ノード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVirtualNode](#)」の「」を参照してください。

update-virtual-router

次の例は、update-virtual-router を使用する方法を説明しています。

AWS CLI

仮想ルーターを更新するには

次のupdate-virtual-router例では、JSON入力ファイルを使用して仮想ルーターリスナーポートを更新します。

```
aws appmesh update-virtual-router \
  --cli-input-json file://update-virtual-router.json
```

update-virtual-router.json の内容:

```
{
  "meshName": "app1",
  "spec": {
    "listeners": [
      {
        "portMapping": {
          "port": 8080,
          "protocol": "http"
        }
      }
    ]
  }
}
```

```
    },  
    "virtualRouterName": "vrServiceB"  
  }  
}
```

出力:

```
{  
  "virtualRouter": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB",  
      "createdAt": 1563810546.59,  
      "lastUpdatedAt": 1563819431.352,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 2  
    },  
    "spec": {  
      "listeners": [  
        {  
          "portMapping": {  
            "port": 8080,  
            "protocol": "http"  
          }  
        }  
      ]  
    },  
    "status": {  
      "status": "ACTIVE"  
    },  
    "virtualRouterName": "vrServiceB"  
  }  
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想ルーター](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVirtualRouter](#)」の「」を参照してください。

update-virtual-service

次の例は、update-virtual-service を使用する方法を説明しています。

AWS CLI

仮想サービスを更新するには

次のupdate-virtual-service例では、JSON入力ファイルを使用して仮想ルータープロバイダーを使用するように仮想サービスを更新します。

```
aws appmesh update-virtual-service \  
  --cli-input-json file://update-virtual-service.json
```

update-virtual-service.json の内容:

```
{  
  "meshName": "app1",  
  "spec": {  
    "provider": {  
      "virtualRouter": {  
        "virtualRouterName": "vrServiceA"  
      }  
    }  
  },  
  "virtualServiceName": "serviceA.svc.cluster.local"  
}
```

出力:

```
{  
  "virtualService": {  
    "meshName": "app1",  
    "metadata": {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualService/  
serviceA.svc.cluster.local",  
      "createdAt": 1563810859.474,  
      "lastUpdatedAt": 1563820257.411,  
      "uid": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "version": 3  
    },  
    "spec": {  
      "provider": {  
        "virtualRouter": {  
          "virtualRouterName": "vrServiceA"  
        }  
      }  
    }  
  }  
}
```



```
    },
    "status": {
      "status": "ACTIVE"
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
  }
}
```

詳細については、AWS App Mesh ユーザーガイドの「[仮想サービス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVirtualService](#)」の「」を参照してください。

を使用した App Runner の例 AWS CLI

次のコード例は、App Runner AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-custom-domain

次の例は、associate-custom-domain を使用する方法を説明しています。

AWS CLI

ドメイン名と www サブドメインをサービスに関連付けるには

次のassociate-custom-domain例では、ユーザーが制御するカスタムドメイン名を App Runner サービスに関連付けます。ドメイン名はルートドメインでexample.com、大文字と小文字のサブドメインが含まれますwww.example.com。

```
aws apprunner associate-custom-domain \  
--cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "DomainName": "example.com",  
  "EnableWWWSubdomain": true  
}
```

出力:

```
{  
  "CustomDomain": {  
    "CertificateValidationRecords": [  
      {  
        "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",  
        "Status": "PENDING_VALIDATION",  
        "Type": "CNAME",  
        "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-  
validations.aws."  
      },  
      {  
        "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",  
        "Status": "PENDING_VALIDATION",  
        "Type": "CNAME",  
        "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-  
validations.aws."  
      }  
    ],  
    "DomainName": "example.com",  
    "EnableWWWSubdomain": true,  
    "Status": "CREATING"  
  },  
  "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateCustomDomain](#)」の「」を参照してください。

create-auto-scaling-configuration

次のコード例は、create-auto-scaling-configuration を使用する方法を示しています。

AWS CLI

高可用性自動スケーリング設定を作成するには

次のcreate-auto-scaling-configuration例では、を 5 MinSizeに設定することで高可用性に最適化された自動スケーリング設定を作成します。この設定では、App Runner は AWS リージョンに応じて、最大 5 つのアベイラビリティゾーンにサービスインスタンスを分散しようとします。

呼び出しは、他の設定がデフォルトに設定されているAutoScalingConfigurationオブジェクトを返します。この例では、これは という名前の設定を作成する最初の呼び出しですhigh-availability。リビジョンは 1 に設定され、最新のリビジョンです。

```
aws apprunner create-auto-scaling-configuration \  
--cli-input-json file://input.json
```

input.json の内容:

```
{  
  "AutoScalingConfigurationName": "high-availability",  
  "MinSize": 5  
}
```

出力:

```
{  
  "AutoScalingConfiguration": {  
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-east-1:123456789012:autoscalingconfiguration/high-availability/1/2f50e7656d7819fead0f59672e68042e",  
    "AutoScalingConfigurationName": "high-availability",  
    "AutoScalingConfigurationRevision": 1,  
    "CreatedAt": "2020-11-03T00:29:17Z",
```

```
    "Latest": true,  
    "Status": "ACTIVE",  
    "MaxConcurrency": 100,  
    "MaxSize": 50,  
    "MinSize": 5  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateAutoScalingConfiguration](#)」の「」を参照してください。

create-connection

次の例は、create-connection を使用する方法を説明しています。

AWS CLI

GitHub 接続を作成するには

次のcreate-connection例では、プライベート GitHub コードリポジトリへの接続を作成します。呼び出しが成功した後の接続ステータスは `PENDING_HANDSHAKE` です。これは、プロバイダーとの認証ハンドシェイクがまだ行われていないためです。App Runner コンソールを使用してハンドシェイクを完了します。

```
aws apprunner create-connection \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ConnectionName": "my-github-connection",  
  "ProviderType": "GITHUB"  
}
```

出力:

```
{  
  "Connection": {  
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-connection",  
    "ConnectionName": "my-github-connection",
```

```
    "Status": "PENDING_HANDSHAKE",
    "CreatedAt": "2020-11-03T00:32:51Z",
    "ProviderType": "GITHUB"
  }
}
```

詳細については、[App Runner デベロッパーガイドの「App Runner 接続の管理」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConnection](#)」の「」を参照してください。

create-service

次の例は、create-service を使用する方法を説明しています。

AWS CLI

例 1: ソースコードリポジトリサービスを作成するには

次のcreate-service例では、Python ソースコードリポジトリに基づいて App Runner サービスを作成します。

```
aws apprunner create-service \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "ServiceName": "python-app",
  "SourceConfiguration": {
    "AuthenticationConfiguration": {
      "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-connection/e7656250f67242d7819feade6800f59e"
    },
    "AutoDeploymentsEnabled": true,
    "CodeRepository": {
      "RepositoryUrl": "https://github.com/my-account/python-hello",
      "SourceCodeVersion": {
        "Type": "BRANCH",
        "Value": "main"
      }
    }
  },
}
```

```

    "CodeConfiguration": {
      "ConfigurationSource": "API",
      "CodeConfigurationValues": {
        "Runtime": "PYTHON_3",
        "BuildCommand": "pip install -r requirements.txt",
        "StartCommand": "python server.py",
        "Port": "8080",
        "RuntimeEnvironmentVariables": [
          {
            "NAME": "Jane"
          }
        ]
      }
    },
    "InstanceConfiguration": {
      "CPU": "1 vCPU",
      "Memory": "3 GB"
    }
  }
}

```

出力:

```

{
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
  "Service": {
    "CreatedAt": "2020-11-20T19:05:25Z",
    "UpdatedAt": "2020-11-20T19:05:25Z",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
        "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
      },
      "AutoDeploymentsEnabled": true,
      "CodeRepository": {
        "CodeConfiguration": {

```

```
    "CodeConfigurationValues": {
      "BuildCommand": "pip install -r requirements.txt",
      "Port": "8080",
      "Runtime": "PYTHON_3",
      "RuntimeEnvironmentVariables": [
        {
          "NAME": "Jane"
        }
      ],
      "StartCommand": "python server.py"
    },
    "ConfigurationSource": "Api"
  },
  "RepositoryUrl": "https://github.com/my-account/python-hello",
  "SourceCodeVersion": {
    "Type": "BRANCH",
    "Value": "main"
  }
}
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
  "CPU": "1 vCPU",
  "Memory": "3 GB"
}
}
```

例 2: ソースコードリポジトリサービスを作成するには

次の `create-service` 例では、Python ソースコードリポジトリに基づいて App Runner サービスを作成します。

```
aws apprunner create-service \
  --cli-input-json file://input.json
```

`input.json` の内容:

```
{
  "ServiceName": "python-app",
  "SourceConfiguration": {
    "AuthenticationConfiguration": {
```

```

    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection/e7656250f67242d7819feade6800f59e"
  },
  "AutoDeploymentsEnabled": true,
  "CodeRepository": {
    "RepositoryUrl": "https://github.com/my-account/python-hello",
    "SourceCodeVersion": {
      "Type": "BRANCH",
      "Value": "main"
    },
  },
  "CodeConfiguration": {
    "ConfigurationSource": "API",
    "CodeConfigurationValues": {
      "Runtime": "PYTHON_3",
      "BuildCommand": "pip install -r requirements.txt",
      "StartCommand": "python server.py",
      "Port": "8080",
      "RuntimeEnvironmentVariables": [
        {
          "NAME": "Jane"
        }
      ]
    }
  }
}
}
}
},
"InstanceConfiguration": {
  "CPU": "1 vCPU",
  "Memory": "3 GB"
}
}

```

出力:

```

{
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",
  "Service": {
    "CreatedAt": "2020-11-20T19:05:25Z",
    "UpdatedAt": "2020-11-20T19:05:25Z",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",
  }
}

```



```
"ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
"SourceConfiguration": {
  "AuthenticationConfiguration": {
    "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
  },
  "AutoDeploymentsEnabled": true,
  "CodeRepository": {
    "CodeConfiguration": {
      "CodeConfigurationValues": {
        "BuildCommand": "pip install -r requirements.txt",
        "Port": "8080",
        "Runtime": "PYTHON_3",
        "RuntimeEnvironmentVariables": [
          {
            "NAME": "Jane"
          }
        ],
        "StartCommand": "python server.py"
      },
      "ConfigurationSource": "Api"
    },
    "RepositoryUrl": "https://github.com/my-account/python-hello",
    "SourceCodeVersion": {
      "Type": "BRANCH",
      "Value": "main"
    }
  }
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
  "CPU": "1 vCPU",
  "Memory": "3 GB"
}
}
```

例 3: ソースイメージリポジトリサービスを作成するには

次の `create-service` 例では、Elastic Container Registry () に保存されているイメージに基づいて App Runner サービスを作成します ECR。

```
aws apprunner create-service \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ServiceName": "golang-container-app",  
  "SourceConfiguration": {  
    "AuthenticationConfiguration": {  
      "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"  
    },  
    "AutoDeploymentsEnabled": true,  
    "ImageRepository": {  
      "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/golang-  
app:latest",  
      "ImageConfiguration": {  
        "Port": "8080",  
        "RuntimeEnvironmentVariables": [  
          {  
            "NAME": "Jane"  
          }  
        ]  
      },  
      "ImageRepositoryType": "ECR"  
    }  
  },  
  "InstanceConfiguration": {  
    "CPU": "1 vCPU",  
    "Memory": "3 GB"  
  }  
}
```

出力:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-06T23:15:30Z",  
    "UpdatedAt": "2020-11-06T23:15:30Z",  
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-  
container-app/51728f8a20ce46d39b25398a6c8e9d1a",  
    "ServiceId": "51728f8a20ce46d39b25398a6c8e9d1a",  
  }  
}
```

```
"ServiceName": "golang-container-app",
"ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
"SourceConfiguration": {
  "AuthenticationConfiguration": {
    "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
  },
  "AutoDeploymentsEnabled": true,
  "ImageRepository": {
    "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/
golang-app:latest",
    "ImageConfiguration": {
      "Port": "8080",
      "RuntimeEnvironmentVariables": [
        {
          "NAME": "Jane"
        }
      ]
    },
    "ImageRepositoryType": "ECR"
  }
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
  "CPU": "1 vCPU",
  "Memory": "3 GB"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateService](#)」の「」を参照してください。

delete-auto-scaling-configuration

次のコード例は、delete-auto-scaling-configuration を使用する方法を示しています。

AWS CLI

例 1: 自動スケーリング設定の最新のアクティブなリビジョンを削除するには

次のdelete-auto-scaling-configuration例では、App Runner 自動スケーリング設定の最新のアクティブなリビジョンを削除します。最新のアクティブなリビジョンを削除するには、リ

ビジョンコンポーネントを使用せずに、設定名で終わる Amazon リソースネーム (ARN) を指定します。

この例では、このアクションの前に 2 つのリビジョンがあります。したがって、リビジョン 2 (最新) は削除されます。ただし、削除後に最新のアクティブなリビジョンではなくなる "Latest": false ため、が表示されるようになりました。

```
aws apprunner delete-auto-scaling-configuration \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability"  
}
```

出力:

```
{  
  "AutoScalingConfiguration": {  
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability/2/  
e76562f50d78042e819fead0f59672e6",  
    "AutoScalingConfigurationName": "high-availability",  
    "AutoScalingConfigurationRevision": 2,  
    "CreatedAt": "2021-02-25T17:42:59Z",  
    "DeletedAt": "2021-03-02T08:07:06Z",  
    "Latest": false,  
    "Status": "INACTIVE",  
    "MaxConcurrency": 30,  
    "MaxSize": 90,  
    "MinSize": 5  
  }  
}
```

例 2: 自動スケーリング設定の特定のリビジョンを削除するには

次の delete-auto-scaling-configuration 例では、App Runner 自動スケーリング設定の特定のリビジョンを削除します。特定のリビジョンを削除するには、リビジョン番号ARNを含むを指定します。

この例では、このアクションの前にいくつかのリビジョンがあります。アクションはリビジョンを削除します1。

```
aws apprunner delete-auto-scaling-configuration \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability/1"  
}
```

出力:

```
{  
  "AutoScalingConfiguration": {  
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-  
availability/1/2f50e7656d7819fead0f59672e68042e",  
    "AutoScalingConfigurationName": "high-availability",  
    "AutoScalingConfigurationRevision": 1,  
    "CreatedAt": "2020-11-03T00:29:17Z",  
    "DeletedAt": "2021-03-02T08:07:06Z",  
    "Latest": false,  
    "Status": "INACTIVE",  
    "MaxConcurrency": 100,  
    "MaxSize": 50,  
    "MinSize": 5  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAutoScalingConfiguration](#)」の「」を参照してください。

delete-connection

次のコード例は、delete-connection を使用する方法を示しています。

AWS CLI

接続を削除するには

次のdelete-connection例では、App Runner 接続を削除します。呼び出しが成功した後の接続ステータスは DELETED。これは、接続が使用できなくなったためです。

```
aws apprunner delete-connection \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-  
connection"  
}
```

出力:

```
{  
  "Connection": {  
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-  
github-connection",  
    "ConnectionName": "my-github-connection",  
    "Status": "DELETED",  
    "CreatedAt": "2020-11-03T00:32:51Z",  
    "ProviderType": "GITHUB"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteConnection](#)」の「」を参照してください。

delete-service

次のコード例は、delete-service を使用する方法を示しています。

AWS CLI

サービスを削除するには

次のdelete-service例では、App Runner サービスを削除します。

```
aws apprunner delete-service \  
--cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

出力:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-20T19:05:25Z",  
    "UpdatedAt": "2020-11-20T19:05:25Z",  
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceName": "python-app",  
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",  
    "SourceConfiguration": {  
      "AuthenticationConfiguration": {  
        "ConnectionArn": "arn:aws:apprunner:us-  
east-1:123456789012:connection/my-github-connection/  
e7656250f67242d7819feade6800f59e"  
      },  
      "AutoDeploymentsEnabled": true,  
      "CodeRepository": {  
        "CodeConfiguration": {  
          "CodeConfigurationValues": {  
            "BuildCommand": "pip install -r requirements.txt",  
            "Port": "8080",  
            "Runtime": "PYTHON_3",  
            "RuntimeEnvironmentVariables": [  
              {  
                "NAME": "Jane"  
              }  
            ],  
          }  
        },  
      }  
    }  
  }  
}
```

```

        "StartCommand": "python server.py"
    },
    "ConfigurationSource": "Api"
},
"RepositoryUrl": "https://github.com/my-account/python-hello",
"SourceCodeVersion": {
    "Type": "BRANCH",
    "Value": "main"
}
}
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
    "CPU": "1 vCPU",
    "Memory": "3 GB"
}
}
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteService](#)」の「」を参照してください。

describe-auto-scaling-configuration

次の例は、describe-auto-scaling-configuration を使用する方法を説明しています。

AWS CLI

例 1: 自動スケーリング設定の最新のアクティブなリビジョンを記述するには

次のdescribe-auto-scaling-configuration例では、App Runner 自動スケーリング設定の最新アクティブリビジョンの説明を取得します。最新のアクティブなリビジョンを記述するには、リビジョンコンポーネントを使用せずに、設定名でARN終わる を指定します。

この例では、2つのリビジョンがあります。したがって、リビジョン 2 (最新) について説明します。結果のオブジェクトには が表示されます"Latest": true。

```
aws apprunner describe-auto-scaling-configuration \
  --cli-input-json file://input.json
```

input.json の内容:


```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

出力:

```
{
  "AutoScalingConfiguration": {
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
    "AutoScalingConfigurationName": "high-availability",
    "AutoScalingConfigurationRevision": 2,
    "CreatedAt": "2021-02-25T17:42:59Z",
    "Latest": true,
    "Status": "ACTIVE",
    "MaxConcurrency": 30,
    "MaxSize": 90,
    "MinSize": 5
  }
}
```

例 2: 自動スケーリング設定の特定のレビジョンを記述するには

次のdescribe-auto-scaling-configuration例では、App Runner 自動スケーリング設定の特定のレビジョンの説明を取得します。特定のレビジョンを記述するには、レビジョン番号 ARNを含む を指定します。

この例では、複数のレビジョンが存在し、レビジョン1がクエリされます。結果のオブジェクトには が表示されます"Latest": false。

```
aws apprunner describe-auto-scaling-configuration \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

出力:

```
{
  "AutoScalingConfiguration": {
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-
availability/1/2f50e7656d7819fead0f59672e68042e",
    "AutoScalingConfigurationName": "high-availability",
    "AutoScalingConfigurationRevision": 1,
    "CreatedAt": "2020-11-03T00:29:17Z",
    "Latest": false,
    "Status": "ACTIVE",
    "MaxConcurrency": 100,
    "MaxSize": 50,
    "MinSize": 5
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAutoScalingConfiguration](#)」の「」を参照してください。

describe-custom-domains

次のコード例は、describe-custom-domains を使用する方法を示しています。

AWS CLI

サービスに関連付けられたカスタムドメイン名の説明を取得するには

次のdescribe-custom-domains例では、App Runner サービスに関連付けられたカスタムドメイン名の説明とステータスを取得します。

```
aws apprunner describe-custom-domains \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "DomainName": "example.com",
}
```

```
"EnableWWWSubdomain": true
}
```

出力:

```
{
  "CustomDomains": [
    {
      "CertificateValidationRecords": [
        {
          "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
          "Status": "PENDING_VALIDATION",
          "Type": "CNAME",
          "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
        },
        {
          "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
          "Status": "PENDING_VALIDATION",
          "Type": "CNAME",
          "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
        }
      ],
      "DomainName": "example.com",
      "EnableWWWSubdomain": true,
      "Status": "PENDING_CERTIFICATE_DNS_VALIDATION"
    },
    {
      "CertificateValidationRecords": [
        {
          "Name": "_a94f784c70d3f507c72dc28f55db2f6b.deals.example.com",
          "Status": "SUCCESS",
          "Type": "CNAME",
          "Value": "_2db02504c1270c137383c6307b6834b0.bsgbmzkfwj.acm-
validations.aws."
        }
      ],
      "DomainName": "deals.example.com",
      "EnableWWWSubdomain": false,
      "Status": "ACTIVE"
    }
  ],
}
```

```
"DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
"ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeCustomDomains](#)」の「」を参照してください。

describe-service

次の例は、describe-service を使用する方法を説明しています。

AWS CLI

サービスを記述するには

次のdescribe-service例では、App Runner サービスの説明を取得します。

```
aws apprunner describe-service \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

出力:

```
{
  "Service": {
    "CreatedAt": "2020-11-20T19:05:25Z",
    "UpdatedAt": "2020-11-20T19:05:25Z",
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
```

```
    "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
  },
  "AutoDeploymentsEnabled": true,
  "CodeRepository": {
    "CodeConfiguration": {
      "CodeConfigurationValues": {
        "BuildCommand": "pip install -r requirements.txt",
        "Port": "8080",
        "Runtime": "PYTHON_3",
        "RuntimeEnvironmentVariables": [
          {
            "NAME": "Jane"
          }
        ],
        "StartCommand": "python server.py"
      },
      "ConfigurationSource": "Api"
    },
    "RepositoryUrl": "https://github.com/my-account/python-hello",
    "SourceCodeVersion": {
      "Type": "BRANCH",
      "Value": "main"
    }
  }
},
"Status": "RUNNING",
"InstanceConfiguration": {
  "CPU": "1 vCPU",
  "Memory": "3 GB"
}
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeService](#)」の「」を参照してください。

disassociate-custom-domain

次のコード例は、disassociate-custom-domain を使用する方法を示しています。

AWS CLI

サービスからドメイン名の関連付けを解除するには

次の`disassociate-custom-domain`例では、App Runner サービス`example.com`からドメインの関連付けを解除します。コールは、ルートドメイン`www.example.com`に関連付けられているサブドメインの関連付けも解除します。

```
aws apprunner disassociate-custom-domain \  
  --cli-input-json file://input.json
```

`input.json` の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "DomainName": "example.com"  
}
```

出力:

```
{  
  "CustomDomain": {  
    "CertificateValidationRecords": [  
      {  
        "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",  
        "Status": "PENDING_VALIDATION",  
        "Type": "CNAME",  
        "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-  
validations.aws."  
      },  
      {  
        "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",  
        "Status": "PENDING_VALIDATION",  
        "Type": "CNAME",  
        "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-  
validations.aws."  
      }  
    ],  
    "DomainName": "example.com",  
    "EnableWWWSubdomain": true,  
    "Status": "DELETING"  
  }  
}
```

```
  },
  "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateCustomDomain](#)」の「」を参照してください。

list-auto-scaling-configurations

次のコード例は、list-auto-scaling-configurations を使用する方法を示しています。

AWS CLI

App Runner 自動スケーリング設定のページ分割リストを取得するには

次のlist-auto-scaling-configurations例では、AWS アカウント内のすべての App Runner 自動スケーリング設定を一覧表示します。各レスポンスには、最大 5 つの自動スケーリング設定が一覧表示されます。AutoScalingConfigurationName および LatestOnlyは指定されません。それらのデフォルトにより、すべてのアクティブな設定の最新リビジョンが一覧表示されます。

この例では、レスポンスに 2 つの結果が含まれており、追加の結果がないため、返NextTokenされません。

```
aws apprunner list-auto-scaling-configurations \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "MaxResults": 5
}
```

出力:

```
{
  "AutoScalingConfigurationSummaryList": [
    {
```

```
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/2/
e76562f50d78042e819fead0f59672e6",
    "AutoScalingConfigurationName": "high-availability",
    "AutoScalingConfigurationRevision": 2
  },
  {
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/low-
cost/1/50d7804e7656fead0f59672e62f2e819",
    "AutoScalingConfigurationName": "low-cost",
    "AutoScalingConfigurationRevision": 1
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAutoScalingConfigurations](#)」の「」を参照してください。

list-connections

次の例は、list-connections を使用する方法を説明しています。

AWS CLI

例 1: すべての接続を一覧表示するには

次のlist-connections例では、AWS アカウント内のすべての App Runner 接続を一覧表示します。

```
aws apprunner list-connections
```

出力:

```
{
  "ConnectionSummaryList": [
    {
      "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-connection",
      "ConnectionName": "my-github-connection",
      "Status": "AVAILABLE",
```



```
    "CreatedAt": "2020-11-03T00:32:51Z",
    "ProviderType": "GITHUB"
  },
  {
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
    "ConnectionName": "my-github-org-connection",
    "Status": "AVAILABLE",
    "CreatedAt": "2020-11-03T02:54:17Z",
    "ProviderType": "GITHUB"
  }
]
}
```

例 2: 接続を名前で一覧表示するには

次のlist-connections例では、接続をその名前で一覧表示します。

```
aws apprunner list-connections \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "ConnectionName": "my-github-org-connection"
}
```

出力:

```
{
  "ConnectionSummaryList": [
    {
      "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/
my-github-org-connection",
      "ConnectionName": "my-github-org-connection",
      "Status": "AVAILABLE",
      "CreatedAt": "2020-11-03T02:54:17Z",
      "ProviderType": "GITHUB"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListConnections](#)」の「」を参照してください。

list-operations

次のコード例は、list-operations を使用する方法を示しています。

AWS CLI

サービスで発生したオペレーションを一覧表示するには

次のlist-operations例では、これまでに App Runner サービスで発生したすべてのオペレーションを一覧表示します。この例では、サービスは新しく、タイプのオペレーションCREATE_SERVICEは 1 つだけ発生しています。

```
aws apprunner list-operations \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

出力:

```
{  
  "OperationSummaryList": [  
    {  
      "EndedAt": 1606156217,  
      "Id": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
      "StartedAt": 1606156014,  
      "Status": "SUCCEEDED",  
      "TargetArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
      "Type": "CREATE_SERVICE",  
      "UpdatedAt": 1606156217  
    }  
  ]  
}
```



```
    },
    {
      "CreatedAt": "2020-11-06T23:15:30Z",
      "UpdatedAt": "2020-11-23T13:21:22Z",
      "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/golang-
container-app/ab8f94cfe29a460fb8760afd2ee87555",
      "ServiceId": "ab8f94cfe29a460fb8760afd2ee87555",
      "ServiceName": "golang-container-app",
      "ServiceUrl": "e2m8rrrx33.us-east-1.awsapprunner.com",
      "Status": "RUNNING"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServices](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

App Runner サービスに関連付けられたタグを一覧表示するには

次のlist-tags-for-resource例では、App Runner サービスに関連付けられているすべてのタグを一覧表示します。

```
aws apprunner list-tags-for-resource \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

出力:

```
{
```

```
"Tags": [  
  {  
    "Key": "Department",  
    "Value": "Retail"  
  },  
  {  
    "Key": "CustomerId",  
    "Value": "56439872357912"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

pause-service

次のコード例は、`pause-service` を使用する方法を示しています。

AWS CLI

サービスを一時停止するには

次の`pause-service`例では、App Runner サービスを一時停止します。

```
aws apprunner pause-service \  
  --cli-input-json file://input.json
```

`input.json` の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

出力:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-20T19:05:25Z",  
    "UpdatedAt": "2020-11-23T12:41:37Z",
```

```
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "ServiceName": "python-app",
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",
    "SourceConfiguration": {
      "AuthenticationConfiguration": {
        "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
      },
      "AutoDeploymentsEnabled": true,
      "CodeRepository": {
        "CodeConfiguration": {
          "CodeConfigurationValues": {
            "BuildCommand": "pip install -r requirements.txt",
            "Port": "8080",
            "Runtime": "PYTHON_3",
            "RuntimeEnvironmentVariables": [
              {
                "NAME": "Jane"
              }
            ],
            "StartCommand": "python server.py"
          },
          "ConfigurationSource": "Api"
        },
        "RepositoryUrl": "https://github.com/my-account/python-hello",
        "SourceCodeVersion": {
          "Type": "BRANCH",
          "Value": "main"
        }
      }
    },
    "Status": "OPERATION_IN_PROGRESS",
    "InstanceConfiguration": {
      "CPU": "1 vCPU",
      "Memory": "3 GB"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PauseService](#)」の「」を参照してください。

resume-service

次の例は、resume-service を使用する方法を説明しています。

AWS CLI

サービスを再開するには

次のresume-service例では、App Runner サービスを再開します。

```
aws apprunner resume-service \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

出力:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-20T19:05:25Z",  
    "UpdatedAt": "2020-11-23T12:41:37Z",  
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceName": "python-app",  
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",  
    "SourceConfiguration": {  
      "AuthenticationConfiguration": {  
        "ConnectionArn": "arn:aws:apprunner:us-  
east-1:123456789012:connection/my-github-connection/  
e7656250f67242d7819feade6800f59e"  
      },  
    },  
  },  
}
```

```
    "AutoDeploymentsEnabled": true,
    "CodeRepository": {
      "CodeConfiguration": {
        "CodeConfigurationValues": {
          "BuildCommand": "pip install -r requirements.txt",
          "Port": "8080",
          "Runtime": "PYTHON_3",
          "RuntimeEnvironmentVariables": [
            {
              "NAME": "Jane"
            }
          ],
          "StartCommand": "python server.py"
        },
        "ConfigurationSource": "Api"
      },
      "RepositoryUrl": "https://github.com/my-account/python-hello",
      "SourceCodeVersion": {
        "Type": "BRANCH",
        "Value": "main"
      }
    }
  },
  "Status": "OPERATION_IN_PROGRESS",
  "InstanceConfiguration": {
    "CPU": "1 vCPU",
    "Memory": "3 GB"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResumeService](#)」の「」を参照してください。

start-deployment

次の例は、start-deployment を使用する方法を説明しています。

AWS CLI

手動デプロイを開始するには

次のstart-deployment例では、App Runner サービスへの手動デプロイを実行します。


```
aws apprunner start-deployment \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

出力:

```
{  
  "OperationId": "853a7d5b-fc9f-4730-831b-fd8037ab832a"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [StartDeployment](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

App Runner サービスにタグを追加するには

次のtag-resource例では、App Runner サービスに 2 つのタグを追加します。

```
aws apprunner tag-resource \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "Tags": [  
    {  
      "Key": "Department",
```

```
        "Value": "Retail"
      },
      {
        "Key": "CustomerId",
        "Value": "56439872357912"
      }
    ]
  }
}
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、`untag-resource` を使用する方法を示しています。

AWS CLI

App Runner サービスからタグを削除するには

次の`untag-resource`例では、App Runner サービスから 2 つのタグを削除します。

```
aws apprunner untag-resource \  
  --cli-input-json file://input.json
```

`input.json` の内容:

```
{  
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "TagKeys": [  
    "Department",  
    "CustomerId"  
  ]  
}
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-service

次のコード例は、update-service を使用する方法を示しています。

AWS CLI

メモリサイズを更新するには

次のupdate-service例では、App Runner サービスのインスタンス (スケーリングユニット) のメモリサイズを 2048 MiB に更新します。

呼び出しが成功すると、App Runner は非同期更新プロセスを開始します。呼び出しによって返されるService構造には、この呼び出しによって適用される新しいメモリ値が反映されます。

```
aws apprunner update-service \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "InstanceConfiguration": {  
    "Memory": "4 GB"  
  }  
}
```

出力:

```
{  
  "OperationId": "17fe9f55-7e91-4097-b243-fcabbb69a4cf",  
  "Service": {  
    "CreatedAt": "2020-11-20T19:05:25Z",  
    "UpdatedAt": "2020-11-23T12:41:37Z",  
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",  
    "ServiceName": "python-app",  
    "ServiceUrl": "psbqam834h.us-east-1.awsapprunner.com",  
    "SourceConfiguration": {  
      "AuthenticationConfiguration": {
```

```
    "ConnectionArn": "arn:aws:apprunner:us-
east-1:123456789012:connection/my-github-connection/
e7656250f67242d7819feade6800f59e"
  },
  "AutoDeploymentsEnabled": true,
  "CodeRepository": {
    "CodeConfiguration": {
      "CodeConfigurationValues": {
        "BuildCommand": "pip install -r requirements.txt",
        "Port": "8080",
        "Runtime": "PYTHON_3",
        "RuntimeEnvironmentVariables": [
          {
            "NAME": "Jane"
          }
        ],
        "StartCommand": "python server.py"
      },
      "ConfigurationSource": "Api"
    },
    "RepositoryUrl": "https://github.com/my-account/python-hello",
    "SourceCodeVersion": {
      "Type": "BRANCH",
      "Value": "main"
    }
  }
},
"Status": "OPERATION_IN_PROGRESS",
"InstanceConfiguration": {
  "CPU": "1 vCPU",
  "Memory": "4 GB"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateService](#)」の「」を参照してください。

AWS AppConfig を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS AppConfig。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-application

次のコード例は、create-application を使用する方法を示しています。

AWS CLI

アプリケーションを作成するには

次のcreate-application例では、 でアプリケーションを作成します AWS AppConfig。

```
aws appconfig create-application \  
  --name "example-application" \  
  --description "An application used for creating an example."
```

出力:

```
{  
  "Description": "An application used for creating an example.",  
  "Id": "339ohji",  
  "Name": "example-application"  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 1: AWS AppConfig アプリケーションの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateApplication](#)」の「」を参照してください。

create-configuration-profile

次の例は、create-configuration-profile を使用する方法を説明しています。

AWS CLI

設定プロファイルを作成するには

次のcreate-configuration-profile例では、Systems Manager の一機能である Parameter Store に保存されている設定を使用して設定プロファイルを作成します。

```
aws appconfig create-configuration-profile \  
  --application-id "339ohji" \  
  --name "Example-Configuration-Profile" \  
  --location-uri "ssm-parameter://Example-Parameter" \  
  --retrieval-role-arn "arn:aws:iam::111122223333:role/Example-App-Config-Role"
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "Description": null,  
  "Id": "ur8hx2f",  
  "LocationUri": "ssm-parameter://Example-Parameter",  
  "Name": "Example-Configuration-Profile",  
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role",  
  "Type": null,  
  "Validators": null  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 3: 設定と設定プロファイルの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateConfigurationProfile](#)」の「」を参照してください。

create-environment

次の例は、create-environment を使用する方法を説明しています。

AWS CLI

環境を作成するには

次のcreate-environment例では、create-application を使用して作成したアプリケーションを使用して Example-Environment という名前の AWS AppConfig 環境を作成します。

```
aws appconfig create-environment \  
  --application-id "339ohji" \  
  --name "Example-Environment"
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "Description": null,  
  "Id": "54j1r29",  
  "Monitors": null,  
  "Name": "Example-Environment",  
  "State": "ReadyForDeployment"  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 2: 環境の作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateEnvironment](#)」の「」を参照してください。

create-extension-association

次のコード例は、create-extension-association を使用する方法を示しています。

AWS CLI

拡張機能の関連付けを作成するには

次のcreate-extension-association例では、 に新しい拡張機能の関連付けを作成します AWS AppConfig。

```
aws appconfig create-extension-association \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension \  
  --resource-identifier "arn:aws:appconfig:us-west-2:123456789012:application/  
Finance" \  
  --parameters S3bucket=FinanceConfigurationBackup
```

出力:

```
{
  "Id": "a1b2c3d4",
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-extension/1",
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",
  "Parameters": {
    "S3bucket": "FinanceConfigurationBackup"
  },
  "ExtensionVersionNumber": 1
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateExtensionAssociation](#)」の「」を参照してください。

create-extension

次の例は、create-extension を使用する方法を説明しています。

AWS CLI

拡張機能を作成するには

次のcreate-extension例では、 に新しい拡張機能を作成します AWS AppConfig。

```
aws appconfig create-extension \
  --region us-west-2 \
  --name S3-backup-extension \
  --
actions PRE_CREATE_HOSTED_CONFIGURATION_VERSION=[{Name=S3backup,Uri=arn:aws:lambda:us-west-2:123456789012:function:s3backupfunction,RoleArn=arn:aws:iam::123456789012:role/appconfigextensionrole}] \
  --parameters S3bucket={Required=true}
```

出力:

```
{
```



```
"Id": "1A2B3C4D",
>Name": "S3-backup-extension",
>VersionNumber": 1,
>Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",
>Actions": {
  "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
    {
      "Name": "S3backup",
      "Uri": "arn:aws:lambda:us-
west-2:123456789012:function:s3backupfunction",
      "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"
    }
  ]
},
>Parameters": {
  "S3bucket": {
    "Required": true
  }
}
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateExtension](#)」の「」を参照してください。

create-hosted-configuration-version

次の例は、create-hosted-configuration-version を使用する方法を説明しています。

AWS CLI

ホスト設定バージョンを作成するには

次のcreate-hosted-configuration-version例では、AWS AppConfig ホストされた設定ストアに新しい設定を作成します。設定コンテンツは、まず base64 に変換する必要があります。

```
aws appconfig create-hosted-configuration-version \
  --application-id "339ohji" \
  --configuration-profile-id "ur8hx2f" \
```

```
--  
content eyAiTmFtZSI6ICJFeGFtcGxlQXBwbGljYXRpb24iLCAiSWQiOiBFFeGFtcGxlSUQsICJSYW5rIjogMyB9  
\  
--content-type "application/json" \  
configuration_version_output_file
```

configuration_version_output_file の内容:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "ConfigurationProfileId": "ur8hx2f",  
  "VersionNumber": "1",  
  "ContentType": "application/json"  
}
```

詳細については、「Appconfig ユーザーガイド」の[AWS AppConfig 「ホストされた設定ストアについて」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateHostedConfigurationVersion](#)」の「」を参照してください。

delete-application

次の例は、delete-application を使用する方法を説明しています。

AWS CLI

アプリケーションを削除するには

次のdelete-application例では、指定されたアプリケーションを削除します。

```
aws appconfig delete-application \  
--application-id 339ohji
```

このコマンドでは何も出力されません。

詳細については、AWS AppConfig 「ユーザーガイド」の[「ステップ 1: AWS AppConfig アプリケーションの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApplication](#)」の「」を参照してください。

delete-configuration-profile

次の例は、delete-configuration-profile を使用する方法を説明しています。

AWS CLI

設定プロファイルを削除するには

次のdelete-configuration-profile例では、指定された設定プロファイルを削除します。

```
aws appconfig delete-configuration-profile \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f
```

このコマンドでは何も出力されません。

詳細については、AWS AppConfig 「ユーザーガイド」の「[ステップ 3: 設定と設定プロファイルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConfigurationProfile](#)」の「」を参照してください。

delete-deployment-strategy

次の例は、delete-deployment-strategy を使用する方法を説明しています。

AWS CLI

デプロイ戦略を削除するには

次のdelete-deployment-strategy例では、指定されたデプロイ戦略を削除します。

```
aws appconfig delete-deployment-strategy \  
  --deployment-strategy-id 1225qzk
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[ステップ 4: デプロイ戦略の作成](#)」を参照してください。AWS AppConfig

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDeploymentStrategy](#)」の「」を参照してください。

delete-environment

次の例は、delete-environment を使用する方法を説明しています。

AWS CLI

環境を削除するには

次のdelete-environment例では、指定されたアプリケーション環境を削除します。

```
aws appconfig delete-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

このコマンドでは何も出力されません。

詳細については、AWS AppConfig 「ユーザーガイド」の「[ステップ 2: 環境の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEnvironment](#)」の「」を参照してください。

delete-extension-association

次のコード例は、delete-extension-association を使用する方法を示しています。

AWS CLI

拡張機能の関連付けを削除するには

次のdelete-extension-association例では、 から拡張機能の関連付けを削除します AWS AppConfig。

```
aws appconfig delete-extension-association \  
  --region us-west-2 \  
  --extension-association-id a1b2c3d4
```

このコマンドでは何も出力されません。

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteExtensionAssociation](#)」の「」を参照してください。

delete-extension

次のコード例は、delete-extension を使用する方法を示しています。

AWS CLI

拡張機能を削除するには

次のdelete-extension例では、 から拡張機能を削除します AWS AppConfig。

```
aws appconfig delete-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension
```

このコマンドでは何も出力されません。

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteExtension](#)」の「」を参照してください。

delete-hosted-configuration-version

次の例は、delete-hosted-configuration-version を使用する方法を説明しています。

AWS CLI

ホスト設定バージョンを削除するには

次のdelete-hosted-configuration-version例では、ホストされた設定ストアで AWS AppConfig ホストされている設定バージョンを削除します。

```
aws appconfig delete-hosted-configuration-version \  
  --application-id 339ohji \  
  --configuration-version-id 1234567890
```

```
--configuration-profile-id ur8hx2f \  
--version-number 1
```

出力:: このコマンドは出力を生成しません。

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 3: 設定と設定プロファイルの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteHostedConfigurationVersion](#)」の「」を参照してください。

get-application

次のコード例は、get-application を使用する方法を示しています。

AWS CLI

アプリケーションの詳細を一覧表示するには

次のget-application例では、指定されたアプリケーションの詳細を一覧表示します。

```
aws appconfig get-application \  
--application-id 339ohji
```

出力:

```
{  
  "Description": "An application used for creating an example.",  
  "Id": "339ohji",  
  "Name": "example-application"  
}
```

詳細については、「ユーザーガイド」の [「AWS AppConfig の仕組み」](#)を参照してください。

AWS AppConfig

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApplication](#)」の「」を参照してください。

get-configuration-profile

次の例は、get-configuration-profile を使用する方法を説明しています。

AWS CLI

設定プロファイルの詳細を取得するには

次のget-configuration-profile例では、指定された設定プロファイルの詳細を返します。

```
aws appconfig get-configuration-profile \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "ur8hx2f",  
  "Name": "Example-Configuration-Profile",  
  "LocationUri": "ssm-parameter://Example-Parameter",  
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[「ステップ 3: 設定と設定プロファイルの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConfigurationProfile](#)」の「」を参照してください。

get-configuration

次の例は、get-configuration を使用する方法を説明しています。

AWS CLI

設定の詳細を取得するには

次のget-configuration例では、サンプルアプリケーションの設定の詳細を返します。後続のget-configuration 呼び出しでは、client-configuration-versionパラメータを使用して、バージョンが変更された場合にのみアプリケーションの設定を更新します。バージョンが変更された場合にのみ設定を更新すると、get-configuration を呼び出すことで発生する超過料金を回避できます。

```
aws appconfig get-configuration \  
  --application "example-application" \  
  --client-configuration-version 1
```

```
--environment "Example-Environment" \  
--configuration "Example-Configuration-Profile" \  
--client-id "test-id" \  
configuration-output-file
```

configuration-output-file の内容:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

出力:

```
{  
  "ConfigurationVersion": "1",  
  "ContentType": "application/json"  
}
```

詳細については、AWS AppConfig 「[ユーザーガイド](#)」の「[ステップ 6: 設定の受信](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetConfiguration](#)」の「」を参照してください。

get-deployment-strategy

次の例は、get-deployment-strategy を使用する方法を説明しています。

AWS CLI

デプロイ戦略の詳細を取得するには

次のget-deployment-strategy例では、指定されたデプロイ戦略の詳細を一覧表示します。

```
aws appconfig get-deployment-strategy \  
--deployment-strategy-id 1225qzk
```

出力:

```
{  
  "Id": "1225qzk",  
  "Name": "Example-Deployment",  
  "DeploymentDurationInMinutes": 15,  
  "GrowthType": "LINEAR",
```



```
"GrowthFactor": 25.0,  
"FinalBakeTimeInMinutes": 0,  
"ReplicateTo": "SSM_DOCUMENT"  
}
```

詳細については、「ユーザーガイド」の[「ステップ 4: デプロイ戦略の作成」](#)を参照してください。AWS AppConfig

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeploymentStrategy](#)」の「」を参照してください。

get-deployment

次のコード例は、get-deployment を使用する方法を示しています。

AWS CLI

デプロイの詳細を取得するには

次のget-deployment例は、指定された環境とデプロイ内のアプリケーションへのデプロイの詳細を一覧表示します。

```
aws appconfig get-deployment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-number 1
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "EnvironmentId": "54j1r29",  
  "DeploymentStrategyId": "1225qzk",  
  "ConfigurationProfileId": "ur8hx2f",  
  "DeploymentNumber": 1,  
  "ConfigurationName": "Example-Configuration-Profile",  
  "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",  
  "ConfigurationVersion": "1",  
  "DeploymentDurationInMinutes": 15,  
  "GrowthType": "LINEAR",  
  "GrowthFactor": 25.0,  
  "FinalBakeTimeInMinutes": 0,  
  "State": "COMPLETE",
```

```
"EventLog": [  
  {  
    "EventType": "DEPLOYMENT_COMPLETED",  
    "TriggeredBy": "APPCONFIG",  
    "Description": "Deployment completed",  
    "OccurredAt": "2021-09-17T21:59:03.888000+00:00"  
  },  
  {  
    "EventType": "BAKE_TIME_STARTED",  
    "TriggeredBy": "APPCONFIG",  
    "Description": "Deployment bake time started",  
    "OccurredAt": "2021-09-17T21:58:57.722000+00:00"  
  },  
  {  
    "EventType": "PERCENTAGE_UPDATED",  
    "TriggeredBy": "APPCONFIG",  
    "Description": "Configuration available to 100.00% of clients",  
    "OccurredAt": "2021-09-17T21:55:56.816000+00:00"  
  },  
  {  
    "EventType": "PERCENTAGE_UPDATED",  
    "TriggeredBy": "APPCONFIG",  
    "Description": "Configuration available to 75.00% of clients",  
    "OccurredAt": "2021-09-17T21:52:56.567000+00:00"  
  },  
  {  
    "EventType": "PERCENTAGE_UPDATED",  
    "TriggeredBy": "APPCONFIG",  
    "Description": "Configuration available to 50.00% of clients",  
    "OccurredAt": "2021-09-17T21:49:55.737000+00:00"  
  },  
  {  
    "EventType": "PERCENTAGE_UPDATED",  
    "TriggeredBy": "APPCONFIG",  
    "Description": "Configuration available to 25.00% of clients",  
    "OccurredAt": "2021-09-17T21:46:55.187000+00:00"  
  },  
  {  
    "EventType": "DEPLOYMENT_STARTED",  
    "TriggeredBy": "USER",  
    "Description": "Deployment started",  
    "OccurredAt": "2021-09-17T21:43:54.205000+00:00"  
  }  
],
```

```
"PercentageComplete": 100.0,  
"StartedAt": "2021-09-17T21:43:54.205000+00:00",  
"CompletedAt": "2021-09-17T21:59:03.888000+00:00"  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の「[ステップ 5: 設定のデプロイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeployment](#)」の「」を参照してください。

get-environment

次のコード例は、get-environment を使用する方法を示しています。

AWS CLI

環境の詳細を取得するには

次のget-environment例では、指定された環境の詳細と状態を返します。

```
aws appconfig get-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "54j1r29",  
  "Name": "Example-Environment",  
  "State": "ReadyForDeployment"  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の「[ステップ 2: 環境の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEnvironment](#)」の「」を参照してください。

get-extension-association

次の例は、get-extension-association を使用する方法を説明しています。

AWS CLI

拡張機能の関連付けの詳細を取得するには

次のget-extension-association例は、拡張機能の関連付けに関する情報を示しています。

```
aws appconfig get-extension-association \  
  --region us-west-2 \  
  --extension-association-id a1b2c3d4
```

出力:

```
{  
  "Id": "a1b2c3d4",  
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-extension/1",  
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",  
  "Parameters": {  
    "S3bucket": "FinanceConfigurationBackup"  
  },  
  "ExtensionVersionNumber": 1  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetExtensionAssociation](#)」の「」を参照してください。

get-extension

次のコード例は、get-extension を使用する方法を示しています。

AWS CLI

拡張機能の詳細を取得するには

次のget-extension例は、拡張機能に関する情報を示しています。

```
aws appconfig get-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension
```

出力:

```
{  
  "Id": "1A2B3C4D",  
  "Name": "S3-backup-extension",  
  "VersionNumber": 1,  
  "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-  
extension/1",  
  "Actions": {  
    "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [  
      {  
        "Name": "S3backup",  
        "Uri": "arn:aws:lambda:us-  
west-2:123456789012:function:S3backupfunction",  
        "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"  
      }  
    ]  
  },  
  "Parameters": {  
    "S3bucket": {  
      "Required": true  
    }  
  }  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetExtension](#)」の「」を参照してください。

get-hosted-configuration-version

次の例は、get-hosted-configuration-version を使用する方法を説明しています。

AWS CLI

ホスト設定の詳細を取得するには

次のget-hosted-configuration-version例では、AWS AppConfig ホストされた設定の設定の詳細を取得します。

```
aws appconfig get-hosted-configuration-version \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f \  
  --version-number 1 \  
  hosted-configuration-version-output
```

hosted-configuration-version-output の内容:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "ConfigurationProfileId": "ur8hx2f",  
  "VersionNumber": "1",  
  "ContentType": "application/json"  
}
```

詳細については、AWS AppConfig 「[ユーザーガイド](#)」の [AWS AppConfig 「ホストされた設定ストアについて」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetHostedConfigurationVersion](#)」の「」を参照してください。

list-applications

次の例は、list-applications を使用する方法を説明しています。

AWS CLI

使用可能なアプリケーションを一覧表示するには

次のlist-applications例は、AWS アカウントで使用可能なアプリケーションを一覧表示します。

```
aws appconfig list-applications
```

出力:

```
{
  "Items": [
    {
      "Id": "339ohji",
      "Name": "test-application",
      "Description": "An application used for creating an example."
    },
    {
      "Id": "rwalwu7",
      "Name": "Test-Application"
    }
  ]
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[「ステップ 1: AWS AppConfig アプリケーションの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListApplications](#)」の「」を参照してください。

list-configuration-profiles

次のコード例は、list-configuration-profiles を使用する方法を示しています。

AWS CLI

使用可能な設定プロファイルを一覧表示するには

次のlist-configuration-profiles例では、指定されたアプリケーションで使用可能な設定プロファイルを一覧表示します。

```
aws appconfig list-configuration-profiles \
  --application-id 339ohji
```

出力:

```
{
  "Items": [
    {
      "ApplicationId": "339ohji",
```

```
        "Id": "ur8hx2f",
        "Name": "Example-Configuration-Profile",
        "LocationUri": "ssm-parameter://Example-Parameter"
    }
]
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 3: 設定と設定プロファイルの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListConfigurationProfiles](#)」の「」を参照してください。

list-deployment-strategies

次の例は、list-deployment-strategies を使用する方法を説明しています。

AWS CLI

利用可能なデプロイ戦略を一覧表示するには

次のlist-deployment-strategies例では、AWS アカウントで使用可能なデプロイ戦略を一覧表示します。

```
aws appconfig list-deployment-strategies
```

出力:

```
{
  "Items": [
    {
      "Id": "1225qzk",
      "Name": "Example-Deployment",
      "DeploymentDurationInMinutes": 15,
      "GrowthType": "LINEAR",
      "GrowthFactor": 25.0,
      "FinalBakeTimeInMinutes": 0,
      "ReplicateTo": "SSM_DOCUMENT"
    },
    {
      "Id": "AppConfig.AllAtOnce",
      "Name": "AppConfig.AllAtOnce",

```



```
    "Description": "Quick",
    "DeploymentDurationInMinutes": 0,
    "GrowthType": "LINEAR",
    "GrowthFactor": 100.0,
    "FinalBakeTimeInMinutes": 10,
    "ReplicateTo": "NONE"
  },
  {
    "Id": "AppConfig.Linear50PercentEvery30Seconds",
    "Name": "AppConfig.Linear50PercentEvery30Seconds",
    "Description": "Test/Demo",
    "DeploymentDurationInMinutes": 1,
    "GrowthType": "LINEAR",
    "GrowthFactor": 50.0,
    "FinalBakeTimeInMinutes": 1,
    "ReplicateTo": "NONE"
  },
  {
    "Id": "AppConfig.Canary10Percent20Minutes",
    "Name": "AppConfig.Canary10Percent20Minutes",
    "Description": "AWS Recommended",
    "DeploymentDurationInMinutes": 20,
    "GrowthType": "EXPONENTIAL",
    "GrowthFactor": 10.0,
    "FinalBakeTimeInMinutes": 10,
    "ReplicateTo": "NONE"
  }
]
}
```

詳細については、「ユーザーガイド」の[「ステップ 4: デプロイ戦略の作成」](#)を参照してください。AWS AppConfig

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeploymentStrategies](#)」の「」を参照してください。

list-deployments

次のコード例は、list-deployments を使用する方法を示しています。

AWS CLI

使用可能なデプロイを一覧表示するには

次のlist-deployments例では、指定されたアプリケーションと環境の AWS アカウントで使用可能なデプロイを一覧表示します。

```
aws appconfig list-deployments \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

出力:

```
{  
  "Items": [  
    {  
      "DeploymentNumber": 1,  
      "ConfigurationName": "Example-Configuration-Profile",  
      "ConfigurationVersion": "1",  
      "DeploymentDurationInMinutes": 15,  
      "GrowthType": "LINEAR",  
      "GrowthFactor": 25.0,  
      "FinalBakeTimeInMinutes": 0,  
      "State": "COMPLETE",  
      "PercentageComplete": 100.0,  
      "StartedAt": "2021-09-17T21:43:54.205000+00:00",  
      "CompletedAt": "2021-09-17T21:59:03.888000+00:00"  
    }  
  ]  
}
```

詳細については、AWS AppConfig 「[ユーザーガイド](#)」の「[ステップ 5: 設定のデプロイ](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListDeployments](#)」の「」を参照してください。

list-environments

次のコード例は、list-environments を使用する方法を示しています。

AWS CLI

使用可能な環境を一覧表示するには

次のlist-environments例では、指定されたアプリケーションの AWS アカウントで使用可能な環境を一覧表示します。

```
aws appconfig list-environments \  
  --application-id 339ohji
```

出力:

```
{  
  "Items": [  
    {  
      "ApplicationId": "339ohji",  
      "Id": "54j1r29",  
      "Name": "Example-Environment",  
      "State": "ReadyForDeployment"  
    }  
  ]  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 2: 環境の作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListEnvironments](#)」の「」を参照してください。

list-extension-associations

次の例は、list-extension-associations を使用する方法を説明しています。

AWS CLI

AWS リージョンの AWS アカウント内のすべての AWS AppConfig 拡張機能の関連付けを一覧表示するには

次のlist-extension-associations例では、特定の AWS リージョンの現在の AWS アカウントのすべての AWS AppConfig 拡張機能の関連付けを一覧表示します。

```
aws appconfig list-extension-associations \  
  --region us-west-2
```

出力:

```
{
  "Items": [
    {
      "Id": "a1b2c3d4",
      "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-
backup-extension/1",
      "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/
Finance"
    }
  ]
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListExtensionAssociations](#)」の「」を参照してください。

list-extensions

次の例は、list-extensions を使用する方法を説明しています。

AWS CLI

AWS リージョンの AWS アカウント内のすべての AWS AppConfig 拡張機能を一覧表示するには次のlist-extensions例では、特定の AWS リージョンの現在の AWS アカウントのすべての AWS AppConfig 拡張機能を一覧表示します。コマンドは、カスタム拡張子とオー AWS サリングされた拡張子を返します。

```
aws appconfig list-extensions \
  --region us-west-2
```

出力:

```
{
  "Items": [
    {
      "Id": "1A2B3C4D",
      "Name": "S3-backup-extension",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1"
    }
  ]
}
```

```
    },
    {
      "Id": "AWS.AppConfig.FeatureFlags",
      "Name": "AppConfig Feature Flags Helper",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.FeatureFlags/1",
      "Description": "Validates AppConfig feature flag data automatically
against a JSON schema that includes structure and constraints. Also transforms
feature flag data prior to sending to the client. This extension is automatically
associated to configuration profiles with type \"AWS.AppConfig.FeatureFlags\"."
    },
    {
      "Id": "AWS.AppConfig.JiraIntegration",
      "Name": "AppConfig integration with Atlassian Jira",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.JiraIntegration/1",
      "Description": "Exports feature flag data from AWS AppConfig into
Jira. The lifecycle of each feature flag in AppConfig is tracked in Jira as an
individual issue. Customers can see in Jira when flags are updated, turned on or
off. Works in conjunction with the AppConfig app in the Atlassian Marketplace and
is automatically associated to configuration profiles configured within that app."
    },
    {
      "Id": "AWS.AppConfig.DeploymentNotificationsToEventBridge",
      "Name": "AppConfig deployment events to Amazon EventBridge",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToEventBridge/1",
      "Description": "Sends events to Amazon EventBridge when a deployment
of configuration data in AppConfig is started, completed, or rolled back. Can
be associated to the following resources in AppConfig: Application, Environment,
Configuration Profile."
    },
    {
      "Id": "AWS.AppConfig.DeploymentNotificationsToSqs",
      "Name": "AppConfig deployment events to Amazon SQS",
      "VersionNumber": 1,
      "Arn": "arn:aws:appconfig:us-west-2::extension/
AWS.AppConfig.DeploymentNotificationsToSqs/1",
      "Description": "Sends messages to the configured Amazon SQS queue when
a deployment of configuration data in AppConfig is started, completed, or rolled
```

```

back. Can be associated to the following resources in AppConfig: Application,
Environment, Configuration Profile."
    },
    {
      "Id": "AWS.AppConfig.DeploymentNotificationsToSns",
      "Name": "AppConfig deployment events to Amazon SNS",
      "VersionNumber": 1,
      "Description": "Sends events to the configured Amazon SNS topic when
a deployment of configuration data in AppConfig is started, completed, or rolled
back. Can be associated to the following resources in AppConfig: Application,
Environment, Configuration Profile."
    }
  ]
}

```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListExtensions](#)」の「」を参照してください。

list-hosted-configuration-versions

次の例は、list-hosted-configuration-versions を使用する方法を説明しています。

AWS CLI

利用可能なホスト設定バージョンを一覧表示するには

次のlist-hosted-configuration-versions例では、指定されたアプリケーションと設定プロファイルの AWS AppConfig ホスト設定ストアでホストされている設定バージョンを一覧表示します。

```

aws appconfig list-hosted-configuration-versions \
  --application-id 339ohji \
  --configuration-profile-id ur8hx2f

```

出力:

```

{
  "Items": [
    {

```

```
    "ApplicationId": "339ohji",
    "ConfigurationProfileId": "ur8hx2f",
    "VersionNumber": 1,
    "ContentType": "application/json"
  }
]
```

詳細については、AWS AppConfig 「ユーザーガイド」の [AWS AppConfig 「ホストされた設定ストアについて」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListHostedConfigurationVersions](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

アプリケーションのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたアプリケーションのタグを一覧表示します。

```
aws appconfig list-tags-for-resource \
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji
```

出力:

```
{
  "Tags": {
    "group1": "1"
  }
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 1: AWS AppConfig アプリケーションの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListTagsForResource](#)」の「」を参照してください。

start-deployment

次の例は、start-deployment を使用する方法を説明しています。

AWS CLI

設定デプロイを開始するには

次のstart-deployment例では、指定された環境、デプロイ戦略、設定プロファイルを使用してアプリケーションへのデプロイを開始します。

```
aws appconfig start-deployment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-strategy-id 1225qzk \  
  --configuration-profile-id ur8hx2f \  
  --configuration-version 1
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "EnvironmentId": "54j1r29",  
  "DeploymentStrategyId": "1225qzk",  
  "ConfigurationProfileId": "ur8hx2f",  
  "DeploymentNumber": 1,  
  "ConfigurationName": "Example-Configuration-Profile",  
  "ConfigurationLocationUri": "ssm-parameter://Example-Parameter",  
  "ConfigurationVersion": "1",  
  "DeploymentDurationInMinutes": 15,  
  "GrowthType": "LINEAR",  
  "GrowthFactor": 25.0,  
  "FinalBakeTimeInMinutes": 0,  
  "State": "DEPLOYING",  
  "EventLog": [  
    {  
      "EventType": "DEPLOYMENT_STARTED",  
      "TriggeredBy": "USER",  
      "Description": "Deployment started",  
      "OccurredAt": "2021-09-17T21:43:54.205000+00:00"  
    }  
  ],  
  "PercentageComplete": 0.0,
```



```
"StartedAt": "2021-09-17T21:43:54.205000+00:00"
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 5: 設定のデプロイ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartDeployment](#)」の「」を参照してください。

stop-deployment

次の例は、stop-deployment を使用する方法を説明しています。

AWS CLI

設定のデプロイを停止するには

次のstop-deployment例では、指定された環境へのアプリケーション設定のデプロイを停止します。

```
aws appconfig stop-deployment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --deployment-number 2
```

出力:

```
{
  "DeploymentNumber": 0,
  "DeploymentDurationInMinutes": 0,
  "GrowthFactor": 0.0,
  "FinalBakeTimeInMinutes": 0,
  "PercentageComplete": 0.0
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 5: 設定のデプロイ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StopDeployment](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

アプリケーションにタグを付けるには

次のtag-resource例では、アプリケーションリソースにタグを付けます。

```
aws appconfig tag-resource \  
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \  
  --tags '{"group1" : "1"}'
```

このコマンドでは何も出力されません。

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 1: AWS AppConfig アプリケーションの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

アプリケーションからタグを削除するには

次のuntag-resource例では、指定されたアプリケーションから group1 タグを削除します。

```
aws appconfig untag-resource \  
  --resource-arn arn:aws:appconfig:us-east-1:111122223333:application/339ohji \  
  --tag-keys '["group1"]'
```

このコマンドでは何も出力されません。

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 1: AWS AppConfig アプリケーションの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-application

次のコード例は、update-application を使用する方法を示しています。

AWS CLI

アプリケーションを更新するには

次のupdate-application例では、指定されたアプリケーションの名前を更新します。

```
aws appconfig update-application \  
  --application-id 339ohji \  
  --name "Example-Application"
```

出力:

```
{  
  "Id": "339ohji",  
  "Name": "Example-Application",  
  "Description": "An application used for creating an example."  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の「[ステップ 1: AWS AppConfig アプリケーションの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateApplication](#)」の「」を参照してください。

update-configuration-profile

次のコード例は、update-configuration-profile を使用する方法を示しています。

AWS CLI

設定プロファイルを更新するには

次のupdate-configuration-profile例では、指定された設定プロファイルの説明を更新します。

```
aws appconfig update-configuration-profile \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f \  
  --description "Configuration profile used for examples."
```

出力:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "ur8hx2f",  
  "Name": "Example-Configuration-Profile",  
  "Description": "Configuration profile used for examples.",  
  "LocationUri": "ssm-parameter://Example-Parameter",  
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の [「ステップ 3: 設定と設定プロファイルの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateConfigurationProfile](#)」の「」を参照してください。

update-deployment-strategy

次のコード例は、update-deployment-strategy を使用する方法を示しています。

AWS CLI

デプロイ戦略を更新するには

次のupdate-deployment-strategy例では、指定されたデプロイ戦略の最終ベーク時間を 20 分に更新します。

```
aws appconfig update-deployment-strategy \  
  --deployment-strategy-id 1225qzk \  
  --final-bake-time-in-minutes 20
```

出力:

```
{  
  "Id": "1225qzk",
```

```
"Name": "Example-Deployment",
"DeploymentDurationInMinutes": 15,
"GrowthType": "LINEAR",
"GrowthFactor": 25.0,
"FinalBakeTimeInMinutes": 20,
"ReplicateTo": "SSM_DOCUMENT"
}
```

詳細については、「ユーザーガイド」の[「ステップ 4: デプロイ戦略の作成」](#)を参照してください。AWS AppConfig

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDeploymentStrategy](#)」の「」を参照してください。

update-environment

次の例は、update-environment を使用する方法を説明しています。

AWS CLI

環境を更新するには

次のupdate-environment例では、環境の説明を更新します。

```
aws appconfig update-environment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --description "An environment for examples."
```

出力:

```
{
  "ApplicationId": "339ohji",
  "Id": "54j1r29",
  "Name": "Example-Environment",
  "Description": "An environment for examples.",
  "State": "RolledBack"
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[「ステップ 2: 環境の作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateEnvironment](#)」の「」を参照してください。

update-extension-association

次のコード例は、update-extension-association を使用する方法を示しています。

AWS CLI

AWS AppConfig 拡張機能の関連付けを更新するには

次のupdate-extension-association例では、の拡張関連付けに新しいパラメータ値を追加します AWS AppConfig。

```
aws appconfig update-extension-association \  
  --region us-west-2 \  
  --extension-association-id a1b2c3d4 \  
  --parameters S3bucket=FinanceMobileApp
```

出力:

```
{  
  "Id": "a1b2c3d4",  
  "ExtensionArn": "arn:aws:appconfig:us-west-2:123456789012:extension/S3-backup-extension/1",  
  "ResourceArn": "arn:aws:appconfig:us-west-2:123456789012:application/Finance",  
  "Parameters": {  
    "S3bucket": "FinanceMobileApp"  
  },  
  "ExtensionVersionNumber": 1  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateExtensionAssociation](#)」の「」を参照してください。

update-extension

次のコード例は、update-extension を使用する方法を示しています。

AWS CLI

AWS AppConfig 拡張機能を更新するには

次のupdate-extension例では、 の拡張機能にパラメータキーを追加します AWS AppConfig。

```
aws appconfig update-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension \  
  --parameters S3bucket={Required=true}, CampaignID={Required=false}
```

出力:

```
{  
  "Id": "1A2B3C4D",  
  "Name": "S3-backup-extension",  
  "VersionNumber": 1,  
  "Arn": "arn:aws:appconfig:us-west-2:123456789012:extension/1A2B3C4D/1",  
  "Actions": {  
    "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [  
      {  
        "Name": "S3backup",  
        "Uri": "arn:aws:lambda:us-  
west-2:123456789012:function:S3backupfunction",  
        "RoleArn": "arn:aws:iam::123456789012:role/appconfigextensionrole"  
      }  
    ]  
  },  
  "Parameters": {  
    "CampaignID": {  
      "Required": false  
    },  
    "S3bucket": {  
      "Required": true  
    }  
  }  
}
```

詳細については、AWS AppConfig 「ユーザーガイド」の[AWS AppConfig 「拡張機能の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateExtension](#)」の「」を参照してください。

validate-configuration

次の例は、validate-configuration を使用する方法を説明しています。

AWS CLI

設定を検証するには

次のvalidate-configuration例では、設定プロファイルのバリデータを使用して設定を検証します。

```
aws appconfig validate-configuration \  
  --application-id abc1234 \  
  --configuration-profile-id ur8hx2f \  
  --configuration-version 1
```

コマンドは出力を生成しません。

詳細については、「ユーザーガイド」の[「ステップ 3: 設定と設定プロファイルの作成」](#)を参照してください。AWS AppConfig

- API 詳細については、AWS CLI 「コマンドリファレンス[ValidateConfiguration](#)」の「」を参照してください。

を使用した Application Auto Scaling の例 AWS CLI

次のコード例は、Application Auto Scaling AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-scaling-policy

次のコード例は、delete-scaling-policy を使用する方法を示しています。

AWS CLI

スケーリングポリシーを削除するには

この例では、デフォルトのクラスターで実行されている Amazon ECS サービスウェブアプリのスケーリングポリシーを削除します。

コマンド:

```
aws application-autoscaling delete-scaling-policy --policy-name web-app-cpu-lt-25 --scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app --service-namespace ecs
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteScalingPolicy](#)」の「」を参照してください。

delete-scheduled-action

次の例は、delete-scheduled-action を使用する方法を説明しています。

AWS CLI

スケジュールされたアクションを削除するには

次のdelete-scheduled-action例では、指定された Amazon AppStream 2.0 フリートから指定されたスケジュール済みアクションを削除します。

```
aws application-autoscaling delete-scheduled-action \
  --service-namespace appstream \
  --scalable-dimension appstream:fleet:DesiredCapacity \
  --resource-id fleet/sample-fleet \
  --scheduled-action-name my-recurring-action
```

このコマンドでは何も出力されません。

詳細については、「Application Auto Scaling ユーザーガイド」の「[スケジュールされたスケーリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteScheduledAction](#)」の「」を参照してください。

deregister-scalable-target

次の例は、deregister-scalable-target を使用する方法を説明しています。

AWS CLI

スケーラブルターゲットの登録を解除するには

この例では、デフォルトのクラスターで実行されているウェブアプリと呼ばれる Amazon ECS サービスのスケーラブルターゲットを登録解除します。

コマンド:

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

この例では、カスタムリソースのスケーラブルターゲットを登録解除します。.txt ファイルには、リソース ID を識別する文字列が含まれています custom-resource-id。これは、カスタムリソースの場合、Amazon API Gateway エンドポイントを介したカスタムリソースへのパスです。

コマンド:

```
aws application-autoscaling deregister-scalable-target --service-namespace custom-resource --scalable-dimension custom-resource:ResourceType:Property --resource-id file://~/custom-resource-id.txt
```

custom-resource-id.txt ファイルの内容 :

```
https://example.execute-api.us-west-2.amazonaws.com/prod/scalableTargetDimensions/1-23456789
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterScalableTarget](#)」の「」を参照してください。

describe-scalable-targets

次のコード例は、describe-scalable-targets を使用する方法を示しています。

AWS CLI

スケーラブルターゲットを記述するには

次のdescribe-scalable-targets例では、ecsサービス名前空間のスケーラブルターゲットについて説明します。

```
aws application-autoscaling describe-scalable-targets \  
  --service-namespace ecs
```

出力:

```
{  
  "ScalableTargets": [  
    {  
      "ServiceNamespace": "ecs",  
      "ScalableDimension": "ecs:service:DesiredCount",  
      "ResourceId": "service/default/web-app",  
      "MinCapacity": 1,  
      "MaxCapacity": 10,  
      "RoleARN": "arn:aws:iam::123456789012:role/  
aws-service-role/ecs.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_ECSService",  
      "CreationTime": 1462558906.199,  
      "SuspendedState": {  
        "DynamicScalingOutSuspended": false,  
        "ScheduledScalingSuspended": false,  
        "DynamicScalingInSuspended": false  
      },  
      "ScalableTargetARN": "arn:aws:application-autoscaling:us-  
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"  
    }  
  ]  
}
```

詳細については、[AWS 「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling で使用できるサービス」](#)を参照してください。 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeScalableTargets](#)」の「」を参照してください。

describe-scaling-activities

次の例は、describe-scaling-activities を使用する方法を説明しています。

AWS CLI

例 1: 指定された Amazon ECSサービスのスケーリングアクティビティを記述するには

次のdescribe-scaling-activities例では、defaultクラスターで実行web-appされているという Amazon ECSサービスのスケーリングアクティビティについて説明します。出力には、スケーリングポリシーによって開始されたスケーリングアクティビティが表示されます。

```
aws application-autoscaling describe-scaling-activities \  
  --service-namespace ecs \  
  --resource-id service/default/web-app
```

出力:

```
{  
  "ScalingActivities": [  
    {  
      "ScalableDimension": "ecs:service:DesiredCount",  
      "Description": "Setting desired count to 1.",  
      "ResourceId": "service/default/web-app",  
      "ActivityId": "e6c5f7d1-dbbb-4a3f-89b2-51f33e766399",  
      "StartTime": 1462575838.171,  
      "ServiceNamespace": "ecs",  
      "EndTime": 1462575872.111,  
      "Cause": "monitor alarm web-app-cpu-lt-25 in state ALARM triggered  
policy web-app-cpu-lt-25",  
      "StatusMessage": "Successfully set desired count to 1. Change  
successfully fulfilled by ecs.",  
      "StatusCode": "Successful"  
    }  
  ]  
}
```

詳細については、[「Application Auto Scaling ユーザーガイド」](#)の「[Application Auto Scaling のスケーリングアクティビティ](#)」を参照してください。 Auto Scaling

例 2: 指定された DynamoDB テーブルのスケーリングアクティビティを記述するには

次のdescribe-scaling-activities例では、と呼ばれる DynamoDB テーブルのスケーリングアクティビティについて説明しますTestTable。出力には、2つの異なるスケジュールされたアクションによって開始されたスケーリングアクティビティが表示されます。

```
aws application-autoscaling describe-scaling-activities \  
  --service-namespace dynamodb \  
  --resource-id table/TestTable
```

出力:

```
{  
  "ScalingActivities": [  
    {  
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",  
      "Description": "Setting write capacity units to 10.",  
      "ResourceId": "table/my-table",  
      "ActivityId": "4d1308c0-bbcf-4514-a673-b0220ae38547",  
      "StartTime": 1561574415.086,  
      "ServiceNamespace": "dynamodb",  
      "EndTime": 1561574449.51,  
      "Cause": "maximum capacity was set to 10",  
      "StatusMessage": "Successfully set write capacity units to 10. Change  
successfully fulfilled by dynamodb.",  
      "StatusCode": "Successful"  
    },  
    {  
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",  
      "Description": "Setting min capacity to 5 and max capacity to 10",  
      "ResourceId": "table/my-table",  
      "ActivityId": "f2b7847b-721d-4e01-8ef0-0c8d3bacc1c7",  
      "StartTime": 1561574414.644,  
      "ServiceNamespace": "dynamodb",  
      "Cause": "scheduled action name my-second-scheduled-action was  
triggered",  
      "StatusMessage": "Successfully set min capacity to 5 and max capacity to  
10",  
      "StatusCode": "Successful"  
    },  
    {  
      "ScalableDimension": "dynamodb:table:WriteCapacityUnits",  
      "Description": "Setting write capacity units to 15.",
```

```
    "ResourceId": "table/my-table",
    "ActivityId": "d8ea4de6-9eaa-499f-b466-2cc5e681ba8b",
    "StartTime": 1561574108.904,
    "ServiceNamespace": "dynamodb",
    "EndTime": 1561574140.255,
    "Cause": "minimum capacity was set to 15",
    "StatusMessage": "Successfully set write capacity units to 15. Change
successfully fulfilled by dynamodb.",
    "StatusCode": "Successful"
  },
  {
    "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
    "Description": "Setting min capacity to 15 and max capacity to 20",
    "ResourceId": "table/my-table",
    "ActivityId": "3250fd06-6940-4e8e-bb1f-d494db7554d2",
    "StartTime": 1561574108.512,
    "ServiceNamespace": "dynamodb",
    "Cause": "scheduled action name my-first-scheduled-action was
triggered",
    "StatusMessage": "Successfully set min capacity to 15 and max capacity
to 20",
    "StatusCode": "Successful"
  }
]
}
```

詳細については、[「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling のスケーリングアクティビティ」](#)を参照してください。 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeScalingActivities](#)」の「」を参照してください。

describe-scaling-policies

次のコード例は、describe-scaling-policies を使用する方法を示しています。

AWS CLI

スケーリングポリシーを記述するには

この例では、ECS サービス名前空間のスケーリングポリシーについて説明します。

コマンド:

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

出力:

```
{
  "ScalingPolicies": [
    {
      "PolicyName": "web-app-cpu-gt-75",
      "ScalableDimension": "ecs:service:DesiredCount",
      "ResourceId": "service/default/web-app",
      "CreationTime": 1462561899.23,
      "StepScalingPolicyConfiguration": {
        "Cooldown": 60,
        "StepAdjustments": [
          {
            "ScalingAdjustment": 200,
            "MetricIntervalLowerBound": 0.0
          }
        ],
        "AdjustmentType": "PercentChangeInCapacity"
      },
      "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-gt-75",
      "PolicyType": "StepScaling",
      "Alarms": [
        {
          "AlarmName": "web-app-cpu-gt-75",
          "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-gt-75"
        }
      ],
      "ServiceNamespace": "ecs"
    },
    {
      "PolicyName": "web-app-cpu-lt-25",
      "ScalableDimension": "ecs:service:DesiredCount",
      "ResourceId": "service/default/web-app",
      "CreationTime": 1462562575.099,
      "StepScalingPolicyConfiguration": {
        "Cooldown": 1,
        "StepAdjustments": [
          {
```

```

        "ScalingAdjustment": -50,
        "MetricIntervalUpperBound": 0.0
      }
    ],
    "AdjustmentType": "PercentChangeInCapacity"
  },
  "PolicyARN": "arn:aws:autoscaling:us-
west-2:012345678910:scalingPolicy:6d8972f3-efc8-437c-92d1-6270f29a66e7:resource/ecs/
service/default/web-app:policyName/web-app-cpu-lt-25",
  "PolicyType": "StepScaling",
  "Alarms": [
    {
      "AlarmName": "web-app-cpu-lt-25",
      "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-lt-25"
    }
  ],
  "ServiceNamespace": "ecs"
}
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeScalingPolicies](#)」の「」を参照してください。

describe-scheduled-actions

次のコード例は、describe-scheduled-actions を使用する方法を示しています。

AWS CLI

スケジュールされたアクションを記述するには

次のdescribe-scheduled-actions例では、指定されたサービス名前空間のスケジュールされたアクションの詳細を表示します。

```
aws application-autoscaling describe-scheduled-actions \
  --service-namespace dynamodb
```

出力:

```
{
```



```

    "ScheduledActions": [
      {
        "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
        "Schedule": "at(2019-05-20T18:35:00)",
        "ResourceId": "table/my-table",
        "CreationTime": 1561571888.361,
        "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-first-scheduled-action",
        "ScalableTargetAction": {
          "MinCapacity": 15,
          "MaxCapacity": 20
        },
        "ScheduledActionName": "my-first-scheduled-action",
        "ServiceNamespace": "dynamodb"
      },
      {
        "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
        "Schedule": "at(2019-05-20T18:40:00)",
        "ResourceId": "table/my-table",
        "CreationTime": 1561571946.021,
        "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledAction:2d36aa3b-cdf9-4565-b290-81db519b227d:resource/
dynamodb/table/my-table:scheduledActionName/my-second-scheduled-action",
        "ScalableTargetAction": {
          "MinCapacity": 5,
          "MaxCapacity": 10
        },
        "ScheduledActionName": "my-second-scheduled-action",
        "ServiceNamespace": "dynamodb"
      }
    ]
  }
}

```

詳細については、「Application Auto Scaling ユーザーガイド」の「[スケジュールされたスケールリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeScheduledActions](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

スケーラブルターゲットのタグを一覧表示するには

次のlist-tags-for-resource例では、で指定されたスケーラブルターゲットにアタッチされているタグキーの名前と値を一覧表示しますARN。

```
aws application-autoscaling list-tags-for-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123
```

出力:

```
{  
  "Tags": {  
    "environment": "production"  
  }  
}
```

詳細については、[「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling のタグ付けサポート」](#)を参照してください。 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-scaling-policy

次のコード例は、put-scaling-policy を使用する方法を示しています。

AWS CLI

例 1: 事前定義されたメトリクス指定を使用してターゲット追跡スケーリングポリシーを適用するには

次のput-scaling-policy例では、デフォルトのクラスターのウェブアプリと呼ばれる Amazon ECS サービスに、事前定義されたメトリクス仕様を持つターゲット追跡スケーリングポリシーを適用します。このポリシーは、サービスの平均CPU使用率を 75% に保ち、スケールアウトおよびスケールインのクールダウン期間は 60 秒です。出力には、ユーザーに代わって作成された 2 つの CloudWatch アラームの ARNs と名前が含まれます。

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \  

```

```
--scalable-dimension ecs:service:DesiredCount \  
--resource-id service/default/web-app \  
--policy-name cpu75-target-tracking-scaling-policy --policy-  
type TargetTrackingScaling \  
--target-tracking-scaling-policy-configuration file://config.json
```

この例では、現在のディレクトリに次の内容の config.json ファイルがあることを前提としています。

```
{  
  "TargetValue": 75.0,  
  "PredefinedMetricSpecification": {  
    "PredefinedMetricType": "ECSServiceAverageCPUUtilization"  
  },  
  "ScaleOutCooldown": 60,  
  "ScaleInCooldown": 60  
}
```

出力:

```
{  
  "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:6d8972f3-  
efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/cpu75-  
target-tracking-scaling-policy",  
  "Alarms": [  
    {  
      "AlarmARN": "arn:aws:cloudwatch:us-  
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-  
b46e-434a-a60f-3b36d653feca",  
      "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-  
b46e-434a-a60f-3b36d653feca"  
    },  
    {  
      "AlarmARN": "arn:aws:cloudwatch:us-  
west-2:012345678910:alarm:TargetTracking-service/default/web-app-AlarmLow-1b437334-  
d19b-4a63-a812-6c67aaf2910d",  
      "AlarmName": "TargetTracking-service/default/web-app-AlarmLow-1b437334-  
d19b-4a63-a812-6c67aaf2910d"  
    }  
  ]  
}
```

例 2: カスタマイズされたメトリクス仕様を使用してターゲット追跡スケーリングポリシーを適用するには

次のput-scaling-policy例では、カスタマイズされたメトリクス仕様を持つターゲット追跡スケーリングポリシーを、デフォルトクラスターのウェブアプリと呼ばれる Amazon ECSサービスに適用します。このポリシーは、サービスの平均使用率を 75% に維持し、スケールアウトおよびスケールインのクールダウン期間は 60 秒です。出力には、ユーザーに代わって作成された 2 つの CloudWatch アラームの ARNs と名前が含まれます。

```
aws application-autoscaling put-scaling-policy --service-namespace ecs \  
--scalable-dimension ecs:service:DesiredCount \  
--resource-id service/default/web-app \  
--policy-name cms75-target-tracking-scaling-policy \  
--policy-type TargetTrackingScaling \  
--target-tracking-scaling-policy-configuration file://config.json
```

この例では、現在のディレクトリに次の内容の config.json ファイルがあることを前提としています。

```
{  
  "TargetValue":75.0,  
  "CustomizedMetricSpecification":{  
    "MetricName":"MyUtilizationMetric",  
    "Namespace":"MyNamespace",  
    "Dimensions": [  
      {  
        "Name":"MyOptionalMetricDimensionName",  
        "Value":"MyOptionalMetricDimensionValue"  
      }  
    ],  
    "Statistic":"Average",  
    "Unit":"Percent"  
  },  
  "ScaleOutCooldown": 60,  
  "ScaleInCooldown": 60  
}
```

出力:

```
{
```

```

    "PolicyARN": "arn:aws:autoscaling:us-west-2:012345678910:scalingPolicy:
8784a896-b2ba-47a1-b08c-27301cc499a1:resource/ecs/service/default/web-
app:policyName/cms75-target-tracking-scaling-policy",
    "Alarms": [
      {
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0",
        "AlarmName": "TargetTracking-service/default/web-app-
AlarmHigh-9bc77b56-0571-4276-ba0f-d4178882e0a0"
      },
      {
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4",
        "AlarmName": "TargetTracking-service/default/web-app-
AlarmLow-9b6ad934-6d37-438e-9e05-02836ddcbdc4"
      }
    ]
  }
}

```

例 3: スケールアウトにのみターゲット追跡スケーリングポリシーを適用するには

次のput-scaling-policy例では、ターゲット追跡スケーリングポリシーをデフォルトのクラスターweb-appでと呼ばれる Amazon ECSサービスに適用します。このポリシーは、Application Load Balancer のRequestCountPerTargetメトリクスがしきい値を超えたときにECSサービスをスケールアウトするために使用されます。出力には、ユーザーに代わって作成された CloudWatch アラームの ARNと名前が含まれます。

```

aws application-autoscaling put-scaling-policy \
  --service-namespace ecs \
  --scalable-dimension ecs:service:DesiredCount \
  --resource-id service/default/web-app \
  --policy-name alb-scale-out-target-tracking-scaling-policy \
  --policy-type TargetTrackingScaling \
  --target-tracking-scaling-policy-configuration file://config.json

```

config.json の内容:

```

{
  "TargetValue": 1000.0,
  "PredefinedMetricSpecification": {

```

```

    "PredefinedMetricType": "ALBRequestCountPerTarget",
    "ResourceLabel": "app/EC2Co-EcsE1-1TKLTMITMM0E0/f37c06a68c1748aa/
targetgroup/EC2Co-Defau-LDNM7Q3ZH1ZN/6d4ea56ca2d6a18d"
  },
  "ScaleOutCooldown": 60,
  "ScaleInCooldown": 60,
  "DisableScaleIn": true
}

```

出力:

```

{
  "PolicyARN": "arn:aws:autoscaling:us-west-2:123456789012:scalingPolicy:6d8972f3-
efc8-437c-92d1-6270f29a66e7:resource/ecs/service/default/web-app:policyName/alb-
scale-out-target-tracking-scaling-policy",
  "Alarms": [
    {
      "AlarmName": "TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca",
      "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-service/default/web-app-AlarmHigh-d4f0770c-
b46e-434a-a60f-3b36d653feca"
    }
  ]
}

```

詳細については、[「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling のターゲット追跡スケーリングポリシー」](#)を参照してください。AWS Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[PutScalingPolicy](#)」の「」を参照してください。

put-scheduled-action

次のコード例は、put-scheduled-action を使用する方法を示しています。

AWS CLI

スケジュールされたアクションを DynamoDB テーブルに追加するには

この例では、定期的なスケジュールでスケールアウト TestTable するために という DynamoDB テーブルにスケジュールされたアクションを追加します。指定されたスケジュール (毎日

12:15pm UTC) で、現在の容量が に指定された値を下回ると MinCapacity、Application Auto Scaling は に指定された値にスケールアウトします MinCapacity。

コマンド:

```
aws application-autoscaling put-scheduled-action --service-namespace dynamodb
--scheduled-action-name my-recurring-action --schedule "cron(15 12 * * ? *)" --
resource-id table/TestTable --scalable-dimension dynamodb:table:WriteCapacityUnits
--scalable-target-action MinCapacity=6
```

詳細については、「Application Auto Scaling ユーザーガイド」の「スケジュールされたスケールリング」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutScheduledAction](#)」の「」を参照してください。

register-scalable-target

次のコード例は、register-scalable-target を使用する方法を示しています。

AWS CLI

例 1: ECSサービスをスケーラブルターゲットとして登録するには

次のregister-scalable-target例では、Amazon ECSサービスを Application Auto Scaling に登録します。また、スケーラブルターゲットproductionにキー名environmentと値を含むタグを追加します。

```
aws application-autoscaling register-scalable-target \
--service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id service/default/web-app \
--min-capacity 1 --max-capacity 10 \
--tags environment=production
```

出力:

```
{
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

その他の AWS サービスやカスタムリソースの例については、[AWS Application Auto Scaling ユーザーガイドの「Application Auto Scaling で使用できるサービスのトピック」](#)を参照してください。 Auto Scaling

例 2: スケーラブルターゲットのスケールリングアクティビティを停止するには

次のregister-scalable-target例では、既存のスケラブルターゲットのスケールリングアクティビティを一時停止します。

```
aws application-autoscaling register-scalable-target \  
  --service-namespace dynamodb \  
  --scalable-dimension dynamodb:table:ReadCapacityUnits \  
  --resource-id table/my-table \  
  --suspended-  
state DynamicScalingInSuspended=true,DynamicScalingOutSuspended=true,ScheduledScalingSuspended=true
```

出力:

```
{  
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-  
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"  
}
```

詳細については、[「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling のスケールリングの一時停止と再開」](#)を参照してください。 Auto Scaling

例 3: スケーラブルターゲットのスケールリングアクティビティを再開するには

次のregister-scalable-target例では、既存のスケラブルターゲットのスケールリングアクティビティを再開します。

```
aws application-autoscaling register-scalable-target \  
  --service-namespace dynamodb \  
  --scalable-dimension dynamodb:table:ReadCapacityUnits \  
  --resource-id table/my-table \  
  --suspended-  
state DynamicScalingInSuspended=false,DynamicScalingOutSuspended=false,ScheduledScalingSuspended=false
```

出力:

```
{
```



```
"ScalableTargetARN": "arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

詳細については、[「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling のスケールリングの一時停止と再開」](#)を参照してください。 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterScalableTarget](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

スケーラブルターゲットにタグを追加するには

次のtag-resource例では、キー名environmentと値を含むタグproductionを、で指定されたスケーラブルターゲットに追加しますARN。

```
aws application-autoscaling tag-resource \
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123 \
  --tags environment=production
```

このコマンドでは何も出力されません。

詳細については、[「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling のタグ付けサポート」](#)を参照してください。 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

スケーラブルターゲットからタグを削除するには

次の`untag-resource`例では、 によって指定されたスケラブルターゲット`environment`からキー名を持つタグペアを削除しますARN。

```
aws application-autoscaling untag-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123 \  
  --tag-keys "environment"
```

このコマンドでは何も出力されません。

詳細については、[「Application Auto Scaling ユーザーガイド」の「Application Auto Scaling のタグ付けサポート」](#)を参照してください。 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

を使用した Application Discovery Service の例 AWS CLI

次のコード例は、Application Discovery Service AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

`describe-agents`

次の例は、`describe-agents` を使用する方法を説明しています。

AWS CLI

指定された `collectionStatus` 状態のエージェントを記述する

このコマンド例では、コレクションステータスがSTARTED「」またはSTOPPED「」のコレクションエージェントについて説明します。

コマンド:

```
aws discovery describe-agents --filters
  name="collectionStatus",values="STARTED","STOPPED",condition="EQUALS" --max-
  results 3
```

出力:

```
{
  "Snapshots": [
    {
      "version": "1.0.40.0",
      "agentType": "EC2",
      "hostName": "ip-172-31-40-234",
      "collectionStatus": "STOPPED",
      "agentNetworkInfoList": [
        {
          "macAddress": "06:b5:97:14:fc:0d",
          "ipAddress": "172.31.40.234"
        }
      ],
      "health": "UNKNOWN",
      "agentId": "i-003305c02a776e883",
      "registeredTime": "2016-12-09T19:05:06Z",
      "lastHealthPingTime": "2016-12-09T19:05:10Z"
    },
    {
      "version": "1.0.40.0",
      "agentType": "EC2",
      "hostName": "ip-172-31-39-64",
      "collectionStatus": "STARTED",
      "agentNetworkInfoList": [
        {
          "macAddress": "06:a1:0e:c7:b2:73",
          "ipAddress": "172.31.39.64"
        }
      ],
      "health": "SHUTDOWN",
      "agentId": "i-003a5e5e2b36cf8bd",
      "registeredTime": "2016-11-16T16:36:25Z",
```

```
        "lastHealthPingTime": "2016-11-16T16:47:37Z"
      }
    ]
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAgents](#)」の「」を参照してください。

describe-configurations

次の例は、describe-configurations を使用する方法を説明しています。

AWS CLI

選択したアセット設定の説明

この例では、2 つの指定されたサーバーの設定について説明します。アクションは、設定 ID からアセットのタイプを検出します。コマンドごとに許可されるアセットのタイプは 1 つだけです。

コマンド:

```
aws discovery describe-configurations --configuration-ids "d-  
server-099385097ef9fbcfb" "d-server-0c4f2dd1fee22c6c1"
```

出力:

```
{
  "configurations": [
    {
      "server.performance.maxCpuUsagePct": "0.0",
      "server.performance.maxDiskReadIOPS": "0.0",
      "server.performance.avgCpuUsagePct": "0.0",
      "server.type": "EC2",
      "server.performance.maxNetworkReadsPerSecondInKB": "0.19140625",
      "server.hostName": "ip-172-31-35-152",
      "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
      "server.tags.hasMoreValues": "false",
      "server.performance.minFreeRAMInKB": "1543496.0",
      "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
      "server.performance.maxDiskReadsPerSecondInKB": "0.0",
      "server.applications": "[]",
      "server.performance.numDisks": "1",
```

```

        "server.performance.numCpus": "1",
        "server.performance.numCores": "1",
        "server.performance.maxDiskWriteIOPS": "0.0",
        "server.performance.maxNetworkWritesPerSecondInKB": "0.82421875",
        "server.performance.avgDiskWritesPerSecondInKB": "0.0",
        "server.networkInterfaceInfo": "[{\"name\":\"eth0\",
\\\"macAddress\\\":\\\"06:A7:7D:3F:54:57\\\",\\\"ipAddress\\\":\\\"172.31.35.152\\\",\\\"netMask\\\":
\\\"255.255.240.0\\\"},{\"name\":\"lo\\\",\\\"macAddress\\\":\\\"00:00:00:00:00:00\\\",\\\"ipAddress
\\\":\\\"127.0.0.1\\\",\\\"netMask\\\":\\\"255.0.0.0\\\"},{\"name\":\"eth0\\\",\\\"macAddress\\\":
\\\"06:A7:7D:3F:54:57\\\",\\\"ipAddress\\\":\\\"fe80::4a7:7dff:fe3f:5457\\\"},{\"name\":\"lo\\\",
\\\"macAddress\\\":\\\"00:00:00:00:00:00\\\",\\\"ipAddress\\\":\\\"::1\\\"}]",
        "server.performance.avgNetworkReadsPerSecondInKB":
"0.049153645833333333",
        "server.tags": "[]",
        "server.applications.hasMoreValues": "false",
        "server.timeOfCreation": "2016-10-28 23:44:00.0",
        "server.agentId": "i-4447bc1b",
        "server.performance.maxDiskWritesPerSecondInKB": "0.0",
        "server.performance.avgDiskReadIOPS": "0.0",
        "server.performance.avgFreeRAMInKB": "1547210.1333333333",
        "server.performance.avgDiskReadsPerSecondInKB": "0.0",
        "server.performance.avgDiskWriteIOPS": "0.0",
        "server.performance.numNetworkCards": "2",
        "server.hypervisor": "xen",
        "server.networkInterfaceInfo.hasMoreValues": "false",
        "server.performance.avgNetworkWritesPerSecondInKB": "0.1380859375",
        "server.osName": "Linux - Amazon Linux AMI release 2015.03",
        "server.performance.totalRAMInKB": "1694732.0",
        "server.cpuType": "x64"
    },
    {
        "server.performance.maxCpuUsagePct": "100.0",
        "server.performance.maxDiskReadIOPS": "0.0",
        "server.performance.avgCpuUsagePct": "14.733333333333333",
        "server.type": "EC2",
        "server.performance.maxNetworkReadsPerSecondInKB": "13.400390625",
        "server.hostName": "ip-172-31-42-208",
        "server.configurationId": "d-server-099385097ef9fbcfb",
        "server.tags.hasMoreValues": "false",
        "server.performance.minFreeRAMInKB": "1531104.0",
        "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
        "server.performance.maxDiskReadsPerSecondInKB": "0.0",
        "server.applications": "[]",
        "server.performance.numDisks": "1",

```

```

        "server.performance.numCpus": "1",
        "server.performance.numCores": "1",
        "server.performance.maxDiskWriteIOPS": "1.0",
        "server.performance.maxNetworkWritesPerSecondInKB": "12.271484375",
        "server.performance.avgDiskWritesPerSecondInKB":
"0.5333333333333334",
        "server.networkInterfaceInfo": "[{"name":"eth0",
\\"macAddress\\":\\"06:4A:79:60:75:61\\",\\"ipAddress\\":\\"172.31.42.208\\",\\"netMask
\\":\\"255.255.240.0\\"}, {"name":"eth0",\\"macAddress\\":\\"06:4A:79:60:75:61\\",
\\"ipAddress\\":\\"fe80::44a:79ff:fe60:7561\\"}, {"name":"lo",\\"macAddress\\":
\\"00:00:00:00:00:00\\",\\"ipAddress\\":\\":::1\\"}, {"name":"lo",\\"macAddress\\":
\\"00:00:00:00:00:00\\",\\"ipAddress\\":\\"127.0.0.1\\",\\"netMask\\":\\"255.0.0.0\\"}]",
        "server.performance.avgNetworkReadsPerSecondInKB":
"2.8720052083333334",
        "server.tags": "[]",
        "server.applications.hasMoreValues": "false",
        "server.timeOfCreation": "2016-10-28 23:44:30.0",
        "server.agentId": "i-c142b99e",
        "server.performance.maxDiskWritesPerSecondInKB": "4.0",
        "server.performance.avgDiskReadIOPS": "0.0",
        "server.performance.avgFreeRAMInKB": "1534946.4",
        "server.performance.avgDiskReadsPerSecondInKB": "0.0",
        "server.performance.avgDiskWriteIOPS": "0.13333333333333336",
        "server.performance.numNetworkCards": "2",
        "server.hypervisor": "xen",
        "server.networkInterfaceInfo.hasMoreValues": "false",
        "server.performance.avgNetworkWritesPerSecondInKB":
"1.7977864583333332",
        "server.osName": "Linux - Amazon Linux AMI release 2015.03",
        "server.performance.totalRAMInKB": "1694732.0",
        "server.cpuType": "x64"
    }
]
}

```

選択したアセット設定の説明

このコマンド例では、2つの指定されたアプリケーションの設定について説明します。アクションは、設定 ID からアセットのタイプを検出します。コマンドごとに許可されるアセットのタイプは 1 つだけです。

コマンド:

```
aws discovery describe-configurations --configuration-ids "d-  
application-0ac39bc0e4fad0e42" "d-application-02444a45288013764q"
```

出力:

```
{  
  "configurations": [  
    {  
      "application.serverCount": "0",  
      "application.name": "Application-12345",  
      "application.lastModifiedTime": "2016-12-13 23:53:27.0",  
      "application.description": "",  
      "application.timeOfCreation": "2016-12-13 23:53:27.0",  
      "application.configurationId": "d-application-0ac39bc0e4fad0e42"  
    },  
    {  
      "application.serverCount": "0",  
      "application.name": "Application-67890",  
      "application.lastModifiedTime": "2016-12-13 23:53:33.0",  
      "application.description": "",  
      "application.timeOfCreation": "2016-12-13 23:53:33.0",  
      "application.configurationId": "d-application-02444a45288013764"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeConfigurations](#)」の「」を参照してください。

list-configurations

次のコード例は、list-configurations を使用する方法を示しています。

AWS CLI

一連のフィルター条件を満たす検出されたすべてのサーバーを一覧表示するには

このコマンド例では、Ubuntu を実行していない 2 つのホスト名パターンのいずれかに一致する検出されたサーバーを一覧表示します。

コマンド:

```
aws discovery list-configurations --configuration-type SERVER --filters
name="server.hostName",values="172-31-35","172-31-42",condition="CONTAINS"
name="server.osName",values="Ubuntu",condition="NOT_CONTAINS"
```

出力:

```
{
  "configurations": [
    {
      "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
      "server.type": "EC2",
      "server.hostName": "ip-172-31-42-208",
      "server.timeOfCreation": "2016-10-28 23:44:30.0",
      "server.configurationId": "d-server-099385097ef9fbcfb",
      "server.osName": "Linux - Amazon Linux AMI release 2015.03",
      "server.agentId": "i-c142b99e"
    },
    {
      "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
      "server.type": "EC2",
      "server.hostName": "ip-172-31-35-152",
      "server.timeOfCreation": "2016-10-28 23:44:00.0",
      "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
      "server.osName": "Linux - Amazon Linux AMI release 2015.03",
      "server.agentId": "i-4447bc1b"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListConfigurations](#)」の「」を参照してください。

AppRegistry を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AppRegistry。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-attribute-group

次の例は、associate-attribute-group を使用する方法を説明しています。

AWS CLI

属性グループを関連付けるには

次のassociate-attribute-group例では、AWS アカウント内の特定の属性グループを AWS アカウント内の特定のアプリケーションに関連付けます。

```
aws servicecatalog-appregistry associate-attribute-group \  
  --application "ExampleApplication" \  
  --attribute-group "ExampleAttributeGroup"
```

出力:

```
{  
  "applicationArn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
  "attributeGroupArn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1"  
}
```

詳細については、[「Service Catalog 管理者ガイド」の「属性グループの関連付けと関連付け解除」](#)を参照してください。AWS AppRegistry

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateAttributeGroup](#)」の「」を参照してください。

create-application

次のコード例は、create-application を使用する方法を示しています。

AWS CLI

アプリケーションを作成するには

次のcreate-application例では、AWS アカウントに新しいアプリケーションを作成します。

```
aws servicecatalog-appregistry create-application \  
  --name "ExampleApplication"
```

出力:

```
{  
  "application": {  
    "id": "0ars38r6btoohvpvd9gqrptt91",  
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
    "name": "ExampleApplication",  
    "creationTime": "2023-02-28T21:10:10.820000+00:00",  
    "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",  
    "tags": {}  
  }  
}
```

詳細については、「Service Catalog 管理者ガイド」の「[アプリケーションの作成](#)」を参照してください。AWS AppRegistry

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateApplication](#)」の「」を参照してください。

create-attribute-group

次のコード例は、create-attribute-group を使用する方法を示しています。

AWS CLI

属性グループを作成するには

次のcreate-attribute-group例では、AWS アカウントに新しい属性グループを作成します。

```
aws servicecatalog-appregistry create-attribute-group \  
  --name "ExampleAttributeGroup"
```

```
--name "ExampleAttributeGroup" \  
--attributes '{"SomeKey1":"SomeValue1","SomeKey2":"SomeValue2"}'
```

出力:

```
{  
  "attributeGroup": {  
    "id": "01sj5xdwhbw54kejwnt09fnpc1",  
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1",  
    "name": "ExampleAttributeGroup",  
    "creationTime": "2023-02-28T20:38:01.389000+00:00",  
    "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",  
    "tags": {}  
  }  
}
```

詳細については、「Service Catalog 管理者ガイド」の「[属性グループの作成](#)」を参照してください。AWS AppRegistry

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAttributeGroup](#)」の「」を参照してください。

delete-application

次のコード例は、delete-application を使用する方法を示しています。

AWS CLI

アプリケーションを削除するには

次のdelete-application例では、AWS アカウント内の特定のアプリケーションを削除します。

```
aws servicecatalog-appregistry delete-application \  
--application "ExampleApplication3"
```

出力:

```
{
```

```
"application": {
  "id": "055gw7aynr1i5mbv7kjwzx5945",
  "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwzx5945",
  "name": "ExampleApplication3",
  "creationTime": "2023-02-28T22:06:28.228000+00:00",
  "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
}
}
```

詳細については、「Service Catalog 管理者ガイド」の「[アプリケーションの削除](#)」を参照してください。AWS AppRegistry

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApplication](#)」の「」を参照してください。

delete-attribute-group

次の例は、delete-attribute-group を使用する方法を説明しています。

AWS CLI

例 8: 属性グループを削除するには

次のdelete-attribute-group例では、AWS アカウント内の特定の属性グループを削除します。

```
aws servicecatalog-appregistry delete-attribute-group \
  --attribute-group ExampleAttributeGroup3
```

出力:

```
{
  "attributeGroup": {
    "id": "011ge6y3emyjijt8dw8jn6r0hv",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
    "name": "ExampleAttributeGroup3",
    "creationTime": "2023-02-28T22:05:35.224000+00:00",
    "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
  }
}
```

```
}
```

詳細については、「Service Catalog 管理者ガイド」の「[属性グループの削除](#)」を参照してください。AWS AppRegistry

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAttributeGroup](#)」の「」を参照してください。

get-application

次の例は、get-application を使用する方法を説明しています。

AWS CLI

アプリケーションを取得するには

次のget-application例では、AWS アカウント内の特定のアプリケーションに関するメタデータ情報を取得します。

```
aws servicecatalog-appregistry get-application \  
--application "ExampleApplication"
```

出力:

```
{  
  "id": "0ars38r6btoohvpvd9gqrptt91",  
  "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
  "name": "ExampleApplication",  
  "creationTime": "2023-02-28T21:10:10.820000+00:00",  
  "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",  
  "associatedResourceCount": 0,  
  "tags": {  
    "aws:servicecatalog:applicationName": "ExampleApplication"  
  },  
  "integrations": {  
    "resourceGroup": {  
      "state": "CREATE_COMPLETE",  
      "arn": "arn:aws:resource-groups:us-west-2:813737243517:group/  
AWS_AppRegistry_Application-ExampleApplication"  
    }  
  }  
}
```

```
}
```

詳細については、AWS 「Service Catalog AppRegistry 管理者ガイド」の [「アプリケーションの詳細の使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetApplication](#)」の「」を参照してください。

get-attribute-group

次の例は、get-attribute-group を使用する方法を説明しています。

AWS CLI

属性グループを取得するには

次のget-attribute-group例では、AWS アカウント内の特定の属性グループを取得します。

```
aws servicecatalog-appregistry get-attribute-group \  
  --attribute-group "ExampleAttributeGroup"
```

出力:

```
{  
  "id": "01sj5xdwhbw54kejwnt09fnpc1",  
  "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1",  
  "name": "ExampleAttributeGroup",  
  "attributes": "{\"SomeKey1\": \"SomeValue1\", \"SomeKey2\": \"SomeValue2\"}",  
  "creationTime": "2023-02-28T20:38:01.389000+00:00",  
  "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",  
  "tags": {  
    "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"  
  }  
}
```

詳細については、「Service Catalog 管理者ガイド」の [「属性グループのメタデータの管理」](#) を参照してください。AWS AppRegistry

- API 詳細については、AWS CLI 「コマンドリファレンス [GetAttributeGroup](#)」の「」を参照してください。

list-applications

次の例は、list-applications を使用する方法を説明しています。

AWS CLI

アプリケーションを一覧表示するには

次のlist-applications例では、AWS アカウント内のすべてのアプリケーションのリストを取得します。

```
aws servicecatalog-appregistry list-applications
```

出力:

```
{
  "applications": [
    {
      "id": "03axw94pjfj3uan00tcgbrxnkw",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/03axw94pjfj3uan00tcgbrxnkw",
      "name": "ExampleApplication2",
      "creationTime": "2023-02-28T21:59:34.094000+00:00",
      "lastUpdateTime": "2023-02-28T21:59:34.094000+00:00"
    },
    {
      "id": "055gw7aynr1i5mbv7kjwzx5945",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/055gw7aynr1i5mbv7kjwzx5945",
      "name": "ExampleApplication3",
      "creationTime": "2023-02-28T22:06:28.228000+00:00",
      "lastUpdateTime": "2023-02-28T22:06:28.228000+00:00"
    },
    {
      "id": "0ars38r6btoohvpvd9gqrptt91",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt91",
      "name": "ExampleApplication",
      "description": "This is an example application",
      "creationTime": "2023-02-28T21:10:10.820000+00:00",
      "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00"
    }
  ]
}
```

```
}
```

詳細については、AWS 「Service Catalog AppRegistry 管理者ガイド」の「[アプリケーションの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListApplications](#)」の「」を参照してください。

list-associated-attribute-groups

次の例は、list-associated-attribute-groups を使用する方法を説明しています。

AWS CLI

関連付けられた属性グループを一覧表示するには

次のlist-associated-attribute-groups例では、AWS アカウント内の特定のアプリケーションに関連付けられている AWS アカウント内のすべての属性グループのリストを取得します。

```
aws servicecatalog-appregistry list-associated-attribute-groups \
  --application "ExampleApplication"
```

出力:

```
{
  "attributeGroups": [
    "01sj5xdwhbw54kejwnt09fnpc1"
  ]
}
```

詳細については、「[Service Catalog 管理者ガイド](#)」の「[属性グループの関連付けと関連付け解除](#)」を参照してください。AWS AppRegistry

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssociatedAttributeGroups](#)」の「」を参照してください。

list-attribute-groups-for-application

次の例は、list-attribute-groups-for-application を使用する方法を説明しています。

AWS CLI

アプリケーションの属性グループを一覧表示するには

次のlist-attribute-groups-for-application例では、AWS アカウント内の特定のアプリケーションに関連付けられている AWS アカウント内のすべての属性グループの詳細を一覧表示します。

```
aws servicecatalog-appregistry list-attribute-groups-for-application \  
--application "ExampleApplication"
```

出力:

```
{  
  "attributeGroupsDetails": [  
    {  
      "id": "01sj5xdwhbw54kejwnt09fnpc1",  
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1",  
      "name": "ExampleAttributeGroup"  
    }  
  ]  
}
```

詳細については、AWS 「Service Catalog AppRegistry 管理者ガイド」の「[属性グループの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAttributeGroupsForApplication](#)」の「」を参照してください。

list-attribute-groups

次の例は、list-attribute-groups を使用する方法を説明しています。

AWS CLI

属性グループを一覧表示するには

次のlist-attribute-groups例では、AWS アカウント内のすべての属性グループのリストを取得します。

```
aws servicecatalog-appregistry list-attribute-groups
```

出力:

```
{
  "attributeGroups": [
    {
      "id": "011ge6y3emyjijt8dw8jn6r0hv",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/011ge6y3emyjijt8dw8jn6r0hv",
      "name": "ExampleAttributeGroup3",
      "creationTime": "2023-02-28T22:05:35.224000+00:00",
      "lastUpdateTime": "2023-02-28T22:05:35.224000+00:00"
    },
    {
      "id": "01sj5xdwhbw54kejwnt09fnpc1",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpc1",
      "name": "ExampleAttributeGroup",
      "description": "This is an example attribute group",
      "creationTime": "2023-02-28T20:38:01.389000+00:00",
      "lastUpdateTime": "2023-02-28T21:02:04.559000+00:00"
    },
    {
      "id": "03n1yffgq6d18vwrzxf0c70nm3",
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/03n1yffgq6d18vwrzxf0c70nm3",
      "name": "ExampleAttributeGroup2",
      "creationTime": "2023-02-28T21:57:30.687000+00:00",
      "lastUpdateTime": "2023-02-28T21:57:30.687000+00:00"
    }
  ]
}
```

詳細については、AWS「[Service Catalog AppRegistry 管理者ガイド](#)」の「[属性グループの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスListAttributeGroups](#)」の「」を参照してください。

update-application

次のコード例は、update-application を使用する方法を示しています。

AWS CLI

アプリケーションを更新するには

次のupdate-application例では、AWS アカウント内の特定のアプリケーションを更新して説明を含めます。

```
aws servicecatalog-appregistry update-application \  
  --application "ExampleApplication" \  
  --description "This is an example application"
```

出力:

```
{  
  "application": {  
    "id": "0ars38r6btoohvpvd9gqrptt91",  
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
    "name": "ExampleApplication",  
    "description": "This is an example application",  
    "creationTime": "2023-02-28T21:10:10.820000+00:00",  
    "lastUpdateTime": "2023-02-28T21:24:19.729000+00:00",  
    "tags": {  
      "aws:servicecatalog:applicationName": "ExampleApplication"  
    }  
  }  
}
```

詳細については、AWS「[Service Catalog AppRegistry 管理者ガイド](#)」の「[アプリケーションの編集](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスUpdateApplication](#)」の「」を参照してください。

update-attribute-group

次のコード例は、update-attribute-group を使用する方法を示しています。

AWS CLI

属性グループを更新するには

次のupdate-attribute-group例では、AWS アカウント内の特定の属性グループを更新して、記述を含めます。

```
aws servicecatalog-appregistry update-attribute-group \  
  --attribute-group "ExampleAttributeGroup" \  
  --description "This is an example attribute group"
```

出力:

```
{  
  "attributeGroup": {  
    "id": "01sj5xdwhbw54kejwnt09fnpc1",  
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1",  
    "name": "ExampleAttributeGroup",  
    "description": "This is an example attribute group",  
    "creationTime": "2023-02-28T20:38:01.389000+00:00",  
    "lastUpdateTime": "2023-02-28T21:02:04.559000+00:00",  
    "tags": {  
      "aws:servicecatalog:attributeGroupName": "ExampleAttributeGroup"  
    }  
  }  
}
```

詳細については、AWS 「Service Catalog AppRegistry 管理者ガイド」の「[属性グループの編集](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAttributeGroup](#)」の「」を参照してください。

を使用した Athena の例 AWS CLI

次のコード例は、Athena AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-get-named-query

次のコード例は、batch-get-named-query を使用する方法を示しています。

AWS CLI

複数のクエリに関する情報を返すには

次のbatch-get-named-query例では、指定された を持つ名前付きクエリに関する情報を返しますIDs。

```
aws athena batch-get-named-query \  
  --named-query-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-cdef-  
EXAMPLE22222 a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

出力:

```
{  
  "NamedQueries": [  
    {  
      "Name": "Flights Select Query",  
      "Description": "Sample query to get the top 10 airports with the most  
number of departures since 2000",  
      "Database": "sampledb",  
      "QueryString": "SELECT origin, count(*) AS total_departures\  
\nFROM  
\nflights_parquet\  
\nWHERE year >= '2000'\nGROUP BY origin\  
\nORDER BY total_departures  
DESC\  
\nLIMIT 10;",  
      "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "WorkGroup": "primary"  
    },  
    {  
      "Name": "Load flights table partitions",  
      "Description": "Sample query to load flights table partitions using MSCK  
REPAIR TABLE statement",  
      "Database": "sampledb",  
      "QueryString": "MSCK REPAIR TABLE flights_parquet;",  
      "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    }  
  ]  
}
```

```

        "WorkGroup": "primary"
    },
    {
        "Name": "CloudFront Select Query",
        "Description": "Sample query to view requests per operating system
during a particular time frame",
        "Database": "sampledb",
        "QueryString": "SELECT os, COUNT(*) count FROM cloudfront_logs WHERE
date BETWEEN date '2014-07-05' AND date '2014-08-05' GROUP BY os;",
        "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "WorkGroup": "primary"
    }
],
"UnprocessedNamedQueryIds": []
}

```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetNamedQuery](#)」の「」を参照してください。

batch-get-query-execution

次のコード例は、batch-get-query-execution を使用する方法を示しています。

AWS CLI

1 つ以上のクエリ実行に関する情報を返すには

次のbatch-get-query-execution例では、指定されたクエリを持つクエリのクエリ実行情報を返しますIDs。

```

aws athena batch-get-query-execution \
  --query-execution-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222

```

出力:

```

{
  "QueryExecutions": [
    {
      "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",

```

```
    "Query": "create database if not exists webdata",
    "StatementType": "DDL",
    "ResultConfiguration": {
      "OutputLocation": "s3://awsdoc-example-bucket/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111.txt"
    },
    "QueryExecutionContext": {},
    "Status": {
      "State": "SUCCEEDED",
      "SubmissionDateTime": 1593470720.592,
      "CompletionDateTime": 1593470720.902
    },
    "Statistics": {
      "EngineExecutionTimeInMillis": 232,
      "DataScannedInBytes": 0,
      "TotalExecutionTimeInMillis": 310,
      "ResultConfiguration": {
        "QueryQueueTimeInMillis": 50,
        "ServiceProcessingTimeInMillis": 28
      },
      "WorkGroup": "AthenaAdmin"
    },
  },
  {
    "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Query": "select date, location, browser, uri, status from
cloudfront_logs where method = 'GET' and status = 200 and location like 'SF0%'
limit 10",
    "StatementType": "DML",
    "ResultConfiguration": {
      "OutputLocation": "s3://awsdoc-example-bucket/a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222.csv"
    },
    "QueryExecutionContext": {
      "Database": "mydatabase",
      "Catalog": "awsdatacatalog"
    },
    "Status": {
      "State": "SUCCEEDED",
      "SubmissionDateTime": 1593469842.665,
      "CompletionDateTime": 1593469846.486
    },
    "Statistics": {
      "EngineExecutionTimeInMillis": 3600,
```

```

        "DataScannedInBytes": 203089,
        "TotalExecutionTimeInMillis": 3821,
        "QueryQueueTimeInMillis": 267,
        "QueryPlanningTimeInMillis": 1175
    },
    "WorkGroup": "AthenaAdmin"
}
],
"UnprocessedQueryExecutionIds": []
}

```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetQueryExecution](#)」の「」を参照してください。

create-data-catalog

次のコード例は、create-data-catalog を使用する方法を示しています。

AWS CLI

データカタログを作成するには

次のcreate-data-catalog例では、dynamo_db_catalogデータカタログを作成します。

```

aws athena create-data-catalog \
  --name dynamo_db_catalog \
  --type LAMBDA \
  --description "DynamoDB Catalog" \
  --parameters function=arn:aws:lambda:us-
west-2:111122223333:function:dynamo_db_lambda

```

このコマンドでは何も出力されません。結果を表示するには、`aws athena get-data-catalog --name dynamo_db_catalog` を使用します。

詳細については、[Amazon Athena ユーザーガイド](#)の「[カタログの登録： create-data-catalog](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDataCatalog](#)」の「」を参照してください。

create-named-query

次の例は、create-named-query を使用する方法を説明しています。

AWS CLI

名前付きクエリを作成するには

次のcreate-named-query例では、2016年1月にシアトルからニューヨークへのフライトについてflights_parquetテーブルにクエリを実行するAthenaAdminワークグループに保存されたクエリを作成します。このテーブルの出発と到着の両方が10分以上遅れています。テーブル内の空港コード値は二重引用符(SEA「」など)を含む文字列であるため、バックslashでエスケープされ、一重引用符で囲まれます。

```
aws athena create-named-query \  
  --name "SEA to JFK delayed flights Jan 2016" \  
  --description "Both arrival and departure delayed more than 10 minutes." \  
  --database sampledb \  
  --query-string "SELECT flightdate, carrier, flightnum, origin, dest, depdelayminutes, arrdelayminutes FROM sampledb.flights_parquet WHERE yr = 2016 AND month = 1 AND origin = '\"SEA\"' AND dest = '\"JFK\"' AND depdelayminutes > 10 AND arrdelayminutes > 10" \  
  --work-group AthenaAdmin
```

出力:

```
{  
  "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNamedQuery](#)」の「」を参照してください。

create-work-group

次のコード例は、create-work-group を使用する方法を示しています。

AWS CLI

ワークグループを作成するには

次のcreate-work-group例では、クエリ結果の出力場所 Data_Analyst_Group を持つというワークグループを作成しますs3://awsdoc-example-bucket。コマンドは、クエリ結果の出力場所を含むクライアント設定を上書きするワークグループを作成します。また、このコマンドは CloudWatch メトリクスを有効にし、ワークグループに 3 つのキーと値のタグペアを追加して、他のワークグループと区別します。--configuration 引数には、オプションを区切るカンマの前にスペースがないことに注意してください。

```
aws athena create-work-group \  
  --name Data_Analyst_Group \  
  --configuration ResultConfiguration={OutputLocation="s3://awsdoc-example-  
bucket"},EnforceWorkGroupConfiguration="true",PublishCloudWatchMetricsEnabled="true"  
 \  
  --description "Workgroup for data analysts" \  
  --tags Key=Division,Value=West Key=Location,Value=Seattle Key=Team,Value="Big  
Data"
```

このコマンドでは何も出力されません。結果を表示するには、を使用しますaws athena get-work-group --work-group Data_Analyst_Group。

詳細については、Amazon Athena ユーザーガイド」の「[ワークグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateWorkGroup](#)」の「」を参照してください。

delete-data-catalog

次の例は、delete-data-catalog を使用する方法を説明しています。

AWS CLI

データカタログを削除するには

次のdelete-data-catalog例では、UnusedDataCatalogデータカタログを削除します。

```
aws athena delete-data-catalog \  

```

```
--name UnusedDataCatalog
```

このコマンドでは何も出力されません。

詳細については、[Amazon Athena ユーザーガイド](#)の「[カタログの削除 : delete-data-catalog](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDataCatalog](#)」の「」を参照してください。

delete-named-query

次のコード例は、delete-named-query を使用する方法を示しています。

AWS CLI

名前付きクエリを削除するには

次のdelete-named-query例では、指定された ID を持つ名前付きクエリを削除します。

```
aws athena delete-named-query \  
  --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。 Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNamedQuery](#)」の「」を参照してください。

delete-work-group

次の例は、delete-work-group を使用する方法を説明しています。

AWS CLI

ワークグループを削除するには

次のdelete-work-group例では、TeamBワークグループを削除します。

```
aws athena delete-work-group \  
  --work-group TeamB
```

このコマンドでは何も出力されません。削除を確認するには、`aws athena list-work-groups` を使用します。

詳細については、「Amazon Athena ユーザーガイド」の「[ワークグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteWorkGroup](#)」の「」を参照してください。

get-data-catalog

次の例は、`get-data-catalog` を使用する方法を説明しています。

AWS CLI

データカタログに関する情報を返すには

次の`get-data-catalog`例では、`dynamo_db_catalog`データカタログに関する情報を返します。

```
aws athena get-data-catalog \  
  --name dynamo_db_catalog
```

出力:

```
{  
  "DataCatalog": {  
    "Name": "dynamo_db_catalog",  
    "Description": "DynamoDB Catalog",  
    "Type": "LAMBDA",  
    "Parameters": {  
      "catalog": "dynamo_db_catalog",  
      "metadata-function": "arn:aws:lambda:us-west-2:111122223333:function:dynamo_db_lambda",  
      "record-function": "arn:aws:lambda:us-west-2:111122223333:function:dynamo_db_lambda"  
    }  
  }  
}
```

```
}
```

詳細については、[Amazon Athena ユーザーガイド](#)の「[カタログの詳細の表示: get-data-catalog](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetDataCatalog](#)」の「」を参照してください。

get-database

次の例は、get-database を使用する方法を説明しています。

AWS CLI

データカタログ内のデータベースに関する情報を返すには

次のget-database例では、AwsDataCatalogデータカタログ内のsampleddbデータベースに関する情報を返します。

```
aws athena get-database \  
  --catalog-name AwsDataCatalog \  
  --database-name sampleddb
```

出力:

```
{  
  "Database": {  
    "Name": "sampleddb",  
    "Description": "Sample database",  
    "Parameters": {  
      "CreatedBy": "Athena",  
      "EXTERNAL": "TRUE"  
    }  
  }  
}
```

詳細については、[Amazon Athena ユーザーガイド](#)の「[データベースの詳細の表示: get-database](#)」を参照してください。 Amazon Athena

- API 詳細については、AWS CLI 「[コマンドリファレンスGetDatabase](#)」の「」を参照してください。

get-named-query

次の例は、get-named-query を使用する方法を説明しています。

AWS CLI

名前付きクエリを返すには

次のget-named-query例では、指定された ID を持つクエリに関する情報を返します。

```
aws athena get-named-query \  
  --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "NamedQuery": {  
    "Name": "CloudFront Logs - SF0",  
    "Description": "Shows successful GET request data for SF0",  
    "Database": "default",  
    "QueryString": "select date, location, browser, uri, status from  
cloudfront_logs where method = 'GET' and status = 200 and location like 'SF0%'  
limit 10",  
    "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "WorkGroup": "AthenaAdmin"  
  }  
}
```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。 Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[GetNamedQuery](#)」の「」を参照してください。

get-query-execution

次の例は、get-query-execution を使用する方法を説明しています。

AWS CLI

クエリ実行に関する情報を返すには

次のget-query-execution例では、指定されたクエリ ID を持つクエリに関する情報を返します。

```
aws athena get-query-execution \  
--query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "QueryExecution": {  
    "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Query": "select date, location, browser, uri, status from cloudfront_logs  
where method = 'GET  
' and status = 200 and location like 'SF0%' limit 10",  
    "StatementType": "DML",  
    "ResultConfiguration": {  
      "OutputLocation": "s3://awsdoc-example-bucket/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111.csv"  
    },  
    "QueryExecutionContext": {  
      "Database": "mydatabase",  
      "Catalog": "awsdatacatalog"  
    },  
    "Status": {  
      "State": "SUCCEEDED",  
      "SubmissionDateTime": 1593469842.665,  
      "CompletionDateTime": 1593469846.486  
    },  
    "Statistics": {  
      "EngineExecutionTimeInMillis": 3600,  
      "DataScannedInBytes": 203089,  
      "TotalExecutionTimeInMillis": 3821,  
      "QueryQueueTimeInMillis": 267,  
      "QueryPlanningTimeInMillis": 1175  
    },  
    "WorkGroup": "AthenaAdmin"  
  }  
}
```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。 Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[GetQueryExecution](#)」の「」を参照してください。

get-query-results

次の例は、get-query-results を使用する方法を説明しています。

AWS CLI

クエリの結果を返すには

次のget-query-results例では、指定されたクエリ ID を持つクエリの結果を返します。

```
aws athena get-query-results \  
  --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "ResultSet": {  
    "Rows": [  
      {  
        "Data": [  
          {  
            "VarCharValue": "date"  
          },  
          {  
            "VarCharValue": "location"  
          },  
          {  
            "VarCharValue": "browser"  
          },  
          {  
            "VarCharValue": "uri"  
          },  
          {  
            "VarCharValue": "status"  
          }  
        ]  
      },  
      {  
        "Data": [  
          {
```



```
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Safari"
      },
      {
        "VarCharValue": "/test-image-2.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Opera"
      },
      {
        "VarCharValue": "/test-image-2.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
```

```
        "VarCharValue": "Firefox"
      },
      {
        "VarCharValue": "/test-image-3.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Lynx"
      },
      {
        "VarCharValue": "/test-image-3.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "IE"
      },
      {
        "VarCharValue": "/test-image-2.jpeg"
      },
      {

```

```
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Opera"
      },
      {
        "VarCharValue": "/test-image-1.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
      {
        "VarCharValue": "2014-07-05"
      },
      {
        "VarCharValue": "SF04"
      },
      {
        "VarCharValue": "Chrome"
      },
      {
        "VarCharValue": "/test-image-3.jpeg"
      },
      {
        "VarCharValue": "200"
      }
    ]
  },
  {
    "Data": [
```

```
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "Firefox"
    },
    {
      "VarCharValue": "/test-image-2.jpeg"
    },
    {
      "VarCharValue": "200"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    },
    {
      "VarCharValue": "Chrome"
    },
    {
      "VarCharValue": "/test-image-3.jpeg"
    },
    {
      "VarCharValue": "200"
    }
  ]
},
{
  "Data": [
    {
      "VarCharValue": "2014-07-05"
    },
    {
      "VarCharValue": "SF04"
    }
  ],
```

```
        {
          "VarCharValue": "IE"
        },
        {
          "VarCharValue": "/test-image-2.jpeg"
        },
        {
          "VarCharValue": "200"
        }
      ]
    }
  ],
  "ResultSetMetadata": {
    "ColumnInfo": [
      {
        "CatalogName": "hive",
        "SchemaName": "",
        "TableName": "",
        "Name": "date",
        "Label": "date",
        "Type": "date",
        "Precision": 0,
        "Scale": 0,
        "Nullable": "UNKNOWN",
        "CaseSensitive": false
      },
      {
        "CatalogName": "hive",
        "SchemaName": "",
        "TableName": "",
        "Name": "location",
        "Label": "location",
        "Type": "varchar",
        "Precision": 2147483647,
        "Data": [
          {
            "Scale": 0,
            "Nullable": "UNKNOWN",
            "CaseSensitive": true
          }
        ],
        {
          "CatalogName": "hive",
          "SchemaName": "",
          "TableName": "",
```

```
        "Name": "browser",
        "Label": "browser",
        "Type": "varchar",
        "Precision": 2147483647,
        "Scale": 0,
        "Nullable": "UNKNOWN",
        "CaseSensitive": true
    },
    {
        "CatalogName": "hive",
        "SchemaName": "",
        "TableName": "",
        "Name": "uri",
        "Label": "uri",
        "Type": "varchar",
        "Precision": 2147483647,
        "Scale": 0,
        "Nullable": "UNKNOWN",
        "CaseSensitive": true
    },
    {
        "CatalogName": "hive",
        "SchemaName": "",
        "TableName": "",
        "Name": "status",
        "Label": "status",
        "Type": "integer",
        "Precision": 10,
        "Scale": 0,
        "Nullable": "UNKNOWN",
        "CaseSensitive": false
    }
]
}
},
"UpdateCount": 0
}
```

詳細については、Amazon Athena ユーザーガイド」の「[クエリ結果、出力ファイル、クエリ履歴の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetQueryResults](#)」の「」を参照してください。

get-table-metadata

次のコード例は、get-table-metadata を使用する方法を示しています。

AWS CLI

テーブルに関するメタデータ情報を返すには

次のget-table-metadata例では、countiesデータAwsDataCatalogカタログのsampleddbデータベースから、列名とそのデータ型を含むテーブルに関するメタデータ情報を返します。

```
aws athena get-table-metadata \  
  --catalog-name AwsDataCatalog \  
  --database-name sampleddb \  
  --table-name counties
```

出力:

```
{  
  "TableMetadata": {  
    "Name": "counties",  
    "CreateTime": 1593559968.0,  
    "LastAccessTime": 0.0,  
    "TableType": "EXTERNAL_TABLE",  
    "Columns": [  
      {  
        "Name": "name",  
        "Type": "string",  
        "Comment": "from deserializer"  
      },  
      {  
        "Name": "boundaryshape",  
        "Type": "binary",  
        "Comment": "from deserializer"  
      },  
      {  
        "Name": "motto",  
        "Type": "string",  
        "Comment": "from deserializer"  
      },  
      {  
        "Name": "population",
```

```

        "Type": "int",
        "Comment": "from deserializer"
    }
],
"PartitionKeys": [],
"Parameters": {
    "EXTERNAL": "TRUE",
    "inputformat": "com.esri.json.hadoop.EnclosedJsonInputFormat",
    "location": "s3://awsdoc-example-bucket/json",
    "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
    "serde.param.serialization.format": "1",
    "serde.serialization.lib": "com.esri.hadoop.hive.serde.JsonSerde",
    "transient_lastDdlTime": "1593559968"
}
}
}
}

```

詳細については、Amazon Athena ユーザーガイド」の「[テーブルの詳細の表示：get-table-metadata](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTableMetadata](#)」の「」を参照してください。

get-work-group

次の例は、get-work-group を使用する方法を説明しています。

AWS CLI

ワークグループに関する情報を返すには

次のget-work-group例では、AthenaAdminワークグループに関する情報を返します。

```
aws athena get-work-group \
  --work-group AthenaAdmin
```

出力:

```
{
  "WorkGroup": {
    "Name": "AthenaAdmin",
    "State": "ENABLED",
```



```
"Configuration": {
  "ResultConfiguration": {
    "OutputLocation": "s3://awsdoc-example-bucket/"
  },
  "EnforceWorkGroupConfiguration": false,
  "PublishCloudWatchMetricsEnabled": true,
  "RequesterPaysEnabled": false
},
"Description": "Workgroup for Athena administrators",
"CreationTime": 1573677174.105
}
```

詳細については、[Amazon Athena ユーザーガイド](#)の「[ワークグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetWorkGroup](#)」の「」を参照してください。

list-data-catalogs

次のコード例は、list-data-catalogs を使用する方法を示しています。

AWS CLI

Athena に登録されているデータカタログを一覧表示するには

次のlist-data-catalogs例では、Athena に登録されているデータカタログを一覧表示します。

```
aws athena list-data-catalogs
```

出力:

```
{
  "DataCatalogsSummary": [
    {
      "CatalogName": "AwsDataCatalog",
      "Type": "GLUE"
    },
    {
      "CatalogName": "cw_logs_catalog",
```

```
        "Type": "LAMBDA"
      },
      {
        "CatalogName": "cw_metrics_catalog",
        "Type": "LAMBDA"
      }
    ]
  }
}
```

詳細については、Amazon Athena ユーザーガイド」の「[登録済みカタログの一覧表示：list-data-catalogs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDataCatalogs](#)」の「」を参照してください。

list-databases

次のコード例は、list-databases を使用する方法を示しています。

AWS CLI

データカタログ内のデータベースを一覧表示するには

次のlist-databases例では、AwsDataCatalogデータカタログ内のデータベースを一覧表示します。

```
aws athena list-databases \  
  --catalog-name AwsDataCatalog
```

出力:

```
{  
  "DatabaseList": [  
    {  
      "Name": "default"  
    },  
    {  
      "Name": "mydatabase"  
    },  
    {  
      "Name": "newdb"  
    },  
    {
```

```
    "Name": "sampledb",
    "Description": "Sample database",
    "Parameters": {
      "CreatedBy": "Athena",
      "EXTERNAL": "TRUE"
    }
  },
  {
    "Name": "webdata"
  }
]
```

詳細については、[「Amazon Athena ユーザーガイド」の「カタログ内のデータベースの一覧表示: list-databases」](#)を参照してください。Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDatabases](#)」の「」を参照してください。

list-named-queries

次の例は、list-named-queries を使用する方法を説明しています。

AWS CLI

ワークグループの名前付きクエリを一覧表示するには

次のlist-named-queries例では、AthenaAdminワークグループの名前付きクエリを一覧表示します。

```
aws athena list-named-queries \
  --work-group AthenaAdmin
```

出力:

```
{
  "NamedQueryIds": [
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
  ]
}
```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[ListNamedQueries](#)」の「」を参照してください。

list-query-executions

次の例は、list-query-executions を使用する方法を説明しています。

AWS CLI

指定されたワークグループ内のクエリIDsのクエリを一覧表示するには

次のlist-query-executions例では、AthenaAdminワークグループIDs内のクエリを最大 10 個まで一覧表示します。

```
aws athena list-query-executions \  
  --work-group AthenaAdmin \  
  --max-items 10
```

出力:

```
{  
  "QueryExecutionIds": [  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11110",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11114",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11115",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11116",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11117",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11118",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11119"  
  ],  
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
```

詳細については、[Amazon Athena ユーザーガイド](#)の「[クエリ結果、出力ファイル、クエリ履歴の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListQueryExecutions](#)」の「」を参照してください。

list-table-metadata

次の例は、list-table-metadata を使用する方法を説明しています。

AWS CLI

データカタログの指定されたデータベース内のテーブルのメタデータを一覧表示するには

次のlist-table-metadata例では、AwsDataCatalogデータカタログのgeographyデータベース内の最大 2 つのテーブルのメタデータ情報を返します。

```
aws athena list-table-metadata \  
  --catalog-name AwsDataCatalog \  
  --database-name geography \  
  --max-items 2
```

出力:

```
{  
  "TableMetadataList": [  
    {  
      "Name": "country_codes",  
      "CreateTime": 1586553454.0,  
      "TableType": "EXTERNAL_TABLE",  
      "Columns": [  
        {  
          "Name": "country",  
          "Type": "string",  
          "Comment": "geo id"  
        },  
        {  
          "Name": "alpha-2 code",  
          "Type": "string",  
          "Comment": "geo id2"  
        },  
        {  
          "Name": "alpha-3 code",  
          "Type": "string",  
          "Comment": "state name"  
        }  
      ]  
    }  
  ]  
}
```

```

    },
    {
      "Name": "numeric code",
      "Type": "bigint",
      "Comment": ""
    },
    {
      "Name": "latitude",
      "Type": "bigint",
      "Comment": "location (latitude)"
    },
    {
      "Name": "longitude",
      "Type": "bigint",
      "Comment": "location (longitude)"
    }
  ],
  "Parameters": {
    "areColumnsQuoted": "false",
    "classification": "csv",
    "columnsOrdered": "true",
    "delimiter": ",",
    "has_encrypted_data": "false",
    "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
    "location": "s3://awsdoc-example-bucket/csv/countrycode",
    "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
    "serde.param.field.delim": ",",
    "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "skip.header.line.count": "1",
    "typeOfData": "file"
  }
},
{
  "Name": "county_populations",
  "CreateTime": 1586553446.0,
  "TableType": "EXTERNAL_TABLE",
  "Columns": [
    {
      "Name": "id",
      "Type": "string",
      "Comment": "geo id"
    }
  ],

```

```

        {
            "Name": "country",

            "Name": "id2",
            "Type": "string",
            "Comment": "geo id2"
        },
        {
            "Name": "county",
            "Type": "string",
            "Comment": "county name"
        },
        {
            "Name": "state",
            "Type": "string",
            "Comment": "state name"
        },
        {
            "Name": "population estimate 2018",
            "Type": "string",
            "Comment": ""
        }
    ],
    "Parameters": {
        "areColumnsQuoted": "false",
        "classification": "csv",
        "columnsOrdered": "true",
        "delimiter": ",",
        "has_encrypted_data": "false",
        "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
        "location": "s3://awsdoc-example-bucket/csv/CountyPopulation",
        "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
        "serde.param.field.delim": ",",
        "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "skip.header.line.count": "1",
        "typeOfData": "file"
    }
}
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

詳細については、Amazon Athena ユーザーガイド」の「[データベース内のすべてのテーブルのメタデータの表示 list-table-metadata](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTableMetadata](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

例 1: ワークグループのタグを一覧表示するには

次のlist-tags-for-resource例では、Data_Analyst_Groupワークグループのタグを一覧表示します。

```
aws athena list-tags-for-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:workgroup/  
Data_Analyst_Group
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Division",  
      "Value": "West"  
    },  
    {  
      "Key": "Team",  
      "Value": "Big Data"  
    },  
    {  
      "Key": "Location",  
      "Value": "Seattle"  
    }  
  ]  
}
```

例 2: データカタログのタグを一覧表示するには

次のlist-tags-for-resource例では、dynamo_db_catalogデータカタログのタグを一覧表示します。

```
aws athena list-tags-for-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Division",  
      "Value": "Mountain"  
    },  
    {  
      "Key": "Organization",  
      "Value": "Retail"  
    },  
    {  
      "Key": "Product_Line",  
      "Value": "Shoes"  
    },  
    {  
      "Key": "Location",  
      "Value": "Denver"  
    }  
  ]  
}
```

詳細については、Amazon Athena ユーザーガイド」の「[リソースのタグの一覧表示 list-tags-for-resource](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-work-groups

次の例は、list-work-groups を使用する方法を説明しています。

AWS CLI

ワークグループを一覧表示するには

次のlist-work-groups例では、現在のアカウントのワークグループを一覧表示します。

```
aws athena list-work-groups
```

出力:

```
{
  "WorkGroups": [
    {
      "Name": "Data_Analyst_Group",
      "State": "ENABLED",
      "Description": "",
      "CreationTime": 1578006683.016
    },
    {
      "Name": "AthenaAdmin",
      "State": "ENABLED",
      "Description": "",
      "CreationTime": 1573677174.105
    },
    {
      "Name": "primary",
      "State": "ENABLED",
      "Description": "",
      "CreationTime": 1567465222.723
    }
  ]
}
```

詳細については、Amazon Athena ユーザーガイドの「[ワークグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListWorkGroups](#)」の「」を参照してください。

start-query-execution

次のコード例は、start-query-execution を使用する方法を示しています。

AWS CLI

例 1: 指定されたデータベースとデータカタログ内の指定されたテーブルでワークグループでクエリを実行するには

次のstart-query-execution例では、AthenaAdminワークグループを使用して、AwsDataCatalogデータカタログの cloudfront_logs テーブル cflogsdatabase でクエリを実行します。

```
aws athena start-query-execution \  
  --query-string "select date, location, browser, uri, status from cloudfront_logs  
  where method = 'GET' and status = 200 and location like 'SF0%' limit 10" \  
  --work-group "AthenaAdmin" \  
  --query-execution-context Database=cflogsdatabase,Catalog=AwsDataCatalog
```

出力:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

例 2: 指定されたワークグループを使用して、指定されたデータカタログにデータベースを作成するクエリを実行するには

次のstart-query-execution例では、AthenaAdminワークグループを使用して、デフォルトのデータカタログ newdb にデータベースを作成しますAwsDataCatalog。

```
aws athena start-query-execution \  
  --query-string "create database if not exists newdb" \  
  --work-group "AthenaAdmin"
```

出力:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112"  
}
```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

例 3: 指定されたデータベースとデータカタログのテーブルにビューを作成するクエリを実行するには

次のstart-query-execution例では、のcloudfront_logsテーブルに SELECTステートメントを使用してビュー cflogsdatabaseを作成しますcf10。

```
aws athena start-query-execution \  
  --query-string "CREATE OR REPLACE VIEW cf10 AS SELECT * FROM cloudfront_logs  
  Limit 10" \  
  --query-execution-context Database=cflogsdatabase
```

出力:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11113"  
}
```

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

- API 詳細については、AWS CLI 「コマンドリファレンス[StartQueryExecution](#)」の「」を参照してください。

stop-query-execution

次の例は、stop-query-execution を使用する方法を説明しています。

AWS CLI

実行中のクエリを停止するには

次のstop-query-execution例では、指定されたクエリ ID を持つクエリを停止します。

```
aws athena stop-query-execution \  
  --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、[Amazon Athena SQL ユーザーガイド](#)の「[Amazon Athena を使用したクエリの実行](#)」を参照してください。Amazon Athena

- API 詳細については、AWS CLI 「[コマンドリファレンスStopQueryExecution](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを追加

次のtag-resource例では、dynamo_db_catalogデータカタログに 3 つのタグを追加します。

```
aws athena tag-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog \  
  --  
tags Key=Organization,Value=Retail Key=Division,Value=Mountain Key=Product_Line,Value=Shoes
```

このコマンドでは何も出力されません。結果を表示するには、`aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog` を使用します。

詳細については、Amazon Athena ユーザーガイドの「[リソースへのタグの追加: tag-resource](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスTagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグを削除するには

次の`untag-resource`例では、`dynamo_db_catalog` データカタログリソースから `Specialization` および `Focus` キーとそれに関連する値を削除します。

```
aws athena untag-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog \  
  --tag-keys Specialization Focus
```

このコマンドでは何も出力されません。結果を表示するには、`list-tags-for-resource` コマンドを使用します。

詳細については、「Amazon Athena ユーザーガイド」の「[リソースからのタグの削除: untag-resource](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-data-catalog

次のコード例は、`update-data-catalog` を使用する方法を示しています。

AWS CLI

データカタログを更新するには

次の`update-data-catalog`例では、Lambda 関数と `cw_logs_catalog` データカタログの説明を更新します。

```
aws athena update-data-catalog \  
  --name cw_logs_catalog \  
  --type LAMBDA \  
  --description "New CloudWatch Logs Catalog" \  
  --function=arn:aws:lambda:us-west-2:111122223333:function:new_cw_logs_lambda
```

このコマンドでは何も出力されません。結果を表示するには、`aws athena get-data-catalog --name cw_logs_catalog` を使用します。

詳細については、「Amazon Athena ユーザーガイド」の「[カタログの更新: update-data-catalog](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateDataCatalog](#)」の「」を参照してください。

update-work-group

次のコード例は、update-work-group を使用する方法を示しています。

AWS CLI

ワークグループを更新するには

次のupdate-work-group例では、Data_Analyst_Groupワークグループを無効にします。ユーザーは無効なワークグループでクエリを実行または作成することはできませんが、メトリクス、データ使用量制限コントロール、ワークグループ設定、クエリ履歴、保存されたクエリを表示することはできます。

```
aws athena update-work-group \  
  --work-group Data_Analyst_Group \  
  --state DISABLED
```

このコマンドでは何も出力されません。状態の変化を確認するには、aws athena get-work-group --work-group Data_Analyst_Groupを使用して出力のStateプロパティを確認します。

詳細については、[Amazon Athena ユーザーガイド](#)の「[ワークグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateWorkGroup](#)」の「」を参照してください。

を使用した Auto Scaling の例 AWS CLI

次のコード例は、Auto Scaling AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

attach-instances

次の例は、attach-instances を使用する方法を説明しています。

AWS CLI

Auto Scaling グループにインスタンスをアタッチするには

この例では、指定されたインスタンスを指定された Auto Scaling グループにアタッチします。

```
aws autoscaling attach-instances \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachInstances](#)」の「」を参照してください。

attach-load-balancer-target-groups

次の例は、attach-load-balancer-target-groups を使用する方法を説明しています。

AWS CLI

ターゲットグループを Auto Scaling グループにアタッチするには

この例では、指定されたターゲットグループを、指定された Auto Scaling グループにアタッチします。

```
aws autoscaling attach-load-balancer-target-groups \  
  --auto-scaling-group-name my-asg \  
  --target-group-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」のElastic Load Balancing](#)と「Amazon Auto Scaling」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachLoadBalancerTargetGroups](#)」の「」を参照してください。

attach-load-balancers

次のコード例は、attach-load-balancers を使用する方法を示しています。

AWS CLI

Classic Load Balancer を Auto Scaling グループにアタッチするには

この例では、指定された Classic Load Balancer を指定された Auto Scaling グループにアタッチします。

```
aws autoscaling attach-load-balancers \  
  --load-balancer-names my-load-balancer \  
  --auto-scaling-group-name my-asg
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」のElastic Load Balancing](#)と「Amazon Auto Scaling」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachLoadBalancers](#)」の「」を参照してください。

cancel-instance-refresh

次のコード例は、cancel-instance-refresh を使用する方法を示しています。

AWS CLI

インスタンスの更新をキャンセルするには

次のcancel-instance-refresh例では、指定された Auto Scaling グループの進行中のインスタンスの更新をキャンセルします。

```
aws autoscaling cancel-instance-refresh \  
  --auto-scaling-group-name my-asg
```

出力:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[インスタンスの更新をキャンセルする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelInstanceRefresh](#)」の「」を参照してください。

complete-lifecycle-action

次の例は、complete-lifecycle-action を使用する方法を説明しています。

AWS CLI

ライフサイクルアクションを完了するには

この例では、インスタンスの起動または終了を完了できるように、指定されたライフサイクルアクションが完了したことを Amazon EC2 Auto Scaling に通知します。

```
aws autoscaling complete-lifecycle-action \
  --lifecycle-hook-name my-launch-hook \
  --auto-scaling-group-name my-asg \
  --lifecycle-action-result CONTINUE \
  --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

このコマンドでは何も出力されません。

詳細については、「[Amazon EC2 Auto Scaling ユーザーガイド](#)」の「[Amazon Auto Scaling ライフサイクルフック](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[CompleteLifecycleAction](#)」の「」を参照してください。

create-auto-scaling-group

次の例は、create-auto-scaling-group を使用する方法を説明しています。

AWS CLI

例 1: Auto Scaling グループを作成するには

次の `create-auto-scaling-group` の例では、リージョン内の複数のアベイラビリティゾーンの子ネット内に Auto Scaling グループを作成します。インスタンスは、指定された起動テンプレートのデフォルトバージョンで起動されます。終了ポリシーやヘルスチェック設定など、他のほとんどの設定にはデフォルトが使用されることに注意してください。

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12 \  
  --min-size 1 \  
  --max-size 5 \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループ](#)」を参照してください。 EC2 Auto Scaling

例 2: Application Load Balancer、Network Load Balancer、または Gateway Load Balancer をアタッチするには

この例ではARN、予想されるトラフィックをサポートするロードバランサーのターゲットグループの を指定します。ヘルスチェックタイプは、Elastic Load Balancing がインスタンスを異常として報告したときに、Auto Scaling グループがそのインスタンスを置き換えるよう ELB を指定します。このコマンドは、ヘルスチェックの猶予期間 (600 秒) も定義します。猶予期間は、新しく起動したインスタンスが早期に終了するのを防ぐのに役立ちます。

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12 \  
  --target-group-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/943f017f100becff \  
  --health-check-type ELB \  
  --health-check-grace-period 600 \  
  --min-size 1 \  
  --max-size 5 \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の[Elastic Load Balancing](#)と「Amazon Auto Scaling」を参照してください。 EC2 Auto Scaling

例 3: プレイメントグループを指定し、起動テンプレートの最新バージョンを使用するには

この例では、単一のアベイラビリティーゾーン内のプレイメントグループ内でインスタンスを起動します。これは、HPCワークロードを持つ低レイテンシーグループに役立ちます。この例では、グループの最小サイズ、最大サイズ、希望する容量も指定しています。

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest' \  
  --min-size 1 \  
  --max-size 5 \  
  --desired-capacity 3 \  
  --placement-group my-placement-group \  
  --vpc-zone-identifier "subnet-6194ea3b"
```

このコマンドでは何も出力されません。

詳細については、「Linux インスタンス用 Amazon ユーザーガイド」の「[プレイメントグループ](#)」を参照してください。 EC2

例 4: 単一のインスタンスの Auto Scaling グループを指定し、特定のバージョンの起動テンプレートを使用するには

この例では、単一のインスタンスが強制的に実行されるように、最小容量と最大容量を 1 に設定した Auto Scaling グループを作成します。コマンドは、既存の ID が ENI 指定されている起動テンプレートの v1 も指定します。eth0 ENI の既存のを指定する起動テンプレートを使用する場合は、リクエストにサブネット ID を指定せずに、ネットワークインターフェイスに一致する Auto Scaling グループのアベイラビリティーゾーンを指定する必要があります。

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg-single-instance \  
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1' \  
  --min-size 1 \  
  --max-size 1 \  
  --availability-zones us-west-2a
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループ](#)」を参照してください。 EC2 Auto Scaling

例 5: 別の終了ポリシーを指定するには

この例では、起動設定を使用して Auto Scaling グループを作成し、最も古いインスタンスを最初に終了するように終了ポリシーを設定します。またこのコマンドは、Role キーと WebServer 値を使用して、グループとインスタンスにタグを適用します。

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-configuration-name my-lc \  
  --min-size 1 \  
  --max-size 5 \  
  --termination-policies "OldestInstance" \  
  --tags "ResourceId=my-asg, ResourceType=auto-scaling-  
group, Key=Role, Value=WebServer, PropagateAtLaunch=true" \  
  --vpc-zone-identifier "subnet-5ea0c127, subnet-6194ea3b, subnet-c934b782"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling 終了ポリシーの使用」](#)を参照してください。 EC2 Auto Scaling

例 6: 起動ライフサイクルフックを指定するには

この例では、インスタンス起動時のカスタムアクションをサポートするライフサイクルフックで Auto Scaling グループを設定します。

```
aws autoscaling create-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

config.json ファイルの内容。

```
{  
  "AutoScalingGroupName": "my-asg",  
  "LaunchTemplate": {  
    "LaunchTemplateId": "lt-1234567890abcde12"  
  },  
  "LifecycleHookSpecificationList": [{  
    "LifecycleHookName": "my-launch-hook",  
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",  
    "NotificationTargetARN": "arn:aws:sqs:us-west-2:123456789012:my-sqs-queue",  
    "RoleARN": "arn:aws:iam::123456789012:role/my-notification-role",  
    "NotificationMetadata": "SQS message metadata",  
  }  
}
```

```
        "HeartbeatTimeout": 4800,  
        "DefaultResult": "ABANDON"  
    }],  
    "MinSize": 1,  
    "MaxSize": 5,  
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",  
    "Tags": [{  
        "ResourceType": "auto-scaling-group",  
        "ResourceId": "my-asg",  
        "PropagateAtLaunch": true,  
        "Value": "test",  
        "Key": "environment"  
    }]  
}
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling ライフサイクルフック」](#)を参照してください。 EC2 Auto Scaling

例 7: 終了ライフサイクルフックを指定するには

次の例は、インスタンス終了時のカスタムアクションをサポートするライフサイクルフックで Auto Scaling グループを設定します。

```
aws autoscaling create-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

config.json の内容:

```
{  
  "AutoScalingGroupName": "my-asg",  
  "LaunchTemplate": {  
    "LaunchTemplateId": "lt-1234567890abcde12"  
  },  
  "LifecycleHookSpecificationList": [{  
    "LifecycleHookName": "my-termination-hook",  
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",  
    "HeartbeatTimeout": 120,  
    "DefaultResult": "CONTINUE"  
  }],  
  "MinSize": 1,  
  "MaxSize": 5,  
}
```

```
"TargetGroupARNs": [  
    "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-  
targets/73e2d6bc24d8a067"  
],  
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"  
}
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling ライフサイクルフック」](#)を参照してください。 EC2 Auto Scaling

例 8: カスタム終了ポリシーを指定するには

この例では、Auto Scaling インスタンスのどのインスタンスを安全に終了できるかを Amazon Auto Scaling に通知するカスタム Lambda 関数終了ポリシーを指定する EC2 Auto Scaling グループを作成します。

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg-single-instance \  
  --launch-template LaunchTemplateName=my-template-for-auto-scaling \  
  --min-size 1 \  
  --max-size 5 \  
  --termination-policies "arn:aws:lambda:us-  
west-2:123456789012:function>HelloFunction:prod" \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の[「Lambda を使用したカスタム終了ポリシーの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAutoScalingGroup](#)」の「」を参照してください。

create-launch-configuration

次の例は、create-launch-configuration を使用する方法を説明しています。

AWS CLI

例 1: 起動設定を作成するには

この例では、シンプルな起動設定を作成します。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「起動設定の作成」](#) を参照してください。

例 2: セキュリティグループ、キーペア、ブートラッピングスクリプトを使用して起動設定を作成するには

この例では、セキュリティグループ、キーペア、ユーザーデータに含まれるブートラッピングスクリプトを使用して起動設定を作成します。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --security-groups sg-eb2af88example \  
  --key-name my-key-pair \  
  --user-data file://myuserdata.txt
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「起動設定の作成」](#) を参照してください。

例 3: IAMロールを使用して起動設定を作成するには

この例では、IAMロールのインスタンスプロファイル名を使用して起動設定を作成します。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --iam-instance-profile my-autoscaling-role
```

このコマンドでは何も出力されません。

詳細については、[IAM「Amazon Auto Scaling ユーザーガイド」の「Amazon EC2インスタンスで実行されるアプリケーションのロール」](#)を参照してください。 EC2 Auto Scaling

例 4: 詳細モニタリングを有効にして起動設定を作成するには

この例では、EC2詳細なモニタリングを有効にした起動設定を作成し、1分 CloudWatch 間隔で EC2メトリクスを に送信します。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --instance-monitoring Enabled=true
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling インスタンスのモニタリングの設定](#)」を参照してください。 EC2 Auto Scaling

例 5: スポットインスタンスを起動する起動設定を作成するには

この例では、スポットインスタンスを唯一の購入オプションとして使用する起動設定を作成します。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --spot-price "0.50"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Auto Scaling ユーザーガイド」の「[スポットインスタンスのリクエスト](#)」を参照してください。 EC2 Auto Scaling

例 6: EC2インスタンスを使用して起動設定を作成するには

この例では、既存のインスタンスの属性に基づいて起動設定を作成します。これにより、プレイメントテナンシーと、`--placement-tenancy` および `--no-associate-public-ip-address` オプションを含めることでパブリック IP アドレスが設定されるかどうかを上書きされます。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc-from-instance \  
  --instance-id i-0123a456700123456 \  
  --instance-type m5.large \  
  --no-associate-public-ip-address \  
  --placement-tenancy dedicated
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の[EC2「インスタンスを使用した起動設定の作成」](#)を参照してください。

例 7: Amazon EBSボリュームのブロックデバイスマッピングを使用して起動設定を作成するには

この例では、デバイス名と/dev/sdhボリュームサイズが 20 の Amazon EBSgp3ボリュームのブロックデバイスマッピングを使用して起動設定を作成します。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --block-device-mappings '[{"DeviceName":"/dev/sdh", "Ebs":  
{"VolumeSize":20, "VolumeType":"gp3"}}']
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling APIリファレンス[EBS](#)」の「」を参照してください。

JSON形式のパラメータ値を引用するための構文については、AWS「[コマンドラインインターフェイスユーザーガイド](#)」の「[の文字列での引用符の使用 AWS CLI](#)」を参照してください。

例 8: インスタンスストアボリュームのブロックデバイスマッピングを使用して起動設定を作成するには

この例では、をデバイス名 のインスタンスストアボリュームephemeral1として起動設定を作成します/dev/sdc。

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large
```

```
--instance-type m5.large \  
--block-device-mappings '[{"DeviceName":"/dev/sdc","VirtualName":"ephemeral1"}]'
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling APIリファレンス[BlockDeviceMapping](#)」の「」を参照してください。

JSON形式のパラメータ値を引用するための構文については、AWS「コマンドラインインターフェイスユーザーガイド」の「[の文字列での引用符の使用 AWS CLI](#)」を参照してください。

例 9: 起動設定を作成し、起動時にブロックデバイスがアタッチされないようにするには

この例では、のブロックデバイスマッピング AMI (など) で指定されたブロックデバイスを抑制する起動設定を作成します /dev/sdf。

```
aws autoscaling create-launch-configuration \  
--launch-configuration-name my-lc \  
--image-id ami-04d5cc9b88example \  
--instance-type m5.large \  
--block-device-mappings '[{"DeviceName":"/dev/sdf","NoDevice":""}]'
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling APIリファレンス[BlockDeviceMapping](#)」の「」を参照してください。

JSON形式のパラメータ値を引用するための構文については、AWS「コマンドラインインターフェイスユーザーガイド」の「[の文字列での引用符の使用 AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateLaunchConfiguration](#)」の「」を参照してください。

create-or-update-tags

次の例は、create-or-update-tags を使用する方法を説明しています。

AWS CLI

Auto Scaling グループのタグを作成または更新するには

この例では、指定された Auto Scaling グループに 2 つのタグを追加します。

```
aws autoscaling create-or-update-tags \  
  --tags ResourceId=my-asg,ResourceType=auto-scaling-  
group,Key=Role,Value=WebServer,PropagateAtLaunch=true ResourceId=my-  
asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research,PropagateAtLaunch=true
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスのタグ付け](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateOrUpdateTags](#)」の「」を参照してください。

delete-auto-scaling-group

次の例は、delete-auto-scaling-group を使用する方法を説明しています。

AWS CLI

例 1: 指定された Auto Scaling グループを削除するには

この例は、指定された Auto Scaling グループを削除します。

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling インフラストラクチャの削除](#)」を参照してください。 EC2 Auto Scaling

例 2: 指定された Auto Scaling グループを強制的に削除するには

グループ内のインスタンスが終了するのを待たずに Auto Scaling グループを削除するには、--force-delete オプションを使用します。

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling インフラストラクチャの削除](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAutoScalingGroup](#)」の「」を参照してください。

delete-launch-configuration

次の例は、delete-launch-configuration を使用する方法を説明しています。

AWS CLI

起動設定を削除するには

この例では、指定された起動設定を削除します。

```
aws autoscaling delete-launch-configuration \  
  --launch-configuration-name my-launch-config
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling インフラストラクチャの削除](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLaunchConfiguration](#)」の「」を参照してください。

delete-lifecycle-hook

次のコード例は、delete-lifecycle-hook を使用する方法を示しています。

AWS CLI

ライフサイクルフックを削除するには

この例では、指定されたライフサイクルフックを削除します。

```
aws autoscaling delete-lifecycle-hook \  
  --lifecycle-hook-name my-lifecycle-hook \  
  --auto-scaling-group-name my-asg
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLifecycleHook](#)」の「」を参照してください。

delete-notification-configuration

次のコード例は、delete-notification-configuration を使用する方法を示しています。

AWS CLI

Auto Scaling 通知を削除するには

この例では、指定された Auto Scaling グループから指定された通知を削除します。

```
aws autoscaling delete-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[通知設定の削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNotificationConfiguration](#)」の「」を参照してください。

delete-policy

次の例は、delete-policy を使用する方法を説明しています。

AWS CLI

スケーリングポリシーを削除するには

この例では、指定されたスケーリングポリシーを削除します。

```
aws autoscaling delete-policy \  
  --auto-scaling-group-name my-asg \  
  --policy-name alb1000-target-tracking-scaling-policy
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicy](#)」の「」を参照してください。

delete-scheduled-action

次のコード例は、delete-scheduled-action を使用する方法を示しています。

AWS CLI

Auto Scaling グループからスケジュールされたアクションを削除するには

この例では、指定された Auto Scaling グループから指定されたスケジュールされたアクションを削除します。

```
aws autoscaling delete-scheduled-action \  
  --auto-scaling-group-name my-asg \  
  --scheduled-action-name my-scheduled-action
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteScheduledAction](#)」の「」を参照してください。

delete-tags

次の例は、delete-tags を使用する方法を説明しています。

AWS CLI

Auto Scaling グループからタグを削除するには

この例では、指定された Auto Scaling グループから指定されたタグを削除します。

```
aws autoscaling delete-tags \  
  --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスのタグ付け](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteTags](#)」の「」を参照してください。

delete-warm-pool

次のコード例は、delete-warm-pool を使用する方法を示しています。

AWS CLI

例 1: ウォームプールを削除するには

次の例では、指定された Auto Scaling グループのウォームプールを削除します。

```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling のウォームプール」](#)を参照してください。 EC2 Auto Scaling

例 2: ウォームプールを強制的に削除するには

インスタンスの終了を待たずにウォームプールを削除するには、`--force-delete` オプションを使用します。

```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling のウォームプール」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteWarmPool](#)」の「」を参照してください。

describe-account-limits

次のコード例は、describe-account-limits を使用する方法を示しています。

AWS CLI

Amazon EC2 Auto Scaling アカウントの制限を記述するには

この例では、AWS アカウントの Amazon EC2 Auto Scaling の制限について説明します。

```
aws autoscaling describe-account-limits
```

出力:

```
{
  "NumberOfLaunchConfigurations": 5,
  "MaxNumberOfLaunchConfigurations": 100,
  "NumberOfAutoScalingGroups": 3,
  "MaxNumberOfAutoScalingGroups": 20
}
```

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling サービスクォータ」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountLimits](#)」の「」を参照してください。

describe-adjustment-types

次のコード例は、describe-adjustment-types を使用する方法を示しています。

AWS CLI

使用可能なスケーリング調整タイプを記述するには

この例では、使用可能な調整タイプについて説明します。

```
aws autoscaling describe-adjustment-types
```

出力:

```
{
  "AdjustmentTypes": [
    {
      "AdjustmentType": "ChangeInCapacity"
    },
  ],
}
```

```
{
  "AdjustmentType": "ExactCapacity"
},
{
  "AdjustmentType": "PercentChangeInCapacity"
}
]
```

詳細については、[「Amazon Auto Scaling ユーザーガイド」](#)の「[スケーリング調整タイプ](#) Auto ScalingEC2」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAdjustmentTypes](#)」の「」を参照してください。

describe-auto-scaling-groups

次のコード例は、describe-auto-scaling-groups を使用する方法を示しています。

AWS CLI

例 1: 指定された Auto Scaling グループを記述するには

この例は、指定された Auto Scaling グループを記述します。

```
aws autoscaling describe-auto-scaling-groups \
  --auto-scaling-group-name my-asg
```

出力:

```
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "my-asg",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-1234567890abcde12"
      },
      "MinSize": 0,
```

```
    "MaxSize": 1,
    "DesiredCapacity": 1,
    "DefaultCooldown": 300,
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2b",
      "us-west-2c"
    ],
    "LoadBalancerNames": [],
    "TargetGroupARNs": [],
    "HealthCheckType": "EC2",
    "HealthCheckGracePeriod": 0,
    "Instances": [
      {
        "InstanceId": "i-06905f55584de02da",
        "InstanceType": "t2.micro",
        "AvailabilityZone": "us-west-2a",
        "HealthStatus": "Healthy",
        "LifecycleState": "InService",
        "ProtectedFromScaleIn": false,
        "LaunchTemplate": {
          "LaunchTemplateName": "my-launch-template",
          "Version": "1",
          "LaunchTemplateId": "lt-1234567890abcde12"
        }
      }
    ],
    "CreatedTime": "2023-10-28T02:39:22.152Z",
    "SuspendedProcesses": [],
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
    "EnabledMetrics": [],
    "Tags": [],
    "TerminationPolicies": [
      "Default"
    ],
    "NewInstancesProtectedFromScaleIn": false,
    "ServiceLinkedRoleARN": "arn",
    "TrafficSources": []
  }
]
```

例 2: 指定された最初の 100 個の Auto Scaling グループを記述するには

この例は、指定された複数の Auto Scaling グループを記述します。最大 100 個のグループ名を指定できます。

```
aws autoscaling describe-auto-scaling-groups \  
  --max-items 100 \  
  --auto-scaling-group-name "group1" "group2" "group3" "group4"
```

出力例については、例 1 を参照してください。

例 3: 指定されたリージョンの Auto Scaling グループを記述するには

この例では、指定されたリージョンの Auto Scaling グループを最大 75 グループまで記述します。

```
aws autoscaling describe-auto-scaling-groups \  
  --max-items 75 \  
  --region us-east-1
```

出力例については、例 1 を参照してください。

例 4: 指定された数の Auto Scaling グループを記述するには

特定の数の Auto Scaling グループを返すには、`--max-items` オプションを使用します。

```
aws autoscaling describe-auto-scaling-groups \  
  --max-items 1
```

出力例については、例 1 を参照してください。

出力に `NextToken` フィールドが含まれている場合は、さらに多くのグループがあることを示しています。追加のグループを取得するには、次のように、以降の呼び出しで `--starting-token` オプションを使用してこのフィールドの値を使用します。

```
aws autoscaling describe-auto-scaling-groups \  
  --starting-token Z3M3LMPEXAMPLE
```

出力例については、例 1 を参照してください。

例 5: 起動設定を使用する Auto Scaling グループを記述するには

この例では、`--query` オプションを使用して、起動設定を使用する Auto Scaling グループを記述します。

```
aws autoscaling describe-auto-scaling-groups \  
--query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

出力:

```
[  
  {  
    "AutoScalingGroupName": "my-asg",  
    "AutoScalingGroupARN": "arn:aws:autoscaling:us-  
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-  
a11a-7b0acd480f03:autoScalingGroupName/my-asg",  
    "LaunchConfigurationName": "my-lc",  
    "MinSize": 0,  
    "MaxSize": 1,  
    "DesiredCapacity": 1,  
    "DefaultCooldown": 300,  
    "AvailabilityZones": [  
      "us-west-2a",  
      "us-west-2b",  
      "us-west-2c"  
    ],  
    "LoadBalancerNames": [],  
    "TargetGroupARNs": [],  
    "HealthCheckType": "EC2",  
    "HealthCheckGracePeriod": 0,  
    "Instances": [  
      {  
        "InstanceId": "i-088c57934a6449037",  
        "InstanceType": "t2.micro",  
        "AvailabilityZone": "us-west-2c",  
        "HealthStatus": "Healthy",  
        "LifecycleState": "InService",  
        "LaunchConfigurationName": "my-lc",  
        "ProtectedFromScaleIn": false  
      }  
    ],  
    "CreatedTime": "2023-10-28T02:39:22.152Z",  
    "SuspendedProcesses": [],  
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",  
    "EnabledMetrics": [],  
    "Tags": [],  
    "TerminationPolicies": [  
      "Default"  
    ]  
  }  
]
```

```
    ],  
    "NewInstancesProtectedFromScaleIn": false,  
    "ServiceLinkedRoleARN": "arn",  
    "TrafficSources": []  
  }  
]
```

詳細については、「コマンドラインインターフェイスユーザーガイド」の「[出力をフィルタリング AWS CLI](#)する」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAutoScalingGroups](#)」の「」を参照してください。

describe-auto-scaling-instances

次のコード例は、describe-auto-scaling-instances を使用する方法を示しています。

AWS CLI

例 1: 1 つまたは複数のインスタンスを記述するには

この例は、指定されたインスタンスを記述します。

```
aws autoscaling describe-auto-scaling-instances \  
  --instance-ids i-06905f55584de02da
```

出力:

```
{  
  "AutoScalingInstances": [  
    {  
      "InstanceId": "i-06905f55584de02da",  
      "InstanceType": "t2.micro",  
      "AutoScalingGroupName": "my-asg",  
      "AvailabilityZone": "us-west-2b",  
      "LifecycleState": "InService",  
      "HealthStatus": "HEALTHY",  
      "ProtectedFromScaleIn": false,  
      "LaunchTemplate": {  
        "LaunchTemplateId": "lt-1234567890abcde12",  
        "LaunchTemplateName": "my-launch-template",  
        "Version": "1"  
      }  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

例 2: 1 つまたは複数のインスタンスを記述するには

この例では、`--max-items` オプションを使用して、この呼び出しで返されるインスタンスの数を指定します。

```
aws autoscaling describe-auto-scaling-instances \  
  --max-items 1
```

出力に `NextToken` フィールドが含まれている場合は、さらに多くのインスタンスがあることを示しています。追加のインスタンスを取得するには、次のように、以降の呼び出しで `--starting-token` オプションを使用してこのフィールドの値を使用します。

```
aws autoscaling describe-auto-scaling-instances \  
  --starting-token Z3M3LMPEXAMPLE
```

出力例については、例 1 を参照してください。

例 3: 起動設定を使用するインスタンスを記述するには

この例では、`--query` オプションを使用して、起動設定を使用するインスタンスを記述します。

```
aws autoscaling describe-auto-scaling-instances \  
  --query 'AutoScalingInstances[?LaunchConfigurationName!=`null`]'
```

出力:

```
[  
  {  
    "InstanceId": "i-088c57934a6449037",  
    "InstanceType": "t2.micro",  
    "AutoScalingGroupName": "my-asg",  
    "AvailabilityZone": "us-west-2c",  
    "LifecycleState": "InService",  
    "HealthStatus": "HEALTHY",  
    "LaunchConfigurationName": "my-lc",  
    "ProtectedFromScaleIn": false  
  }  
]
```

```
]
```

詳細については、「コマンドラインインターフェイスユーザーガイド」の「[出力をフィルタリング AWS CLI](#)する」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAutoScalingInstances](#)」の「」を参照してください。

describe-auto-scaling-notification-types

次のコード例は、describe-auto-scaling-notification-types を使用する方法を示しています。

AWS CLI

使用可能な通知タイプを記述するには

この例では、使用可能な通知タイプについて説明します。

```
aws autoscaling describe-auto-scaling-notification-types
```

出力:

```
{
  "AutoScalingNotificationTypes": [
    "autoscaling:EC2_INSTANCE_LAUNCH",
    "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
    "autoscaling:EC2_INSTANCE_TERMINATE",
    "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
    "autoscaling:TEST_NOTIFICATION"
  ]
}
```

詳細については、「[Amazon Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループがスケールされたときに Amazon SNS通知を取得する](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAutoScalingNotificationTypes](#)」の「」を参照してください。

describe-instance-refreshes

次の例は、describe-instance-refreshes を使用する方法を説明しています。

AWS CLI

インスタンスの更新を記述するには

次のdescribe-instance-refreshes例では、ステータスメッセージや (利用可能な場合) ステータス理由など、指定された Auto Scaling グループのすべてのインスタンス更新リクエストの説明を返します。

```
aws autoscaling describe-instance-refreshes \  
  --auto-scaling-group-name my-asg
```

出力:

```
{  
  "InstanceRefreshes": [  
    {  
      "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b",  
      "AutoScalingGroupName": "my-asg",  
      "Status": "InProgress",  
      "StatusReason": "Waiting for instances to warm up before continuing. For  
example: 0e69cc3f05f825f4f is warming up.",  
      "EndTime": "2023-03-23T16:42:55Z",  
      "PercentageComplete": 0,  
      "InstancesToUpdate": 0,  
      "Preferences": {  
        "MinHealthyPercentage": 100,  
        "InstanceWarmup": 300,  
        "CheckpointPercentages": [  
          50  
        ],  
        "CheckpointDelay": 3600,  
        "SkipMatching": false,  
        "AutoRollback": true,  
        "ScaleInProtectedInstances": "Ignore",  
        "StandbyInstances": "Ignore"  
      }  
    },  
    {  
      "InstanceRefreshId": "dd7728d0-5bc4-4575-96a3-1b2c52bf8bb1",  
      "AutoScalingGroupName": "my-asg",  
      "Status": "Successful",  
      "EndTime": "2022-06-02T16:53:37Z",  
      "PercentageComplete": 100,  
    }  
  ]  
}
```

```

        "InstancesToUpdate": 0,
    "Preferences": {
        "MinHealthyPercentage": 90,
        "InstanceWarmup": 300,
        "SkipMatching": true,
        "AutoRollback": true,
        "ScaleInProtectedInstances": "Ignore",
        "StandbyInstances": "Ignore"
    }
}
]
}

```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[インスタンスの更新のステータスを確認する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInstanceRefreshes](#)」の「」を参照してください。

describe-launch-configurations

次の例は、describe-launch-configurations を使用する方法を説明しています。

AWS CLI

例 1: 指定された起動設定を記述するには

この例では、指定された起動設定について説明します。

```
aws autoscaling describe-launch-configurations \
  --launch-configuration-names my-launch-config
```

出力:

```
{
  "LaunchConfigurations": [
    {
      "LaunchConfigurationName": "my-launch-config",
      "LaunchConfigurationARN": "arn:aws:autoscaling:us-
west-2:123456789012:launchConfiguration:98d3b196-4cf9-4e88-8ca1-8547c24ced8b:launchConfigura
my-launch-config",

```

```
"ImageId": "ami-0528a5175983e7f28",
"KeyName": "my-key-pair-uswest2",
"SecurityGroups": [
  "sg-05eaec502fcdadc2e"
],
"ClassicLinkVPCSecurityGroups": [],
"UserData": "",
"InstanceType": "t2.micro",
"KernelId": "",
"RamdiskId": "",
"BlockDeviceMappings": [
  {
    "DeviceName": "/dev/xvda",
    "Ebs": {
      "SnapshotId": "snap-06c1606ba5ca274b1",
      "VolumeSize": 8,
      "VolumeType": "gp2",
      "DeleteOnTermination": true,
      "Encrypted": false
    }
  }
],
"InstanceMonitoring": {
  "Enabled": true
},
"CreatedTime": "2020-10-28T02:39:22.321Z",
"EbsOptimized": false,
"AssociatePublicIpAddress": true,
"MetadataOptions": {
  "HttpTokens": "required",
  "HttpPutResponseHopLimit": 1,
  "HttpEndpoint": "disabled"
}
}
]
```

例 2: 指定された数の起動設定を記述するには

特定の数の起動設定を返すには、`--max-items` オプションを使用します。

```
aws autoscaling describe-launch-configurations \
  --max-items 1
```

出力に NextToken フィールドが含まれている場合、起動設定がさらに増えます。追加の起動設定を取得するには、次のように、後続の呼び出しで `--starting-token` オプションでこのフィールドの値を使用します。

```
aws autoscaling describe-launch-configurations \  
  --starting-token Z3M3LMPEXAMPLE
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLaunchConfigurations](#)」の「」を参照してください。

describe-lifecycle-hook-types

次の例は、`describe-lifecycle-hook-types` を使用する方法を説明しています。

AWS CLI

使用可能なライフサイクルフックタイプを記述するには

この例では、使用可能なライフサイクルフックタイプについて説明します。

```
aws autoscaling describe-lifecycle-hook-types
```

出力:

```
{  
  "LifecycleHookTypes": [  
    "autoscaling:EC2_INSTANCE_LAUNCHING",  
    "autoscaling:EC2_INSTANCE_TERMINATING"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLifecycleHookTypes](#)」の「」を参照してください。

describe-lifecycle-hooks

次のコード例は、`describe-lifecycle-hooks` を使用する方法を示しています。

AWS CLI

ライフサイクルフックを記述するには

この例では、指定された Auto Scaling グループのライフサイクルフックについて説明します。

```
aws autoscaling describe-lifecycle-hooks \  
  --auto-scaling-group-name my-asg
```

出力:

```
{  
  "LifecycleHooks": [  
    {  
      "GlobalTimeout": 3000,  
      "HeartbeatTimeout": 30,  
      "AutoScalingGroupName": "my-asg",  
      "LifecycleHookName": "my-launch-hook",  
      "DefaultResult": "ABANDON",  
      "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING"  
    },  
    {  
      "GlobalTimeout": 6000,  
      "HeartbeatTimeout": 60,  
      "AutoScalingGroupName": "my-asg",  
      "LifecycleHookName": "my-termination-hook",  
      "DefaultResult": "CONTINUE",  
      "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLifecycleHooks](#)」の「」を参照してください。

describe-load-balancer-target-groups

次の例は、describe-load-balancer-target-groups を使用する方法を説明しています。

AWS CLI

Auto Scaling グループのロードバランサーターゲットグループを記述するには

この例では、指定された Auto Scaling グループにアタッチされたロードバランサーターゲットグループについて説明します。

```
aws autoscaling describe-load-balancer-target-groups \  
--auto-scaling-group-name my-asg
```

出力:

```
{  
  "LoadBalancerTargetGroups": [  
    {  
      "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",  
      "State": "Added"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLoadBalancerTargetGroups](#)」の「」を参照してください。

describe-load-balancers

次の例は、describe-load-balancers を使用する方法を説明しています。

AWS CLI

Auto Scaling グループの Classic Load Balancer を記述するには

この例では、指定された Auto Scaling グループの Classic Load Balancer について説明します。

```
aws autoscaling describe-load-balancers \  
--auto-scaling-group-name my-asg
```

出力:

```
{  
  "LoadBalancers": [  
    {  
      "State": "Added",  
      "LoadBalancerName": "my-load-balancer"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoadBalancers](#)」の「」を参照してください。

describe-metric-collection-types

次のコード例は、describe-metric-collection-types を使用する方法を示しています。

AWS CLI

使用可能なメトリクスコレクションタイプを記述するには

この例では、使用可能なメトリクスコレクションタイプについて説明します。

```
aws autoscaling describe-metric-collection-types
```

出力:

```
{
  "Metrics": [
    {
      "Metric": "GroupMinSize"
    },
    {
      "Metric": "GroupMaxSize"
    },
    {
      "Metric": "GroupDesiredCapacity"
    },
    {
      "Metric": "GroupInServiceInstances"
    },
    {
      "Metric": "GroupInServiceCapacity"
    },
    {
      "Metric": "GroupPendingInstances"
    },
    {
      "Metric": "GroupPendingCapacity"
    },
    {
      "Metric": "GroupTerminatingInstances"
    }
  ]
}
```

```
{
  "Metric": "GroupTerminatingCapacity"
},
{
  "Metric": "GroupStandbyInstances"
},
{
  "Metric": "GroupStandbyCapacity"
},
{
  "Metric": "GroupTotalInstances"
},
{
  "Metric": "GroupTotalCapacity"
}
],
"Granularities": [
  {
    "Granularity": "1Minute"
  }
]
}
```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループメトリクス](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeMetricCollectionTypes](#)」の「」を参照してください。

describe-notification-configurations

次の例は、describe-notification-configurations を使用する方法を説明しています。

AWS CLI

例 1: 指定されたグループの通知設定を記述するには

この例では、指定された Auto Scaling グループの通知設定について説明します。

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-asg
```

出力:


```
{
  "NotificationConfigurations": [
    {
      "AutoScalingGroupName": "my-asg",
      "NotificationType": "autoscaling:TEST_NOTIFICATION",
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"
    },
    {
      "AutoScalingGroupName": "my-asg",
      "NotificationType": "autoscaling:TEST_NOTIFICATION",
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"
    }
  ]
}
```

詳細については、[「Amazon Auto Scaling ユーザーガイド」の「Auto Scaling グループがスケールされたときに Amazon SNS通知を取得する」](#)を参照してください。 EC2 Auto Scaling

例 1: 指定された数の通知設定を記述するには

特定の数の通知設定を返すには、`max-items`パラメータを使用します。

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-auto-scaling-group \
  --max-items 1
```

出力:

```
{
  "NotificationConfigurations": [
    {
      "AutoScalingGroupName": "my-asg",
      "NotificationType": "autoscaling:TEST_NOTIFICATION",
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic-2"
    },
    {
      "AutoScalingGroupName": "my-asg",
      "NotificationType": "autoscaling:TEST_NOTIFICATION",
      "TopicARN": "arn:aws:sns:us-west-2:123456789012:my-sns-topic"
    }
  ]
}
```

出力に NextToken フィールドが含まれている場合、通知設定がさらに増えます。追加の通知設定を取得するには、次のように、後続の呼び出しで starting-token パラメータでこのフィールドの値を使用します。

```
aws autoscaling describe-notification-configurations \  
  --auto-scaling-group-name my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

詳細については、[「Amazon Auto Scaling ユーザーガイド」の「Auto Scaling グループがスケールされたときに Amazon SNS 通知を取得する」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeNotificationConfigurations](#)」の「」を参照してください。

describe-policies

次のコード例は、describe-policies を使用する方法を示しています。

AWS CLI

例 1: 指定されたグループのスケールリングポリシーを記述するには

この例では、指定された Auto Scaling グループのスケールリングポリシーについて説明します。

```
aws autoscaling describe-policies \  
  --auto-scaling-group-name my-asg
```

出力:

```
{  
  "ScalingPolicies": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "PolicyName": "alb1000-target-tracking-scaling-policy",  
      "PolicyARN": "arn:aws:autoscaling:us-west-2:123456789012:scalingPolicy:3065d9c8-9969-4bec-bb6a-3fbe5550fde6:autoScalingGroupName/my-asg:policyName/alb1000-target-tracking-scaling-policy",  
      "PolicyType": "TargetTrackingScaling",  
      "StepAdjustments": [],  
      "Alarms": [  
        {
```

```

        "AlarmName": "TargetTracking-my-asg-
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196",
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-
AlarmHigh-924887a9-12d7-4e01-8686-6f844d13a196"
    },
    {
        "AlarmName": "TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-
a010-c1aaa35da296",
        "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-f96f899d-b8e7-4d09-a010-
c1aaa35da296"
    }
],
"TargetTrackingConfiguration": {
    "PredefinedMetricSpecification": {
        "PredefinedMetricType": "ALBRequestCountPerTarget",
        "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-
alb-target-group/943f017f100becff"
    },
    "TargetValue": 1000.0,
    "DisableScaleIn": false
},
"Enabled": true
},
{
    "AutoScalingGroupName": "my-asg",
    "PolicyName": "cpu40-target-tracking-scaling-policy",
    "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:5fd26f71-39d4-4690-82a9-
b8515c45cdde:autoScalingGroupName/my-asg:policyName/cpu40-target-tracking-scaling-
policy",
    "PolicyType": "TargetTrackingScaling",
    "StepAdjustments": [],
    "Alarms": [
        {
            "AlarmName": "TargetTracking-my-asg-
AlarmHigh-139f9789-37b9-42ad-bea5-b5b147d7f473",
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-139f9789-37b9-42ad-bea5-
b5b147d7f473"
        },
        {

```

```
        "AlarmName": "TargetTracking-my-asg-AlarmLow-bd681c67-  
fc18-4c56-8468-fb8e413009c9",  
        "AlarmARN": "arn:aws:cloudwatch:us-  
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-bd681c67-fc18-4c56-8468-  
fb8e413009c9"  
    }  
],  
"TargetTrackingConfiguration": {  
    "PredefinedMetricSpecification": {  
        "PredefinedMetricType": "ASGAverageCPUUtilization"  
    },  
    "TargetValue": 40.0,  
    "DisableScaleIn": false  
},  
"Enabled": true  
}  
]  
}
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の[「動的スケーリング」](#)を参照してください。

例 2: 指定された名前のスケーリングポリシーを記述するには

特定のスケールリングポリシーを返すには、`--policy-names`オプションを使用します。

```
aws autoscaling describe-policies \  
  --auto-scaling-group-name my-asg \  
  --policy-names cpu40-target-tracking-scaling-policy
```

出力例については、例 1 を参照してください。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の[「動的スケーリング」](#)を参照してください。

例 3: スケーリングポリシーの数を記述するには

特定の数のポリシーを返すには、`--max-items`オプションを使用します。

```
aws autoscaling describe-policies \  
  --auto-scaling-group-name my-asg \  
  --max-items 1
```

出力例については、例 1 を参照してください。

出力に NextToken フィールドが含まれている場合は、このフィールドの値を後続の呼び出しで `--starting-token` オプションとともに使用して、追加のポリシーを取得します。

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg --starting-token Z3M3LMPEXAMPLE
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[動的スケーリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribePolicies](#)」の「」を参照してください。

describe-scaling-activities

次の例は、`describe-scaling-activities` を使用する方法を説明しています。

AWS CLI

例 1: 指定されたグループのスケーリングアクティビティを記述するには

この例は、指定された Auto Scaling グループのスケーリングアクティビティを記述します。

```
aws autoscaling describe-scaling-activities \  
--auto-scaling-group-name my-asg
```

出力:

```
{  
  "Activities": [  
    {  
      "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",  
      "Description": "Launching a new EC2 instance: i-0d44425630326060f",  
      "AutoScalingGroupName": "my-asg",  
      "Cause": "At 2020-10-30T19:35:51Z a user request update of  
AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired  
capacity from 0 to 16. At 2020-10-30T19:36:07Z an instance was started in response  
to a difference between desired and actual capacity, increasing the capacity from 0  
to 16.",  
      "StartTime": "2020-10-30T19:36:09.766Z",  
      "EndTime": "2020-10-30T19:36:41Z",
```

```

        "StatusCode": "Successful",
        "Progress": 100,
        "Details": "{\"Subnet ID\": \"subnet-5ea0c127\", \"Availability Zone\":
\\\"us-west-2b\\\"}"
    }
]
}

```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループのスケールリングアクティビティを検証する](#)」を参照してください。 EC2 Auto Scaling

例 2: 削除されたグループのスケールリングアクティビティを記述するには

Auto Scaling グループが削除された後にスケールリングアクティビティを記述するには、`--include-deleted-groups` オプションを追加します。

```

aws autoscaling describe-scaling-activities \
  --auto-scaling-group-name my-asg \
  --include-deleted-groups

```

出力:

```

{
  "Activities": [
    {
      "ActivityId": "e1f5de0e-f93e-1417-34ac-092a76fba220",
      "Description": "Launching a new EC2 instance. Status Reason: Your Spot
request price of 0.001 is lower than the minimum required Spot request fulfillment
price of 0.0031. Launching EC2 instance failed.",
      "AutoScalingGroupName": "my-asg",
      "Cause": "At 2021-01-13T20:47:24Z a user request update of
AutoScalingGroup constraints to min: 1, max: 5, desired: 3 changing the desired
capacity from 0 to 3. At 2021-01-13T20:47:27Z an instance was started in response
to a difference between desired and actual capacity, increasing the capacity from 0
to 3.",
      "StartTime": "2021-01-13T20:47:30.094Z",
      "EndTime": "2021-01-13T20:47:30Z",
      "StatusCode": "Failed",
      "StatusMessage": "Your Spot request price of 0.001 is lower than the
minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance
failed.",
      "Progress": 100,
    }
  ]
}

```

```

        "Details": "{ \"Subnet ID\": \"subnet-5ea0c127\", \"Availability Zone\":
        \"us-west-2b\" }",
        "AutoScalingGroupState": "Deleted",
        "AutoScalingGroupARN": "arn:aws:autoscaling:us-
        west-2:123456789012:autoScalingGroup:283179a2-
        f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    }
]
}

```

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling のトラブルシューティング」](#)を参照してください。 EC2 Auto Scaling

例 3: 指定された数のスケーリングアクティビティを記述するには

特定の数のアクティビティを返すには、`--max-items` オプションを使用します。

```

aws autoscaling describe-scaling-activities \
  --max-items 1

```

出力:

```

{
  "Activities": [
    {
      "ActivityId": "f9f2d65b-f1f2-43e7-b46d-d86756459699",
      "Description": "Launching a new EC2 instance: i-0d44425630326060f",
      "AutoScalingGroupName": "my-asg",
      "Cause": "At 2020-10-30T19:35:51Z a user request update of
      AutoScalingGroup constraints to min: 0, max: 16, desired: 16 changing the desired
      capacity from 0 to 16. At 2020-10-30T19:36:07Z an instance was started in response
      to a difference between desired and actual capacity, increasing the capacity from 0
      to 16.",
      "StartTime": "2020-10-30T19:36:09.766Z",
      "EndTime": "2020-10-30T19:36:41Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{ \"Subnet ID\": \"subnet-5ea0c127\", \"Availability Zone\":
      \"us-west-2b\" }"
    }
  ]
}

```

出力に NextToken フィールドが含まれている場合は、さらに多くのアクティビティがあることを示しています。追加のアクティビティを取得するには、次のように、以降の呼び出しで `--starting-token` オプションを使用してこのフィールドの値を使用します。

```
aws autoscaling describe-scaling-activities \  
  --starting-token Z3M3LMPEXAMPLE
```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループのスケールリングアクティビティを検証する](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeScalingActivities](#)」の「」を参照してください。

describe-scaling-process-types

次の例は、`describe-scaling-process-types` を使用する方法を説明しています。

AWS CLI

使用可能なプロセスタイプを記述するには

この例では、使用可能なプロセスタイプについて説明します。

```
aws autoscaling describe-scaling-process-types
```

出力:

```
{  
  "Processes": [  
    {  
      "ProcessName": "AZRebalance"  
    },  
    {  
      "ProcessName": "AddToLoadBalancer"  
    },  
    {  
      "ProcessName": "AlarmNotification"  
    },  
    {  
      "ProcessName": "HealthCheck"  
    }  
  ]  
}
```



```
{
  "ProcessName": "InstanceRefresh"
},
{
  "ProcessName": "Launch"
},
{
  "ProcessName": "ReplaceUnhealthy"
},
{
  "ProcessName": "ScheduledActions"
},
{
  "ProcessName": "Terminate"
}
]
```

詳細については、[「Amazon Auto Scaling ユーザーガイド」](#)の「[スケーリングプロセスの一時停止と再開](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeScalingProcessTypes](#)」の「」を参照してください。

describe-scheduled-actions

次の例は、describe-scheduled-actions を使用する方法を説明しています。

AWS CLI

例 1: スケジュールされたすべてのアクションを記述するには

この例では、スケジュールされたすべてのアクションについて説明します。

```
aws autoscaling describe-scheduled-actions
```

出力:

```
{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
```

```

    "ScheduledActionName": "my-recurring-action",
    "Recurrence": "30 0 1 1,6,12 *",
    "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
    "StartTime": "2023-12-01T04:00:00Z",
    "Time": "2023-12-01T04:00:00Z",
    "MinSize": 1,
    "MaxSize": 6,
    "DesiredCapacity": 4,
    "TimeZone": "America/New_York"
  }
]
}

```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「スケジュールされたスケールリング」](#) を参照してください。

例 2: 指定されたグループのスケジュールされたアクションを記述するには

特定の Auto Scaling グループのスケジュールされたアクションを記述するには、`--auto-scaling-group-name` オプションを使用します。

```

aws autoscaling describe-scheduled-actions \
  --auto-scaling-group-name my-asg

```

出力:

```

{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
      "ScheduledActionName": "my-recurring-action",
      "Recurrence": "30 0 1 1,6,12 *",
      "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
      "StartTime": "2023-12-01T04:00:00Z",
      "Time": "2023-12-01T04:00:00Z",
      "MinSize": 1,
      "MaxSize": 6,
      "DesiredCapacity": 4,
    }
  ]
}

```

```

        "TimeZone": "America/New_York"
    }
]
}

```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「スケジュールされたスケールリング」](#) を参照してください。

例 3: 指定されたスケジュールされたアクションを記述するには

特定のスケジュールされたアクションを記述するには、`--scheduled-action-names` オプションを使用します。

```

aws autoscaling describe-scheduled-actions \
  --scheduled-action-names my-recurring-action

```

出力:

```

{
  "ScheduledUpdateGroupActions": [
    {
      "AutoScalingGroupName": "my-asg",
      "ScheduledActionName": "my-recurring-action",
      "Recurrence": "30 0 1 1,6,12 *",
      "ScheduledActionARN": "arn:aws:autoscaling:us-west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
      "StartTime": "2023-12-01T04:00:00Z",
      "Time": "2023-12-01T04:00:00Z",
      "MinSize": 1,
      "MaxSize": 6,
      "DesiredCapacity": 4,
      "TimeZone": "America/New_York"
    }
  ]
}

```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「スケジュールされたスケールリング」](#) を参照してください。

例 4: 指定された開始時刻でスケジュールされたアクションを記述するには

特定の時間に開始されるスケジュールされたアクションを記述するには、`--start-time`オプションを使用します。

```
aws autoscaling describe-scheduled-actions \  
  --start-time "2023-12-01T04:00:00Z"
```

出力:

```
{  
  "ScheduledUpdateGroupActions": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "ScheduledActionName": "my-recurring-action",  
      "Recurrence": "30 0 1 1,6,12 *",  
      "ScheduledActionARN": "arn:aws:autoscaling:us-  
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-  
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",  
      "StartTime": "2023-12-01T04:00:00Z",  
      "Time": "2023-12-01T04:00:00Z",  
      "MinSize": 1,  
      "MaxSize": 6,  
      "DesiredCapacity": 4,  
      "TimeZone": "America/New_York"  
    }  
  ]  
}
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「スケジュールされたスケーリング」](#) を参照してください。

例 5: 指定した時間に終了するスケジュールされたアクションを記述するには

特定の時間に終了するスケジュールされたアクションを記述するには、`--end-time`オプションを使用します。

```
aws autoscaling describe-scheduled-actions \  
  --end-time "2023-12-01T04:00:00Z"
```

出力:

```
{
```

```
"ScheduledUpdateGroupActions": [  
  {  
    "AutoScalingGroupName": "my-asg",  
    "ScheduledActionName": "my-recurring-action",  
    "Recurrence": "30 0 1 1,6,12 *",  
    "ScheduledActionARN": "arn:aws:autoscaling:us-  
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-  
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",  
    "StartTime": "2023-12-01T04:00:00Z",  
    "Time": "2023-12-01T04:00:00Z",  
    "MinSize": 1,  
    "MaxSize": 6,  
    "DesiredCapacity": 4,  
    "TimeZone": "America/New_York"  
  }  
]  
}
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「スケジュールされたスケールリング」](#) を参照してください。

例 6: 指定された数のスケジュールされたアクションを記述するには

特定の数のスケジュールされたアクションを返すには、`--max-items` オプションを使用します。

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg \  
  --max-items 1
```

出力:

```
{  
  "ScheduledUpdateGroupActions": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "ScheduledActionName": "my-recurring-action",  
      "Recurrence": "30 0 1 1,6,12 *",  
      "ScheduledActionARN": "arn:aws:autoscaling:us-  
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-  
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",  
      "StartTime": "2023-12-01T04:00:00Z",
```

```
        "Time": "2023-12-01T04:00:00Z",
        "MinSize": 1,
        "MaxSize": 6,
        "DesiredCapacity": 4,
        "TimeZone": "America/New_York"
    }
]
}
```

出力に NextToken フィールドが含まれている場合、さらにスケジュールされたアクションがあります。追加のスケジュールされたアクションを取得するには、次のように、後続の呼び出しで `--starting-token` オプションでこのフィールドの値を使用します。

```
aws autoscaling describe-scheduled-actions \
  --auto-scaling-group-name my-asg \
  --starting-token Z3M3LMPEXAMPLE
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[スケジュールされたスケールリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeScheduledActions](#)」の「」を参照してください。

describe-tags

次の例は、`describe-tags` を使用する方法を説明しています。

AWS CLI

すべてのタグを記述するには

この例では、すべてのタグについて説明します。

```
aws autoscaling describe-tags
```

出力:

```
{
  "Tags": [
    {
      "ResourceType": "auto-scaling-group",
```

```
    "ResourceId": "my-asg",
    "PropagateAtLaunch": true,
    "Value": "Research",
    "Key": "Dept"
  },
  {
    "ResourceType": "auto-scaling-group",
    "ResourceId": "my-asg",
    "PropagateAtLaunch": true,
    "Value": "WebServer",
    "Key": "Role"
  }
]
```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスのタグ付け](#)」を参照してください。 EC2 Auto Scaling

例 2: 指定されたグループのタグを記述するには

特定の Auto Scaling グループのタグを記述するには、`--filters` オプションを使用します。

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスのタグ付け](#)」を参照してください。 EC2 Auto Scaling

例 3: 指定された数のタグを記述するには

特定の数のタグを返すには、`--max-items` オプションを使用します。

```
aws autoscaling describe-tags \  
  --max-items 1
```

出力に `NextToken` フィールドが含まれている場合、タグがさらに追加されます。追加のタグを取得するには、次のように、後続の呼び出しで `--starting-token` オプションでこのフィールドの値を使用します。

```
aws autoscaling describe-tags \  
  --filters Name=auto-scaling-group,Values=my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスのタグ付け](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

describe-termination-policy-types

次の例は、describe-termination-policy-types を使用する方法を説明しています。

AWS CLI

使用可能な終了ポリシータイプを記述するには

この例では、使用可能な終了ポリシータイプについて説明します。

```
aws autoscaling describe-termination-policy-types
```

出力:

```
{
  "TerminationPolicyTypes": [
    "AllocationStrategy",
    "ClosestToNextInstanceHour",
    "Default",
    "NewestInstance",
    "OldestInstance",
    "OldestLaunchConfiguration",
    "OldestLaunchTemplate"
  ]
}
```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[スケールイン中に終了する Auto Scaling インスタンスの制御](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTerminationPolicyTypes](#)」の「」を参照してください。

describe-warm-pool

次の例は、describe-warm-pool を使用する方法を説明しています。

AWS CLI

ウォームプールを記述するには

この例では、指定された Auto Scaling グループのウォームプールについて説明します。

```
aws autoscaling describe-warm-pool \  
  --auto-scaling-group-name my-asg
```

出力:

```
{  
  "WarmPoolConfiguration": {  
    "MinSize": 2,  
    "PoolState": "Stopped"  
  },  
  "Instances": [  
    {  
      "InstanceId": "i-070a5bbc7e7f40dc5",  
      "InstanceType": "t2.micro",  
      "AvailabilityZone": "us-west-2c",  
      "LifecycleState": "Warmed:Pending",  
      "HealthStatus": "Healthy",  
      "LaunchTemplate": {  
        "LaunchTemplateId": "lt-00a731f6e9fa48610",  
        "LaunchTemplateName": "my-template-for-auto-scaling",  
        "Version": "6"  
      }  
    },  
    {  
      "InstanceId": "i-0b52f061814d3bd2d",  
      "InstanceType": "t2.micro",  
      "AvailabilityZone": "us-west-2b",  
      "LifecycleState": "Warmed:Pending",  
      "HealthStatus": "Healthy",  
      "LaunchTemplate": {  
        "LaunchTemplateId": "lt-00a731f6e9fa48610",  
        "LaunchTemplateName": "my-template-for-auto-scaling",  
        "Version": "6"  
      }  
    }  
  ]  
}
```

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling のウォームプール」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeWarmPool](#)」の「」を参照してください。

detach-instances

次の例は、detach-instances を使用する方法を説明しています。

AWS CLI

Auto Scaling グループからインスタンスをデタッチするには

この例では、指定されたインスタンスを指定された Auto Scaling グループからデタッチします。

```
aws autoscaling detach-instances \  
  --instance-ids i-030017cfa84b20135 \  
  --auto-scaling-group-name my-asg \  
  --should-decrement-desired-capacity
```

出力:

```
{  
  "Activities": [  
    {  
      "ActivityId": "5091cb52-547a-47ce-a236-c9ccbc2cb2c9",  
      "AutoScalingGroupName": "my-asg",  
      "Description": "Detaching EC2 instance: i-030017cfa84b20135",  
      "Cause": "At 2020-10-31T17:35:04Z instance i-030017cfa84b20135 was  
detached in response to a user request, shrinking the capacity from 2 to 1.",  
      "StartTime": "2020-04-12T15:02:16.179Z",  
      "StatusCode": "InProgress",  
      "Progress": 50,  
      "Details": "{\"Subnet ID\": \"subnet-6194ea3b\", \"Availability Zone\":  
\"us-west-2c\"}"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachInstances](#)」の「」を参照してください。

detach-load-balancer-target-groups

次のコード例は、detach-load-balancer-target-groups を使用方法を示しています。

AWS CLI

Auto Scaling グループからロードバランサーターゲットグループをデタッチするには

この例では、指定された Auto Scaling グループから指定されたロードバランサーターゲットグループをデタッチします。

```
aws autoscaling detach-load-balancer-target-groups \  
  --auto-scaling-group-name my-asg \  
  --target-group-arns arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

このコマンドは出力を生成しません

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループにロードバランサーをアタッチする](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「[コマンドリファレンスDetachLoadBalancerTargetGroups](#)」の「」を参照してください。

detach-load-balancers

次のコード例は、detach-load-balancers を使用方法を示しています。

AWS CLI

Classic Load Balancer を Auto Scaling グループからデタッチするには

この例では、指定された Classic Load Balancer を指定された Auto Scaling グループからデタッチします。

```
aws autoscaling detach-load-balancers \  
  --load-balancer-names my-load-balancer \  
  --auto-scaling-group-name my-asg
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループにロードバランサーをアタッチする](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachLoadBalancers](#)」の「」を参照してください。

disable-metrics-collection

次のコード例は、`disable-metrics-collection` を使用する方法を示しています。

AWS CLI

Auto Scaling グループのメトリクス収集を無効にするには

この例では、指定された Auto Scaling グループの `GroupDesiredCapacity` メトリクスの収集を無効にします。

```
aws autoscaling disable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスの CloudWatch メトリクスのモニタリング](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableMetricsCollection](#)」の「」を参照してください。

enable-metrics-collection

次のコード例は、`enable-metrics-collection` を使用する方法を示しています。

AWS CLI

例 1: Auto Scaling グループのメトリクス収集を有効にするには

この例は、指定された Auto Scaling グループのデータの収集を有効にします。

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --granularity "1Minute"
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスの CloudWatch メトリクスのモニタリング](#)」を参照してください。 EC2 Auto Scaling

例 2: Auto Scaling グループの指定されたメトリクスのデータを収集するには

特定のメトリクスのデータを収集するには、`--metrics` オプションを使用します。

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity --granularity "1Minute"
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループとインスタンスの CloudWatch メトリクスのモニタリング](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableMetricsCollection](#)」の「」を参照してください。

enter-standby

次の例は、`enter-standby` を使用する方法を説明しています。

AWS CLI

インスタンスをスタンバイモードに移行するには

この例では、指定されたインスタンスをスタンバイモードにします。これは、現在稼働中のインスタンスの更新またはトラブルシューティングに役立ちます。

```
aws autoscaling enter-standby \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg \  
  --should-decrement-desired-capacity
```

出力:

```
{  
  "Activities": [  
    {  
      "ActivityId": "ffa056b4-6ed3-41ba-ae7c-249dfae6eba1",  
      "AutoScalingGroupName": "my-asg",
```

```

    "Description": "Moving EC2 instance to Standby: i-061c63c5eb45f0416",
    "Cause": "At 2020-10-31T20:31:00Z instance i-061c63c5eb45f0416 was moved
to standby in response to a user request, shrinking the capacity from 1 to 0.",
    "StartTime": "2020-10-31T20:31:00.949Z",
    "StatusCode": "InProgress",
    "Progress": 50,
    "Details": "{\"Subnet ID\": \"subnet-6194ea3b\", \"Availability Zone\":
\\\"us-west-2c\\\"}"
  }
]
}

```

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」](#)の「[Amazon Auto Scaling インスタンスのライフサイクル](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[EnterStandby](#)」の「」を参照してください。

execute-policy

次の例は、execute-policy を使用する方法を説明しています。

AWS CLI

スケーリングポリシーを実行するには

この例では、指定された Auto Scaling グループに my-step-scale-out-policy という名前のスケーリングポリシーを実行します。

```

aws autoscaling execute-policy \
  --auto-scaling-group-name my-asg \
  --policy-name my-step-scale-out-policy \
  --metric-value 95 \
  --breach-threshold 80

```

このコマンドでは何も出力されません。

詳細については、「[Amazon EC2 Auto Scaling ユーザーガイド](#)」の「[ステップポリシーとシンプルなスケーリングポリシー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ExecutePolicy](#)」の「」を参照してください。

exit-standby

次の例は、exit-standby を使用する方法を説明しています。

AWS CLI

インスタンスをスタンバイモードから移動するには

この例では、指定されたインスタンスをスタンバイモードから遠ざけます。

```
aws autoscaling exit-standby \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg
```

出力:

```
{  
  "Activities": [  
    {  
      "ActivityId": "142928e1-a2dc-453a-9b24-b85ad6735928",  
      "AutoScalingGroupName": "my-asg",  
      "Description": "Moving EC2 instance out of Standby:  
i-061c63c5eb45f0416",  
      "Cause": "At 2020-10-31T20:32:50Z instance i-061c63c5eb45f0416 was moved  
out of standby in response to a user request, increasing the capacity from 0 to  
1.",  
      "StartTime": "2020-10-31T20:32:50.222Z",  
      "StatusCode": "PreInService",  
      "Progress": 30,  
      "Details": "{\"Subnet ID\": \"subnet-6194ea3b\", \"Availability Zone\":  
\"us-west-2c\"}"  
    }  
  ]  
}
```

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループからインスタンスを一時的に削除する](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「[コマンドリファレンスExitStandby](#)」の「」を参照してください。

put-lifecycle-hook

次の例は、put-lifecycle-hook を使用する方法を説明しています。

AWS CLI

例 1: ライフサイクルフックを作成するには

この例では、新しく起動したインスタンスで を呼び出すライフサイクルフックを作成し、タイムアウトは 4800 秒です。これは、ユーザーデータスクリプトが終了するまでインスタンスを待機状態に保つ場合や、 を使用して AWS Lambda 関数を呼び出す場合に役立ちます EventBridge。

```
aws autoscaling put-lifecycle-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-hook-name my-launch-hook \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \  
  --heartbeat-timeout 4800
```

このコマンドでは何も出力されません。同じ名前のライフサイクルフックが既に存在する場合、新しいライフサイクルフックによって上書きされます。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling ライフサイクルフック」](#)を参照してください。 EC2 Auto Scaling

例 2: インスタンス状態遷移を通知する Amazon SNS メールメッセージを送信するには

この例では、インスタンスの起動時に通知を受信するために使用する Amazon SNS トピックと IAM ロールを含むライフサイクルフックを作成します。

```
aws autoscaling put-lifecycle-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-hook-name my-launch-hook \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \  
  --notification-target-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \  
  --role-arn arn:aws:iam::123456789012:role/my-auto-scaling-role
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling ライフサイクルフック」](#)を参照してください。 EC2 Auto Scaling

例 3: Amazon SQS キューにメッセージを公開するには

この例では、メタデータを含むメッセージを指定された Amazon SQS キューに発行するライフサイクルフックを作成します。

```
aws autoscaling put-lifecycle-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-hook-name my-launch-hook \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \  
  --notification-target-arn arn:aws:sqs:us-west-2:123456789012:my-sqs-queue \  
  --role-arn arn:aws:iam::123456789012:role/my-notification-role \  
  --notification-metadata "SQS message metadata"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling ライフサイクルフック」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLifecycleHook](#)」の「」を参照してください。

put-notification-configuration

次の例は、put-notification-configuration を使用する方法を説明しています。

AWS CLI

通知を追加するには

この例では、指定された Auto Scaling グループに指定された通知を追加します。

```
aws autoscaling put-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \  
  --notification-type autoscaling:TEST_NOTIFICATION
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Auto Scaling ユーザーガイド」の「Auto Scaling グループがスケールされたときに Amazon SNS 通知を取得する」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[PutNotificationConfiguration](#)」の「」を参照してください。

put-scaling-policy

次の例は、put-scaling-policy を使用する方法を説明しています。

AWS CLI

Auto Scaling グループにターゲット追跡スケーリングポリシーを追加するには

次のput-scaling-policy例では、ターゲット追跡スケーリングポリシーを指定された Auto Scaling グループに適用します。出力には、ユーザーに代わって作成された 2 つの CloudWatch アラームの ARNs と名前が含まれます。同じ名前のスケーリングポリシーが既に存在する場合、新しいスケーリングポリシーによって上書きされます。

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \  
  --policy-name alb1000-target-tracking-scaling-policy \  
  --policy-type TargetTrackingScaling \  
  --target-tracking-configuration file://config.json
```

config.json の内容:

```
{  
  "TargetValue": 1000.0,  
  "PredefinedMetricSpecification": {  
    "PredefinedMetricType": "ALBRequestCountPerTarget",  
    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-  
group/943f017f100becff"  
  }  
}
```

出力:

```
{  
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:228f02c2-  
c665-4bfd-aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/alb1000-target-  
tracking-scaling-policy",  
  "Alarms": [  
    {  
      "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-  
my-asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",  
      "AlarmName": "TargetTracking-my-asg-AlarmHigh-  
fc0e4183-23ac-497e-9992-691c9980c38e"  
    },  
  ],  
}
```

```
{
  "AlarmARN": "arn:aws:cloudwatch:region:account-id:alarm:TargetTracking-
my-asg-AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",
  "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
}
]
```

その他の例については、[「Amazon Auto Scaling ユーザーガイド」の AWS 「コマンドライン インターフェイス \(AWS CLI\) のスケーリングポリシーの例」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[PutScalingPolicy](#)」の「」を参照してください。

put-scheduled-update-group-action

次の例は、put-scheduled-update-group-action を使用する方法を説明しています。

AWS CLI

例 1: スケジュールされたアクションを Auto Scaling グループに追加するには

この例では、指定された Auto Scaling グループに指定されたスケジュール済みアクションを追加します。

```
aws autoscaling put-scheduled-update-group-action \
  --auto-scaling-group-name my-asg \
  --scheduled-action-name my-scheduled-action \
  --start-time "2023-05-12T08:00:00Z" \
  --min-size 2 \
  --max-size 6 \
  --desired-capacity 4
```

このコマンドでは何も出力されません。同じ名前のスケジュールされたアクションが既に存在する場合、新しいスケジュールされたアクションによって上書きされます。

その他の例については、「Amazon EC2 Auto Scaling ユーザーガイド」の[「スケジュールされたスケーリング」](#)を参照してください。

例 2: 定期的なスケジュールを指定するには

この例では、毎年 1 月、6 月、12 月 1 日の 00:30 に実行される予定の定期的なスケジュールに合わせてスケールリングするスケジュールされたアクションを作成します。

```
aws autoscaling put-scheduled-update-group-action \  
  --auto-scaling-group-name my-asg \  
  --scheduled-action-name my-recurring-action \  
  --recurrence "30 0 1 1,6,12 *" \  
  --min-size 2 \  
  --max-size 6 \  
  --desired-capacity 4
```

このコマンドでは何も出力されません。同じ名前のスケジュールされたアクションが既に存在する場合、新しいスケジュールされたアクションによって上書きされます。

その他の例については、「Amazon EC2 Auto Scaling ユーザーガイド」の[「スケジュールされたスケールリング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutScheduledUpdateGroupAction](#)」の「」を参照してください。

put-warm-pool

次の例は、put-warm-pool を使用する方法を説明しています。

AWS CLI

ウォームプールを作成するには

次の例では、指定された Auto Scaling グループのウォームプールを作成します。

```
aws autoscaling put-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --min-size 2
```

このコマンドでは何も出力されません。ウォームプールが既に存在する場合は、更新されます。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」](#)の[「Amazon Auto Scaling のウォームプール」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[PutWarmPool](#)」の「」を参照してください。

record-lifecycle-action-heartbeat

次の例は、record-lifecycle-action-heartbeat を使用する方法を説明しています。

AWS CLI

ライフサイクルアクションハートビートを記録するには

この例では、ライフサイクルアクションのハートビートを記録して、インスタンスを保留状態に保ちます。

```
aws autoscaling record-lifecycle-action-heartbeat \  
  --lifecycle-hook-name my-launch-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」の「Amazon Auto Scaling ライフサイクルフック」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[RecordLifecycleActionHeartbeat](#)」の「」を参照してください。

resume-processes

次のコード例は、resume-processes を使用する方法を示しています。

AWS CLI

中断されたプロセスを再開するには

この例では、指定された Auto Scaling グループの指定された一時停止スケーリングプロセスを再開します。

```
aws autoscaling resume-processes \  
  --auto-scaling-group-name my-asg \  
  --scaling-processes AlarmNotification
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Auto Scaling ユーザーガイド」の「スケーリングプロセスの一時停止と再開」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[ResumeProcesses](#)」の「」を参照してください。

rollback-instance-refresh

次のコード例は、rollback-instance-refresh を使用する方法を示しています。

AWS CLI

インスタンスの更新をロールバックするには

次のrollback-instance-refresh例では、指定された Auto Scaling グループの進行中のインスタンスの更新をロールバックします。

```
aws autoscaling rollback-instance-refresh \  
  --auto-scaling-group-name my-asg
```

出力:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[ロールバックによる変更の元に戻す](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RollbackInstanceRefresh](#)」の「」を参照してください。

set-desired-capacity

次のコード例は、set-desired-capacity を使用する方法を示しています。

AWS CLI

Auto Scaling グループの希望する容量を設定するには

この例は、指定された Auto Scaling グループの希望する容量を設定します。

```
aws autoscaling set-desired-capacity \  
  --auto-scaling-group-name my-asg \  
  --desired-capacity 1
```

```
--desired-capacity 2 \  
--honor-cooldown
```

正常に完了すると、このコマンドはプロンプトに戻ります。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetDesiredCapacity](#)」の「」を参照してください。

set-instance-health

次の例は、set-instance-health を使用する方法を説明しています。

AWS CLI

インスタンスのヘルスステータスを設定するには

この例では、指定されたインスタンスのヘルスステータスを に設定しますUnhealthy。

```
aws autoscaling set-instance-health \  
  --instance-id i-061c63c5eb45f0416 \  
  --health-status Unhealthy
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetInstanceHealth](#)」の「」を参照してください。

set-instance-protection

次のコード例は、set-instance-protection を使用する方法を示しています。

AWS CLI

例 1: インスタンスのインスタンス保護設定を有効にするには

この例では、指定されたインスタンスのインスタンス保護を有効にします。

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg --protected-from-scale-in
```

このコマンドでは何も出力されません。

例 2: インスタンスのインスタンス保護設定を無効にするには

この例では、指定されたインスタンスのインスタンス保護を無効にします。

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg \  
  --no-protected-from-scale-in
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetInstanceProtection](#)」の「」を参照してください。

start-instance-refresh

次の例は、start-instance-refresh を使用する方法を説明しています。

AWS CLI

例 1: コマンドラインパラメータを使用してインスタンスの更新を開始するには

次のstart-instance-refresh例では、コマンドライン引数を使用してインスタンスの更新を開始します。オプションのpreferencesパラメータは、60秒InstanceWarmupのと50パーセントMinHealthyPercentageのを指定します。

```
aws autoscaling start-instance-refresh \  
  --auto-scaling-group-name my-asg \  
  --preferences '{"InstanceWarmup": 60, "MinHealthyPercentage": 50}'
```

出力:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[インスタンスの更新を開始する](#)」を参照してください。

例 2: JSON ファイルを使用してインスタンスの更新を開始するには

次のstart-instance-refresh例では、JSONファイルを使用してインスタンスの更新を開始します。次の例に示すように、Auto Scaling グループを指定し、JSONファイルで必要な設定と設定を定義できます。

```
aws autoscaling start-instance-refresh \  
  --cli-input-json file://config.json
```

config.json の内容:

```
{  
  "AutoScalingGroupName": "my-asg",  
  "DesiredConfiguration": {  
    "LaunchTemplate": {  
      "LaunchTemplateId": "lt-068f72b729example",  
      "Version": "$Default"  
    }  
  },  
  "Preferences": {  
    "InstanceWarmup": 60,  
    "MinHealthyPercentage": 50,  
    "AutoRollback": true,  
    "ScaleInProtectedInstances": Ignore,  
    "StandbyInstances": Terminate  
  }  
}
```

出力:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「インスタンスの更新を開始する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartInstanceRefresh](#)」の「」を参照してください。

suspend-processes

次の例は、suspend-processes を使用する方法を説明しています。

AWS CLI

Auto Scaling プロセスを一時停止するには

この例では、指定された Auto Scaling グループの指定されたスケーリングプロセスを一時停止します。

```
aws autoscaling suspend-processes \  
  --auto-scaling-group-name my-asg \  
  --scaling-processes AlarmNotification
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Auto Scaling ユーザーガイド」の「スケーリングプロセスの一時停止と再開」](#)を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[SuspendProcesses](#)」の「」を参照してください。

terminate-instance-in-auto-scaling-group

次のコード例は、`terminate-instance-in-auto-scaling-group` を使用する方法を示しています。

AWS CLI

Auto Scaling グループのインスタンスを終了するには

この例は、グループのサイズを更新せずに、指定された Auto Scaling グループの指定されたインスタンスを終了します。Amazon EC2 Auto Scaling は、指定されたインスタンスが終了した後に代替インスタンスを起動します。

```
aws autoscaling terminate-instance-in-auto-scaling-group \  
  --instance-id i-061c63c5eb45f0416 \  
  --no-should-decrement-desired-capacity
```

出力:

```
{  
  "Activities": [  
    {  
      "ActivityId": "8c35d601-793c-400c-fcd0-f64a27530df7",
```

```

    "AutoScalingGroupName": "my-asg",
    "Description": "Terminating EC2 instance: i-061c63c5eb45f0416",
    "Cause": "",
    "StartTime": "2020-10-31T20:34:25.680Z",
    "StatusCode": "InProgress",
    "Progress": 0,
    "Details": "{\"Subnet ID\": \"subnet-6194ea3b\", \"Availability Zone\": \"us-west-2c\"}"
  }
]
}

```

- API 詳細については、AWS CLI 「[コマンドリファレンス](#) [TerminateInstanceInAutoScalingGroup](#)」の「」を参照してください。

update-auto-scaling-group

次のコード例は、update-auto-scaling-group を使用する方法を示しています。

AWS CLI

例 1: Auto Scaling グループのサイズ制限を更新するには

この例は、最小サイズが 2、最大サイズが 10 で、指定された Auto Scaling グループを更新します。

```

aws autoscaling update-auto-scaling-group \
  --auto-scaling-group-name my-asg \
  --min-size 2 \
  --max-size 10

```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループの容量制限の設定](#)」を参照してください。 EC2 Auto Scaling

例 2: Elastic Load Balancing ヘルスチェックを追加し、使用するアベイラビリティゾーンとサブネットを指定するには

この例は、指定された Auto Scaling グループを更新して、Elastic Load Balancing のヘルスチェックを追加します。このコマンドは、IDs 複数のアベイラビリティゾーンのサブネットのリスト --vpc-zone-identifier を使用して の値も更新します。

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --health-check-type ELB \  
  --health-check-grace-period 600 \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon EC2 Auto Scaling ユーザーガイド」のElastic Load Balancing](#)と「Amazon Auto Scaling」を参照してください。 EC2 Auto Scaling

例 3: プレイACEMENTグループと終了ポリシーを更新するには

この例は、プレイACEMENTグループと終了ポリシーを更新します。

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --placement-group my-placement-group \  
  --termination-policies "OldestInstance"
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループ](#)」を参照してください。 EC2 Auto Scaling

例 4: 起動テンプレートの最新バージョンを使用するには

この例は、最新の起動テンプレートバージョンを使用するように、指定された Auto Scaling グループを更新します。

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「[テンプレートを起動する](#)」を参照してください。

例 5: 特定のバージョンの起動テンプレートを使用するには

この例は、最新バージョンやデフォルトバージョンではなく、指定された起動テンプレートのバージョンを使用するように、指定された Auto Scaling グループを更新します。

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

このコマンドでは何も出力されません。

詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の [「テンプレートを起動する」](#) を参照してください。

例 6: 混合インスタンスポリシーを定義し、容量のリバランスを有効にするには

この例は、指定された Auto Scaling グループを更新して、混合インスタンスポリシーを使用し、容量のリバランスを有効にします。この構造により、スポット容量とオンデマンド容量でグループを指定し、アーキテクチャごとに異なる起動テンプレートを使用できます。

```
aws autoscaling update-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

config.json の内容:

```
{  
  "AutoScalingGroupName": "my-asg",  
  "CapacityRebalance": true,  
  "MixedInstancesPolicy": {  
    "LaunchTemplate": {  
      "LaunchTemplateSpecification": {  
        "LaunchTemplateName": "my-launch-template-for-x86",  
        "Version": "$Latest"  
      },  
      "Overrides": [  
        {  
          "InstanceType": "c6g.large",  
          "LaunchTemplateSpecification": {  
            "LaunchTemplateName": "my-launch-template-for-arm",  
            "Version": "$Latest"  
          }  
        },  
        {  
          "InstanceType": "c5.large"  
        }  
      ]  
    }  
  }  
}
```

```
        },
        {
            "InstanceType": "c5a.large"
        }
    ]
},
"InstancesDistribution": {
    "OnDemandPercentageAboveBaseCapacity": 50,
    "SpotAllocationStrategy": "capacity-optimized"
}
}
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Auto Scaling ユーザーガイド](#)」の「[複数のインスタスタップと購入オプションを持つ Auto Scaling グループ](#)」を参照してください。 EC2 Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAutoScalingGroup](#)」の「」を参照してください。

を使用した Auto Scaling Plans の例 AWS CLI

次のコード例は、Auto Scaling Plans AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-scaling-plan

次の例は、create-scaling-plan を使用する方法を説明しています。

AWS CLI

スケーリングプランを作成するには

次のcreate-scaling-plan例では、既に作成されたJSONファイル (config.json という名前) my-scaling-planを使用して という名前のスケーリングプランを作成します。スケーリングプランの構造には、 という名前の Auto Scaling グループのスケーリング命令が含まれていますmy-asg。TagFilters プロパティをアプリケーションソースとして指定し、予測スケーリングと動的スケーリングを有効にします。

```
aws autoscaling-plans create-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --cli-input-json file://~/config.json
```

config.json ファイルの内容。

```
{  
  "ApplicationSource": {  
    "TagFilters": [  
      {  
        "Key": "purpose",  
        "Values": [  
          "my-application"  
        ]  
      }  
    ]  
  },  
  "ScalingInstructions": [  
    {  
      "ServiceNamespace": "autoscaling",  
      "ResourceId": "autoScalingGroup/my-asg",  
      "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",  
      "ScheduledActionBufferTime": 300,  
      "PredictiveScalingMaxCapacityBehavior":  
"SetForecastCapacityToMaxCapacity",  
      "PredictiveScalingMode": "ForecastAndScale",  
      "PredefinedLoadMetricSpecification": {  
        "PredefinedLoadMetricType": "ASGTotalCPUUtilization"  
      },  
      "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",  
      "MinCapacity": 1,  
      "MaxCapacity": 4,  
    }  
  ]  
}
```

```
    "TargetTrackingConfigurations": [  
      {  
        "PredefinedScalingMetricSpecification": {  
          "PredefinedScalingMetricType": "ASGAverageCPUUtilization"  
        },  
        "TargetValue": 50  
      }  
    ]  
  }  
]
```

出力:

```
{  
  "ScalingPlanVersion": 1  
}
```

詳細については、[AWS Auto Scaling ユーザーガイド](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateScalingPlan](#)」の「」を参照してください。

delete-scaling-plan

次の例は、delete-scaling-plan を使用する方法を説明しています。

AWS CLI

スケーリングプランを削除するには

次のdelete-scaling-plan例では、指定されたスケーリングプランを削除します。

```
aws autoscaling-plans delete-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1
```

このコマンドでは何も出力されません。

詳細については、[AWS Auto Scaling ユーザーガイド](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteScalingPlan](#)」の「」を参照してください。

describe-scaling-plan-resources

次のコード例は、describe-scaling-plan-resources を使用する方法を示しています。

AWS CLI

スケーリングプランのスケーラブルリソースを記述するには

次のdescribe-scaling-plan-resources例では、指定されたスケーリングプランに関連付けられている単一のスケーラブルリソース (Auto Scaling グループ) の詳細を表示します。

```
aws autoscaling-plans describe-scaling-plan-resources \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1
```

出力:

```
{  
  "ScalingPlanResources": [  
    {  
      "ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity",  
      "ScalingPlanVersion": 1,  
      "ResourceId": "autoScalingGroup/my-asg",  
      "ScalingStatusCode": "Active",  
      "ScalingStatusMessage": "Target tracking scaling policies have been  
applied to the resource.",  
      "ScalingPolicies": [  
        {  
          "PolicyName": "AutoScaling-my-asg-b1ab65ae-4be3-4634-bd64-  
c7471662b251",  
          "PolicyType": "TargetTrackingScaling",  
          "TargetTrackingConfiguration": {  
            "PredefinedScalingMetricSpecification": {  
              "PredefinedScalingMetricType":  
"ALBRequestCountPerTarget",  
              "ResourceLabel": "app/my-alb/f37c06a68c1748aa/  
targetgroup/my-target-group/6d4ea56ca2d6a18d"  
            },  
            "TargetValue": 40.0  
          }  
        }  
      ],  
      "ServiceNamespace": "autoscaling",
```

```
        "ScalingPlanName": "my-scaling-plan"
      }
    ]
  }
```

詳細については、[AWS 「Auto Scaling ユーザーガイド」](#)の「Auto Scaling とは」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeScalingPlanResources](#)」の「」を参照してください。

describe-scaling-plans

次のコード例は、describe-scaling-plans を使用する方法を示しています。

AWS CLI

スケーリングプランを記述するには

次のdescribe-scaling-plans例では、指定されたスケーリングプランの詳細を表示します。

```
aws autoscaling-plans describe-scaling-plans \
  --scaling-plan-names scaling-plan-with-asg-and-ddb
```

出力:

```
{
  "ScalingPlans": [
    {
      "LastMutatingRequestTime": 1565388443.963,
      "ScalingPlanVersion": 1,
      "CreationTime": 1565388443.963,
      "ScalingInstructions": [
        {
          "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
          "ScalableDimension":
            "autoscaling:autoScalingGroup:DesiredCapacity",
          "TargetTrackingConfigurations": [
            {
              "PredefinedScalingMetricSpecification": {
                "PredefinedScalingMetricType":
                  "ASGAverageCPUUtilization"
              }
            }
          ],
        }
      ],
    }
  ],
}
```

```
        "TargetValue": 50.0,
        "EstimatedInstanceWarmup": 300,
        "DisableScaleIn": false
    }
],
"ResourceId": "autoScalingGroup/my-asg",
"DisableDynamicScaling": false,
"MinCapacity": 1,
"ServiceNamespace": "autoscaling",
"MaxCapacity": 10
},
{
    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
    "ScalableDimension": "dynamodb:table:ReadCapacityUnits",
    "TargetTrackingConfigurations": [
        {
            "PredefinedScalingMetricSpecification": {
                "PredefinedScalingMetricType":
"DynamoDBReadCapacityUtilization"
            },
            "TargetValue": 50.0,
            "ScaleInCooldown": 60,
            "DisableScaleIn": false,
            "ScaleOutCooldown": 60
        }
    ],
    "ResourceId": "table/my-table",
    "DisableDynamicScaling": false,
    "MinCapacity": 5,
    "ServiceNamespace": "dynamodb",
    "MaxCapacity": 10000
},
{
    "ScalingPolicyUpdateBehavior": "ReplaceExternalPolicies",
    "ScalableDimension": "dynamodb:table:WriteCapacityUnits",
    "TargetTrackingConfigurations": [
        {
            "PredefinedScalingMetricSpecification": {
                "PredefinedScalingMetricType":
"DynamoDBWriteCapacityUtilization"
            },
            "TargetValue": 50.0,
            "ScaleInCooldown": 60,
            "DisableScaleIn": false,
```

```
        "ScaleOutCooldown": 60
      }
    ],
    "ResourceId": "table/my-table",
    "DisableDynamicScaling": false,
    "MinCapacity": 5,
    "ServiceNamespace": "dynamodb",
    "MaxCapacity": 10000
  }
],
"ApplicationSource": {
  "TagFilters": [
    {
      "Values": [
        "my-application-id"
      ],
      "Key": "application"
    }
  ]
},
"StatusStartTime": 1565388455.836,
"ScalingPlanName": "scaling-plan-with-asg-and-ddb",
"StatusMessage": "Scaling plan has been created and applied to all
resources.",
"StatusCode": "Active"
}
]
}
```

詳細については、[AWS 「Auto Scaling ユーザーガイド」の「Auto Scaling とは」](#)を参照してください。AWS Auto Scaling

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeScalingPlans](#)」の「」を参照してください。

get-scaling-plan-resource-forecast-data

次の例は、get-scaling-plan-resource-forecast-data を使用する方法を説明しています。

AWS CLI

ロード予測データを取得するには

この例では、指定されたスケーリングプランに関連付けられているスケーラブルリソース (Auto Scaling グループ) のロード予測データを取得します。

```
aws autoscaling-plans get-scaling-plan-resource-forecast-data \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1 \  
  --service-namespace "autoscaling" \  
  --resource-id autoScalingGroup/my-asg \  
  --scalable-dimension "autoscaling:autoScalingGroup:DesiredCapacity" \  
  --forecast-data-type "LoadForecast" \  
  --start-time "2019-08-30T00:00:00Z" \  
  --end-time "2019-09-06T00:00:00Z"
```

出力:

```
{  
  "Datapoints": [...]  
}
```

詳細については、[AWS Auto Scaling ユーザーガイド](#)「AWS」の「What Is Auto Scaling」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetScalingPlanResourceForecastData](#)」の「」を参照してください。

update-scaling-plan

次の例は、update-scaling-plan を使用する方法を説明しています。

AWS CLI

スケーリングプランを更新するには

次のupdate-scaling-plan例では、指定されたスケーリングプランの Auto Scaling グループのスケーリングメトリクスを変更します。

```
aws autoscaling-plans update-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1 \  
  --scaling-instructions  
  '{"ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity", "ResourceId": "autoScal
```

```
my-asg", "ServiceNamespace": "autoscaling", "TargetTrackingConfigurations":  
[{"PredefinedScalingMetricSpecification":  
{"PredefinedScalingMetricType": "ALBRequestCountPerTarget", "ResourceLabel": "app/my-  
alb/f37c06a68c1748aa/targetgroup/my-target-  
group/6d4ea56ca2d6a18d"}, "TargetValue": 40.0}], "MinCapacity": 1, "MaxCapacity": 10}'
```

このコマンドでは何も出力されません。

詳細については、[AWS 「 Auto Scaling ユーザーガイドAWS 」のAuto Scaling とは](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateScalingPlan](#)」の「」を参照してください。

AWS Backup を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Backup。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-backup-plan

次のコード例は、create-backup-plan を使用する方法を示しています。

AWS CLI

バックアッププランを作成するには

次のcreate-backup-plan例では、35 日間の保持期間を持つ指定されたバックアッププランを作成します。

```
aws backup create-backup-plan \  
--backup-plan "{\"BackupPlanName\": \"Example-Backup-Plan\", \"Rules\": [{\"RuleName\":  
\"DailyBackups\", \"ScheduleExpression\": \"cron(0 5 ? * * *)\", \"StartWindowMinutes  
\": 480, \"TargetBackupVaultName\": \"Default\", \"Lifecycle\": {\"DeleteAfterDays  
\": 35}}]}"
```

出力:

```
{  
  "BackupPlanId": "1fa3895c-a7f5-484a-a371-2dd6a1a9f729",  
  "BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:1fa3895c-  
a7f5-484a-a371-2dd6a1a9f729",  
  "CreationDate": 1568928754.747,  
  "VersionId": "ZjQ2ZTI5YWQtZDg5Yi00MzYzLWJmZTAzMDE1Mzh1MDhjYjEz"  
}
```

詳細については、[「Backup デベロッパーガイド」の「バックアッププランの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBackupPlan](#)」の「」を参照してください。

create-backup-vault

次の例は、create-backup-vault を使用する方法を説明しています。

AWS CLI

バックアップポールトを作成するには

次のcreate-backup-vault例では、指定された名前のバックアップポールトを作成します。

```
aws backup create-backup-vault  
--backup-vault-name sample-vault
```

このコマンドでは何も出力されません。出力:

```
{  
  "BackupVaultName": "sample-vault",  
  "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:sample-  
vault",  
}
```

```
"CreationDate": 1568928338.385
}
```

詳細については、[「Backup デベロッパーガイド」の「Backup Vault の作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBackupVault](#)」の「」を参照してください。

get-backup-plan-from-template

次のコード例は、`get-backup-plan-from-template` を使用する方法を示しています。

AWS CLI

テンプレートから既存のバックアッププランを取得するには

次の`get-backup-plan-from-template`例では、35 日間の保持を持つ日次バックアップを指定するテンプレートから既存のバックアッププランを取得します。

```
aws backup get-backup-plan-from-template \  
--backup-plan-template-id "87c0c1ef-254d-4180-8fef-2e76a2c38aaa"
```

出力:

```
{  
  "BackupPlanDocument": {  
    "Rules": [  
      {  
        "RuleName": "DailyBackups",  
        "ScheduleExpression": "cron(0 5 ? * * *)",  
        "StartWindowMinutes": 480,  
        "Lifecycle": {  
          "DeleteAfterDays": 35  
        }  
      }  
    ]  
  }  
}
```

詳細については、[「Backup デベロッパーガイド」の「バックアッププランの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBackupPlanFromTemplate](#)」の「」を参照してください。

get-backup-plan

次の例は、get-backup-plan を使用する方法を説明しています。

AWS CLI

バックアッププランの詳細を取得するには

次のget-backup-plan例では、指定されたバックアッププランの詳細を表示します。

```
aws backup get-backup-plan \
  --backup-plan-id "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5"
```

出力:

```
{
  "BackupPlan": {
    "BackupPlanName": "Example-Backup-Plan",
    "Rules": [
      {
        "RuleName": "DailyBackups",
        "TargetBackupVaultName": "Default",
        "ScheduleExpression": "cron(0 5 ? * * *)",
        "StartWindowMinutes": 480,
        "CompletionWindowMinutes": 10080,
        "Lifecycle": {
          "DeleteAfterDays": 35
        },
        "RuleId": "70e0ccdc-e9df-4e83-82ad-c1e5a9471cc3"
      }
    ]
  },
  "BackupPlanId": "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5",
  "BackupPlanArn": "arn:aws:backup:us-west-2:123456789012:backup-plan:fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5",
  "VersionId": "NjQ2ZTZkODktMGVhNy00MmQ0LWE4YjktZTkxNTQ3OTkyYTcw",
  "CreationDate": 1568926091.57
}
```

詳細については、[「Backup デベロッパーガイド」の「バックアッププランの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBackupPlan](#)」の「」を参照してください。

list-backup-jobs

次のコード例は、list-backup-jobs を使用する方法を示しています。

AWS CLI

例 1: すべてのバックアップジョブを一覧表示するには

次のlist-backup-jobs例では、AWS アカウントのバックアップジョブに関するメタデータを返します。

```
aws backup list-backup-jobs
```

出力:

```
{
  "BackupJobs": [
    {
      "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "BackupVaultName": "Default",
      "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:Default",
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-12345678901234567",
      "CreationDate": 1600721892.929,
      "State": "CREATED",
      "PercentDone": "0.0",
      "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/AWSBackupDefaultServiceRole",
      "StartBy": 1600725492.929,
      "ResourceType": "EC2"
    },
    {
      "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "BackupVaultName": "Default",
      "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:Default",

```

```

    "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-12345678",
    "CreationDate": 1600721724.77,
    "CompletionDate": 1600721744.488,
    "State": "COMPLETED",
    "PercentDone": "100.0",
    "BackupSizeInBytes": 71,
    "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
    "StartBy": 1600725324.77,
    "ResourceType": "EFS"
  }
]
}

```

詳細については、[「バックアップデベロッパーガイド」の「バックアップの作成AWS」](#)を参照してください。

例 2: 完了したバックアップジョブを一覧表示するには

次のlist-backup-jobs例では、AWS アカウントで完了したバックアップジョブに関するメタデータを返します。

```

aws backup list-backup-jobs \
  --by-state COMPLETED

```

出力:

```

{
  "BackupJobs": [
    {
      "BackupJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "BackupVaultName": "Default",
      "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-
vault:Default",
      "RecoveryPointArn": "arn:aws:backup:us-west-2:123456789012:recovery-
point:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "ResourceArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-12345678",
      "CreationDate": 1600721724.77,
      "CompletionDate": 1600721744.488,

```

```
    "State": "COMPLETED",
    "PercentDone": "100.0",
    "BackupSizeInBytes": 71,
    "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSBackupDefaultServiceRole",
    "StartBy": 1600725324.77,
    "ResourceType": "EFS"
  }
]
}
```

詳細については、[「バックアップデベロッパーガイド」の「バックアップの作成AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBackupJobs](#)」の「」を参照してください。

AWS Batch を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Batch。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

cancel-job

次のコード例は、cancel-job を使用する方法を示しています。

AWS CLI

ジョブをキャンセルするには

この例では、指定されたジョブ ID のジョブをキャンセルします。

コマンド:

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --  
reason "Cancelling job."
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelJob](#)」の「」を参照してください。

create-compute-environment

次の例は、create-compute-environment を使用する方法を説明しています。

AWS CLI

オンデマンドインスタンスを使用してマネージドコンピューティング環境を作成するには

この例では、オンデマンドで起動される特定の C4 インスタンスタイプを持つマネージドコンピューティング環境を作成します。コンピューティング環境は C4OnDemand と呼ばれます。

コマンド:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/  
C4OnDemand.json
```

JSON ファイル形式 :

```
{  
  "computeEnvironmentName": "C4OnDemand",  
  "type": "MANAGED",  
  "state": "ENABLED",  
  "computeResources": {  
    "type": "EC2",  
    "minvCpus": 0,  
    "maxvCpus": 128,  
    "desiredvCpus": 48,  
    "instanceTypes": [  
      "c4.large",  
      "c4.xlarge",
```

```
    "c4.2xlarge",
    "c4.4xlarge",
    "c4.8xlarge"
  ],
  "subnets": [
    "subnet-220c0e0a",
    "subnet-1a95556d",
    "subnet-978f6dce"
  ],
  "securityGroupIds": [
    "sg-cf5093b2"
  ],
  "ec2KeyPair": "id_rsa",
  "instanceRole": "ecsInstanceRole",
  "tags": {
    "Name": "Batch Instance - C4OnDemand"
  }
},
"serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

出力:

```
{
  "computeEnvironmentName": "C4OnDemand",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/C4OnDemand"
}
```

スポットインスタンスを使用してマネージドコンピューティング環境を作成するには

この例では、スポット入札価格がインスタンスタイプのオンデマンド価格の 20% 以下のときに起動される M4 インスタンスタイプを使用してマネージドコンピューティング環境を作成します。コンピューティング環境は M4Spot と呼ばれます。

コマンド:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/M4Spot.json
```

JSON ファイル形式 :

```
{
  "computeEnvironmentName": "M4Spot",
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "type": "SPOT",
    "spotIamFleetRole": "arn:aws:iam::012345678910:role/aws-ec2-spot-fleet-role",
    "minvCpus": 0,
    "maxvCpus": 128,
    "desiredvCpus": 4,
    "instanceTypes": [
      "m4"
    ],
    "bidPercentage": 20,
    "subnets": [
      "subnet-220c0e0a",
      "subnet-1a95556d",
      "subnet-978f6dce"
    ],
    "securityGroupIds": [
      "sg-cf5093b2"
    ],
    "ec2KeyPair": "id_rsa",
    "instanceRole": "ecsInstanceRole",
    "tags": {
      "Name": "Batch Instance - M4Spot"
    }
  },
  "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole"
}
```

出力:

```
{
  "computeEnvironmentName": "M4Spot",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/M4Spot"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateComputeEnvironment](#)」の「」を参照してください。

create-job-queue

次の例は、create-job-queue を使用する方法を説明しています。

AWS CLI

単一のコンピューティング環境で優先度の低いジョブキューを作成するには

この例では、M4Spot コンピューティング環境 LowPriority を使用する というジョブキューを作成します。

コマンド:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/  
LowPriority.json
```

JSON ファイル形式 :

```
{  
  "jobQueueName": "LowPriority",  
  "state": "ENABLED",  
  "priority": 10,  
  "computeEnvironmentOrder": [  
    {  
      "order": 1,  
      "computeEnvironment": "M4Spot"  
    }  
  ]  
}
```

出力:

```
{  
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/LowPriority",  
  "jobQueueName": "LowPriority"  
}
```

2つのコンピューティング環境で優先度の高いジョブキューを作成するには

この例では、1の順序で C4OnDemand コンピューティング環境を使用し、2の順序で M4Spot コンピューティング環境 HighPriority を使用する というジョブキューを作成します。スケジューラは、まず C4OnDemand コンピューティング環境にジョブを配置しようとします。

コマンド:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/HighPriority.json
```

JSON ファイル形式 :

```
{
  "jobQueueName": "HighPriority",
  "state": "ENABLED",
  "priority": 1,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "C4OnDemand"
    },
    {
      "order": 2,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

出力:

```
{
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
  "jobQueueName": "HighPriority"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateJobQueue](#)」の「」を参照してください。

delete-compute-environment

次の例は、delete-compute-environment を使用する方法を説明しています。

AWS CLI

コンピューティング環境を削除するには

この例では、P2OnDemand compute 環境を削除します。

コマンド:

```
aws batch delete-compute-environment --compute-environment P2OnDemand
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteComputeEnvironment](#)」の「」を参照してください。

delete-job-queue

次の例は、delete-job-queue を使用する方法を説明しています。

AWS CLI

ジョブキューを削除するには

この例では、GPGPUジョブキューを削除します。

コマンド:

```
aws batch delete-job-queue --job-queue GPGPU
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteJobQueue](#)」の「」を参照してください。

deregister-job-definition

次の例は、deregister-job-definition を使用する方法を説明しています。

AWS CLI

ジョブ定義の登録を解除するには

この例では、sleep10 というジョブ定義を登録解除します。

コマンド:

```
aws batch deregister-job-definition --job-definition sleep10
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterJobDefinition](#)」の「」を参照してください。

describe-compute-environments

次のコード例は、describe-compute-environments を使用する方法を示しています。

AWS CLI

コンピューティング環境を記述するには

この例では、P2OnDemand compute 環境について説明します。

コマンド:

```
aws batch describe-compute-environments --compute-environments P2OnDemand
```

出力:

```
{
  "computeEnvironments": [
    {
      "status": "VALID",
      "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole",
      "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/P2OnDemand",
      "computeResources": {
        "subnets": [
          "subnet-220c0e0a",
          "subnet-1a95556d",
          "subnet-978f6dce"
        ],
        "tags": {
          "Name": "Batch Instance - P2OnDemand"
        },
        "desiredvCpus": 48,
        "minvCpus": 0,
        "instanceTypes": [
          "p2"
        ],
        "securityGroupIds": [
          "sg-cf5093b2"
        ],
        "instanceRole": "ecsInstanceRole",
        "maxvCpus": 128,
        "type": "EC2",

```

```
        "ec2KeyPair": "id_rsa"
      },
      "statusReason": "ComputeEnvironment Healthy",
      "ecsClusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/
P20nDemand_Batch_2c06f29d-d1fe-3a49-879d-42394c86effc",
      "state": "ENABLED",
      "computeEnvironmentName": "P20nDemand",
      "type": "MANAGED"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeComputeEnvironments](#)」の「」を参照してください。

describe-job-definitions

次のコード例は、describe-job-definitions を使用する方法を示しています。

AWS CLI

アクティブなジョブ定義を記述するには

この例では、アクティブなすべてのジョブ定義について説明します。

コマンド:

```
aws batch describe-job-definitions --status ACTIVE
```

出力:

```
{
  "jobDefinitions": [
    {
      "status": "ACTIVE",
      "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-
definition/sleep60:1",
      "containerProperties": {
        "mountPoints": [],
        "parameters": {},
        "image": "busybox",
        "environment": {},

```

```
        "vcpus": 1,
        "command": [
            "sleep",
            "60"
        ],
        "volumes": [],
        "memory": 128,
        "ulimits": []
    },
    "type": "container",
    "jobDefinitionName": "sleep60",
    "revision": 1
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeJobDefinitions](#)」の「」を参照してください。

describe-job-queues

次のコード例は、describe-job-queues を使用する方法を示しています。

AWS CLI

ジョブキューを記述するには

この例では、HighPriority ジョブキューについて説明します。

コマンド:

```
aws batch describe-job-queues --job-queues HighPriority
```

出力:

```
{
  "jobQueues": [
    {
      "status": "VALID",
      "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
      "computeEnvironmentOrder": [
```

```
        {
            "computeEnvironment": "arn:aws:batch:us-
east-1:012345678910:compute-environment/C4OnDemand",
            "order": 1
        }
    ],
    "statusReason": "JobQueue Healthy",
    "priority": 1,
    "state": "ENABLED",
    "jobQueueName": "HighPriority"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeJobQueues](#)」の「」を参照してください。

describe-jobs

次のコード例は、describe-jobs を使用する方法を示しています。

AWS CLI

ジョブを記述するには

次のdescribe-jobs例では、指定されたジョブ ID を持つジョブについて説明します。

```
aws batch describe-jobs \  
  --jobs bcf0b186-a532-4122-842e-2ccab8d54efb
```

出力:

```
{
  "jobs": [
    {
      "status": "SUBMITTED",
      "container": {
        "mountPoints": [],
        "image": "busybox",
        "environment": [],
        "vcpus": 1,
        "command": [
```

```
        "sleep",
        "60"
    ],
    "volumes": [],
    "memory": 128,
    "ulimits": []
  },
  "parameters": {},
  "jobDefinition": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep60:1",
  "jobQueue": "arn:aws:batch:us-east-1:012345678910:job-queue/
HighPriority",
  "jobId": "bcf0b186-a532-4122-842e-2ccab8d54efb",
  "dependsOn": [],
  "jobName": "example",
  "createdAt": 1480483387803
}
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeJobs](#)」の「」を参照してください。

list-jobs

次の例は、list-jobs を使用する方法を説明しています。

AWS CLI

実行中のジョブを一覧表示するには

この例では、ジョブキューで実行中の HighPriority ジョブを一覧表示します。

コマンド:

```
aws batch list-jobs --job-queue HighPriority
```

出力:

```
{
  "jobSummaryList": [
    {
```

```
        "jobName": "example",
        "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
    }
]
}
```

送信されたジョブを一覧表示するには

この例では、ジョブステータスの HighPriority ジョブキュー内の SUBMITTED ジョブを一覧表示します。

コマンド:

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

出力:

```
{
  "jobSummaryList": [
    {
      "jobName": "example",
      "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListJobs](#)」の「」を参照してください。

register-job-definition

次のコード例は、register-job-definition を使用する方法を示しています。

AWS CLI

ジョブ定義を登録するには

この例では、単純なコンテナジョブのジョブ定義を登録します。

コマンド:


```
aws batch register-job-definition --job-definition-name sleep30 --type container --  
container-properties '{ "image": "busybox", "vcpus": 1, "memory": 128, "command":  
[ "sleep", "30"]}'
```

出力:

```
{  
  "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-definition/  
sleep30:1",  
  "jobDefinitionName": "sleep30",  
  "revision": 1  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterJobDefinition](#)」の「」を参照してください。

submit-job

次のコード例は、submit-job を使用する方法を示しています。

AWS CLI

ジョブを送信するには

この例では、サンプルという単純なコンテナジョブを HighPriority ジョブキューに送信します。

コマンド:

```
aws batch submit-job --job-name example --job-queue HighPriority --job-  
definition sleep60
```

出力:

```
{  
  "jobName": "example",  
  "jobId": "876da822-4198-45f2-a252-6cea32512ea8"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SubmitJob](#)」の「」を参照してください。

terminate-job

次のコード例は、`terminate-job` を使用する方法を示しています。

AWS CLI

ジョブを終了するには

この例では、指定されたジョブ ID でジョブを終了します。

コマンド:

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --  
reason "Terminating job."
```

- API 詳細については、AWS CLI 「コマンドリファレンス[TerminateJob](#)」の「」を参照してください。

update-compute-environment

次のコード例は、`update-compute-environment` を使用する方法を示しています。

AWS CLI

コンピューティング環境を更新するには

この例では、削除できるように P2OnDemand compute 環境を無効にします。

コマンド:

```
aws batch update-compute-environment --compute-environment P2OnDemand --  
state DISABLED
```

出力:

```
{  
  "computeEnvironmentName": "P2OnDemand",  
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-  
environment/P2OnDemand"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateComputeEnvironment](#)」の「」を参照してください。

update-job-queue

次のコード例は、update-job-queue を使用する方法を示しています。

AWS CLI

ジョブキューを更新するには

この例では、削除できるようにジョブキューを無効にします。

コマンド:

```
aws batch update-job-queue --job-queue GPGPU --state DISABLED
```

出力:

```
{
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/GPGPU",
  "jobQueueName": "GPGPU"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateJobQueue](#)」の「」を参照してください。

AWS Budgets を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Budgets。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-budget

次のコード例は、create-budget を使用する方法を示しています。

AWS CLI

コストと使用状況の予算を作成するには

次のcreate-budgetコマンドは、コストと使用量の予算を作成します。

```
aws budgets create-budget \  
  --account-id 111122223333 \  
  --budget file://budget.json \  
  --notifications-with-subscribers file://notifications-with-subscribers.json
```

budget.json の内容:

```
{  
  "BudgetLimit": {  
    "Amount": "100",  
    "Unit": "USD"  
  },  
  "BudgetName": "Example Tag Budget",  
  "BudgetType": "COST",  
  "CostFilters": {  
    "TagKeyValue": [  
      "user:Key$value1",  
      "user:Key$value2"  
    ]  
  },  
  "CostTypes": {  
    "IncludeCredit": true,  
    "IncludeDiscount": true,  
    "IncludeOtherSubscription": true,  
    "IncludeRecurring": true,  
    "IncludeRefund": true,  
    "IncludeSubscription": true,  
    "IncludeSupport": true,  
    "IncludeTax": true,  
  }  
}
```

```
    "IncludeUpfront": true,  
    "UseBlended": false  
  },  
  "TimePeriod": {  
    "Start": 1477958399,  
    "End": 3706473600  
  },  
  "TimeUnit": "MONTHLY"  
}
```

notifications-with-subscribers.json の内容:

```
[  
  {  
    "Notification": {  
      "ComparisonOperator": "GREATER_THAN",  
      "NotificationType": "ACTUAL",  
      "Threshold": 80,  
      "ThresholdType": "PERCENTAGE"  
    },  
    "Subscribers": [  
      {  
        "Address": "example@example.com",  
        "SubscriptionType": "EMAIL"  
      }  
    ]  
  }  
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBudget](#)」の「」を参照してください。

create-notification

次のコード例は、create-notification を使用する方法を示しています。

AWS CLI

指定されたコストと使用量の予算の通知を作成するには

この例では、指定されたコストと使用量の予算の通知を作成します。

コマンド:

```
aws budgets create-notification --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNotification](#)」の「」を参照してください。

create-subscriber

次のコード例は、create-subscriber を使用する方法を示しています。

AWS CLI

コストと使用量の予算に関連付けられた通知のサブスクライバーを作成するにはこの例では、指定された通知のサブスクライバーを作成します。

コマンド:

```
aws budgets create-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubscriber](#)」の「」を参照してください。

delete-budget

次の例は、delete-budget を使用する方法を説明しています。

AWS CLI

コストと使用状況の予算を削除するにはこの例では、指定されたコストと使用量の予算を削除します。

コマンド:

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBudget](#)」の「」を参照してください。

delete-notification

次のコード例は、delete-notification を使用する方法を示しています。

AWS CLI

予算から通知を削除するには

この例では、指定された予算から指定された通知を削除します。

コマンド:

```
aws budgets delete-notification --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNotification](#)」の「」を参照してください。

delete-subscriber

次のコード例は、delete-subscriber を使用する方法を示しています。

AWS CLI

通知からサブスクライバーを削除するには

この例では、指定された通知から指定されたサブスクライバーを削除します。

コマンド:

```
aws budgets delete-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSubscriber](#)」の「」を参照してください。

describe-budget

次の例は、describe-budget を使用する方法を説明しています。

AWS CLI

アカウントに関連付けられた予算を取得するには

この例では、指定されたコストと使用量の予算を取得します。

コマンド:

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

出力:

```
{
  "Budget": {
    "CalculatedSpend": {
      "ForecastedSpend": {
        "Amount": "2641.548000000000022919266484677791595458984375",
        "Unit": "USD"
      },
      "ActualSpend": {
        "Amount": "604.456000000000000172803993336856365203857421875",
        "Unit": "USD"
      }
    },
    "BudgetType": "COST",
    "BudgetLimit": {
      "Amount": "100",
      "Unit": "USD"
    },
    "BudgetName": "Example Budget",
    "CostTypes": {
      "IncludeOtherSubscription": true,
      "IncludeUpfront": true,
      "IncludeRefund": true,
      "UseBlended": false,
      "IncludeDiscount": true,
      "UseAmortized": false,
      "IncludeTax": true,
      "IncludeCredit": true,
    }
  }
}
```



```
    "IncludeSupport": true,
    "IncludeRecurring": true,
    "IncludeSubscription": true
  },
  "TimeUnit": "MONTHLY",
  "TimePeriod": {
    "Start": 1477958399.0,
    "End": 3706473600.0
  },
  "CostFilters": {
    "AZ": [
      "us-east-1"
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeBudget](#)」の「」を参照してください。

describe-budgets

次の例は、describe-budgets を使用する方法を説明しています。

AWS CLI

アカウントに関連付けられた予算を取得するには

この例では、アカウントのコストと使用状況の予算を取得します。

コマンド:

```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

出力:

```
{
  "Budgets": [
    {
      "CalculatedSpend": {
        "ForecastedSpend": {
```

```
        "Amount": "2641.548000000000022919266484677791595458984375",
        "Unit": "USD"
    },
    "ActualSpend": {
        "Amount": "604.45600000000000172803993336856365203857421875",
        "Unit": "USD"
    }
},
"BudgetType": "COST",
"BudgetLimit": {
    "Amount": "100",
    "Unit": "USD"
},
"BudgetName": "Example Budget",
"CostTypes": {
    "IncludeOtherSubscription": true,
    "IncludeUpfront": true,
    "IncludeRefund": true,
    "UseBlended": false,
    "IncludeDiscount": true,
    "UseAmortized": false,
    "IncludeTax": true,
    "IncludeCredit": true,
    "IncludeSupport": true,
    "IncludeRecurring": true,
    "IncludeSubscription": true
},
"TimeUnit": "MONTHLY",
"TimePeriod": {
    "Start": 1477958399.0,
    "End": 3706473600.0
},
"CostFilters": {
    "AZ": [
        "us-east-1"
    ]
}
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeBudgets](#)」の「」を参照してください。

describe-notifications-for-budget

次のコード例は、describe-notifications-for-budget を使用する方法を示しています。

AWS CLI

予算の通知を取得するには

この例では、コストと使用状況の予算の通知を取得します。

コマンド:

```
aws budgets describe-notifications-for-budget --account-id 111122223333 --budget-name "Example Budget" --max-results 5
```

出力:

```
{
  "Notifications": [
    {
      "Threshold": 80.0,
      "ComparisonOperator": "GREATER_THAN",
      "NotificationType": "ACTUAL"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNotificationsForBudget](#)」の「」を参照してください。

describe-subscribers-for-notification

次のコード例は、describe-subscribers-for-notification を使用する方法を示しています。

AWS CLI

予算通知のサブスクライバーを取得するには

この例では、コストと使用状況の予算通知のサブスクライバーを取得します。

コマンド:

```
aws budgets describe-subscribers-for-notification --  
account-id 111122223333 --budget-name "Example Budget" --  
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE  
--max-results 5
```

出力:

```
{  
  "Subscribers": [  
    {  
      "SubscriptionType": "EMAIL",  
      "Address": "example2@example.com"  
    },  
    {  
      "SubscriptionType": "EMAIL",  
      "Address": "example@example.com"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeSubscribersForNotification](#)」の「」を参照してください。

update-budget

次の例は、update-budget を使用する方法を説明しています。

AWS CLI

コストと使用状況の予算の予算を置き換えるには

この例では、コストと使用量の予算を新しい予算に置き換えます。

コマンド:

```
aws budgets update-budget --account-id 111122223333 --new-budget file://new-budget.json
```

new-budget.json:

```
{
```

```
"BudgetLimit": {
  "Amount": "100",
  "Unit": "USD"
},
"BudgetName": "Example Budget",
"BudgetType": "COST",
"CostFilters": {
  "AZ" : [ "us-east-1" ]
},
"CostTypes": {
  "IncludeCredit": false,
  "IncludeDiscount": true,
  "IncludeOtherSubscription": true,
  "IncludeRecurring": true,
  "IncludeRefund": true,
  "IncludeSubscription": true,
  "IncludeSupport": true,
  "IncludeTax": true,
  "IncludeUpfront": true,
  "UseBlended": false,
  "UseAmortized": true
},
"TimePeriod": {
  "Start": 1477958399,
  "End": 3706473600
},
"TimeUnit": "MONTHLY"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateBudget](#)」の「」を参照してください。

update-notification

次の例は、update-notification を使用する方法を説明しています。

AWS CLI

コストと使用量の予算の通知を置き換えるには

この例では、コストと使用状況の予算の 80% の通知を 90% の通知に置き換えます。

コマンド:

```
aws budgets update-notification --account-id 111122223333 --budget-name "Example Budget" --old-notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT --new-notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=90,ThresholdType=PERCENT
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateNotification](#)」の「」を参照してください。

update-subscriber

次の例は、update-subscriber を使用する方法を説明しています。

AWS CLI

コストと使用量の予算のサブスクライバーを置き換えるには

この例では、サブスクライバーをコストと使用量の予算に置き換えます。

コマンド:

```
aws budgets update-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT --old-subscriber SubscriptionType=EMAIL,Address=example@example.com --new-subscriber SubscriptionType=EMAIL,Address=example2@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSubscriber](#)」の「」を参照してください。

を使用した Amazon Chime の例 AWS CLI

次のコード例は、Amazon Chime AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-phone-number-with-user

次のコード例は、`associate-phone-number-with-user` を使用する方法を示しています。

AWS CLI

電話番号をユーザーに関連付けるには

次の`associate-phone-number-with-user`例では、指定された電話番号をユーザーに関連付けます。

```
aws chime associate-phone-number-with-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \  
  --e164-phone-number " +12065550100"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Chime 管理ガイド」の「[ユーザー電話番号の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociatePhoneNumberWithUser](#)」の「」を参照してください。

associate-phone-numbers-with-voice-connector-group

次のコード例は、`associate-phone-numbers-with-voice-connector-group` を使用する方法を示しています。

AWS CLI

電話番号を Amazon Chime Voice Connector グループに関連付けるには

次の`associate-phone-numbers-with-voice-connector-group`例では、指定された電話番号を Amazon Chime Voice Connector グループと関連付けます。

```
aws chime associate-phone-numbers-with-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jkl8901 \  
  --e164-phone-numbers "+12065550100" "+12065550101" \  
  --force-associate
```

出力:

```
{  
  "PhoneNumberErrors": []  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector グループの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociatePhoneNumbersWithVoiceConnectorGroup](#)」の「」を参照してください。

associate-phone-numbers-with-voice-connector

次のコード例は、`associate-phone-numbers-with-voice-connector` を使用する方法を示しています。

AWS CLI

電話番号を Amazon Chime Voice Connector に関連付けるには

次の `associate-phone-numbers-with-voice-connector` 例では、指定された電話番号を Amazon Chime Voice Connector に関連付けます。

```
aws chime associate-phone-numbers-with-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --e164-phone-numbers "+12065550100" "+12065550101" \  
  --force-associate
```

出力:

```
{  
  "PhoneNumberErrors": []  
}
```


詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociatePhoneNumbersWithVoiceConnector](#)」の「」を参照してください。

associate-signin-delegate-groups-with-account

次のコード例は、`associate-signin-delegate-groups-with-account` を使用する方法を示しています。

AWS CLI

サインイン代理グループを関連付けるには

次の`associate-signin-delegate-groups-with-account`例では、指定されたサインイン委任グループを指定された Amazon Chime アカウントと関連付けます。

```
aws chime associate-signin-delegate-groups-with-account \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --signin-delegate-groups GroupName=my_users
```

このコマンドでは何も出力されません。

詳細については、「Amazon Chime 管理ガイド」の[「ユーザーアクセスとアクセス許可の管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateSigninDelegateGroupsWithAccount](#)」の「」を参照してください。

batch-create-room-membership

次のコード例は、`batch-create-room-membership` を使用する方法を示しています。

AWS CLI

複数のルームメンバーシップを作成するには

次の`batch-create-room-membership`例では、チャットルームのメンバーとして複数のユーザーをチャットルームに追加します。また、管理者ロールとメンバーロールをユーザーに割り当てます。

```
aws chime batch-create-room-membership \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --membership-item-list "MemberId=1ab2345c-67de-8901-  
f23g-45h678901j2k,Role=Administrator" "MemberId=2ab2345c-67de-8901-  
f23g-45h678901j2k,Role=Member"
```

出力:

```
{  
  "ResponseMetadata": {  
    "RequestId": "169ba401-d886-475f-8b3f-e01eac6fadfb",  
    "HTTPStatusCode": 201,  
    "HTTPHeaders": {  
      "x-amzn-requestid": "169ba401-d886-475f-8b3f-e01eac6fadfb",  
      "content-type": "application/json",  
      "content-length": "13",  
      "date": "Mon, 02 Dec 2019 22:46:58 GMT",  
      "connection": "keep-alive"  
    },  
    "RetryAttempts": 0  
  },  
  "Errors": []  
}
```

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchCreateRoomMembership](#)」の「」を参照してください。

batch-delete-phone-number

次の例は、batch-delete-phone-number を使用する方法を説明しています。

AWS CLI

複数の電話番号を削除するには

次のbatch-delete-phone-number例では、指定された電話番号をすべて削除します。

```
aws chime batch-delete-phone-number \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

```
--phone-number-ids "%2B12065550100" "%2B12065550101"
```

このコマンドでは何も出力されません。出力:

```
{
  "PhoneNumberErrors": []
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスBatchDeletePhoneNumber](#)」の「」を参照してください。

batch-suspend-user

次の例は、batch-suspend-user を使用する方法を説明しています。

AWS CLI

複数のユーザーを停止するには

次のbatch-suspend-user例では、指定された Amazon Chime アカウントからリストされたユーザーを一時停止します。

```
aws chime batch-suspend-user \
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

出力:

```
{
  "UserErrors": []
}
```

- API 詳細については、AWS CLI 「[コマンドリファレンスBatchSuspendUser](#)」の「」を参照してください。

batch-unsuspend-user

次の例は、batch-unsuspend-user を使用する方法を説明しています。

AWS CLI

複数のユーザーの一時停止を解除するには

次のbatch-unsuspend-user例では、指定された Amazon Chime アカウントのリストされたユーザーの以前の停止をすべて削除します。

```
aws chime batch-unsuspend-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id-list "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" "a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE" "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
```

出力:

```
{  
  "UserErrors": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchUnsuspendUser](#)」の「」を参照してください。

batch-update-phone-number

次の例は、batch-update-phone-number を使用する方法を説明しています。

AWS CLI

複数の電話番号製品タイプを同時に更新するには

次のbatch-update-phone-number例では、指定されたすべての電話番号の製品タイプを更新します。

```
aws chime batch-update-phone-number \  
  --update-phone-number-request-items PhoneNumberId=  
%2B12065550100,ProductType=BusinessCalling PhoneNumberId=  
%2B12065550101,ProductType=BusinessCalling
```

出力:

```
{  
  "PhoneNumberErrors": []  
}
```

```
}
```

複数の電話番号呼び出し名を同時に更新するには

次のbatch-update-phone-number例では、指定されたすべての電話番号の呼び出し名を更新します。

```
aws chime batch-update-phone-number \  
  --update-phone-number-request-items PhoneNumberId=  
%2B14013143874,CallingName=phonenumber1 PhoneNumberId=  
%2B14013144061,CallingName=phonenumber2
```

出力:

```
{  
  "PhoneNumberErrors": []  
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchUpdatePhoneNumber](#)」の「」を参照してください。

batch-update-user

次の例は、batch-update-user を使用する方法を説明しています。

AWS CLI

1つのコマンドで複数のユーザーを更新するには

次のbatch-update-user例ではLicenseType、指定された Amazon Chime アカウント内のリストされた各ユーザーのを更新します。

```
aws chime batch-update-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE  
  --update-user-request-items "UserId=a1b2c3d4-5678-90ab-  
cdef-22222EXAMPLE,LicenseType=Basic" "UserId=a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE,LicenseType=Basic"
```

出力:

```
{
  "UserErrors": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchUpdateUser](#)」の「」を参照してください。

create-account

次のコード例は、create-account を使用する方法を示しています。

AWS CLI

アカウントを作成するには

次のcreate-account例では、管理者のアカウントの下に Amazon Chime AWS アカウントを作成します。

```
aws chime create-account \
  --name MyChimeAccount
```

出力:

```
{
  "Account": {
    "AwsAccountId": "111122223333",
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "Name": "MyChimeAccount",
    "AccountType": "Team",
    "CreatedTimestamp": "2019-01-04T17:11:22.003Z",
    "DefaultLicense": "Pro",
    "SupportedLicenses": [
      "Basic",
      "Pro"
    ],
    "SigninDelegateGroups": [
      {
        "GroupName": "myGroup"
      }
    ]
  }
}
```

```
}
```

詳細については、「Amazon Chime 管理ガイド」の「[入門](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAccount](#)」の「」を参照してください。

create-bot

次のコード例は、create-bot を使用する方法を示しています。

AWS CLI

Amazon Chime ボットを作成するには

次のcreate-bot例では、指定された Amazon Chime Enterprise アカウントのボットを作成します。

```
aws chime create-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --display-name "myBot" \  
  --domain "example.com"
```

出力:

```
{  
  "Bot": {  
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "DisplayName": "myBot (Bot)",  
    "BotType": "ChatBot",  
    "Disabled": false,  
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "BotEmail": "myBot-chimebot@example.com",  
    "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
  }  
}
```

詳細については、「[Amazon Chime デベロッパーガイド](#)」の「[チャットボットと Amazon Chime の統合](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateBot](#)」の「」を参照してください。

create-phone-number-order

次の例は、create-phone-number-order を使用する方法を説明しています。

AWS CLI

電話番号の順序を作成するには

次のcreate-phone-number-order例では、指定された電話番号の電話番号順序を作成します。

```
aws chime create-phone-number-order \  
  --product-type VoiceConnector \  
  --e164-phone-numbers " +12065550100" "+12065550101" "+12065550102"
```

出力:

```
{  
  "PhoneNumberOrder": {  
    "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",  
    "ProductType": "VoiceConnector",  
    "Status": "Processing",  
    "OrderedPhoneNumbers": [  
      {  
        "E164PhoneNumber": "+12065550100",  
        "Status": "Processing"  
      },  
      {  
        "E164PhoneNumber": "+12065550101",  
        "Status": "Processing"  
      },  
      {  
        "E164PhoneNumber": "+12065550102",  
        "Status": "Processing"  
      }  
    ],  
    "CreatedTimestamp": "2019-08-09T21:35:21.427Z",  
    "UpdatedTimestamp": "2019-08-09T21:35:22.408Z"  
  }  
}
```



```
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePhoneNumberOrder](#)」の「」を参照してください。

create-proxy-session

次のコード例は、create-proxy-session を使用する方法を示しています。

AWS CLI

プロキシセッションを作成するには

次のcreate-proxy-session例では、音声とSMS機能を使用してプロキシセッションを作成します。

```
aws chime create-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --participant-phone-numbers " +14015550101" "+12065550100" \  
  --capabilities "Voice" "SMS"
```

出力:

```
{  
  "ProxySession": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",  
    "Status": "Open",  
    "ExpiryMinutes": 60,  
    "Capabilities": [  
      "SMS",  
      "Voice"  
    ],  
    "CreatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "UpdatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "Participants": [  
      {  
        "PhoneNumber": "+12065550100",  
        "ProxyPhoneNumber": "+19135550199"  
      },  
    ],  
  },  
}
```

```
{
  "PhoneNumber": "+14015550101",
  "ProxyPhoneNumber": "+19135550199"
}
]
```

詳細については、「[Amazon Chime デベロッパーガイド](#)」の「[プロキシフォンセッション](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス CreateProxySession](#)」の「」を参照してください。

create-room-membership

次の例は、create-room-membership を使用する方法を説明しています。

AWS CLI

ルームメンバーシップを作成するには

次のcreate-room-membership例では、指定されたユーザーをチャットルームメンバーとしてチャットルームに追加します。

```
aws chime create-room-membership \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

出力:

```
{
  "RoomMembership": {
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
    "Member": {
      "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
      "MemberType": "User",
      "Email": "janed@example.com",
      "FullName": "Jane Doe",
      "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
    }
  },
```

```
    "Role": "Member",
    "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
    "UpdatedTimestamp": "2019-12-02T22:36:41.969Z"
  }
}
```

詳細については、「Amazon Chime ユーザーガイド」の[「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRoomMembership](#)」の「」を参照してください。

create-room

次のコード例は、create-room を使用する方法を示しています。

AWS CLI

チャットルームを作成するには

次のcreate-room例では、指定された Amazon Chime アカウントのチャットルームを作成します。

```
aws chime create-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --name chatRoom
```

出力:

```
{
  "Room": {
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
    "Name": "chatRoom",
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",
    "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"
  }
}
```

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRoom](#)」の「」を参照してください。

create-user

次のコード例は、create-user を使用する方法を示しています。

AWS CLI

共有デバイスのユーザープロフィールを作成するには

次のcreate-user例では、指定された E メールアドレスの共有デバイスプロフィールを作成します。

```
aws chime create-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --email roomdevice@example.com \  
  --user-type SharedDevice
```

出力:

```
{  
  "User": {  
    "UserId": "1ab2345c-67de-8901-f23g-45h678901j2k",  
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",  
    "PrimaryEmail": "roomdevice@example.com",  
    "DisplayName": "Room Device",  
    "LicenseType": "Pro",  
    "UserType": "SharedDevice",  
    "UserRegistrationStatus": "Registered",  
    "RegisteredOn": "2020-01-15T22:38:09.806Z",  
    "AlexaForBusinessMetadata": {  
      "IsAlexaForBusinessEnabled": false  
    }  
  }  
}
```

詳細については、「Amazon Chime 管理ガイド」の「[セットアップの準備](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUser](#)」の「」を参照してください。

create-voice-connector-group

次のコード例は、`create-voice-connector-group` を使用する方法を示しています。

AWS CLI

Amazon Chime Voice Connector グループを作成するには

次の`create-voice-connector-group`例では、指定された Amazon Chime Voice Connector を含む Amazon Chime Voice Connector グループを作成します。

```
aws chime create-voice-connector-group \  
  --name myGroup \  
  --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=2
```

出力:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl18901",  
    "Name": "myGroup",  
    "VoiceConnectorItems": [],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector Groups の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVoiceConnectorGroup](#)」の「」を参照してください。

create-voice-connector

次のコード例は、`create-voice-connector` を使用する方法を示しています。

AWS CLI

Amazon Chime Voice Connector を作成するには

次の`create-voice-connector`例では、暗号化を有効にして、指定された AWS リージョンに Amazon Chime Voice Connector を作成します。

```
aws chime create-voice-connector \  
  --name newVoiceConnector \  
  --aws-region us-west-2 \  
  --require-encryption
```

出力:

```
{  
  "VoiceConnector": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "AwsRegion": "us-west-2",  
    "Name": "newVoiceConnector",  
    "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",  
    "RequireEncryption": true,  
    "CreatedTimestamp": "2019-09-18T20:34:01.352Z",  
    "UpdatedTimestamp": "2019-09-18T20:34:01.352Z"  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVoiceConnector](#)」の「」を参照してください。

delete-account

次のコード例は、delete-account を使用する方法を示しています。

AWS CLI

アカウントを削除するには

次のdelete-account例では、指定されたアカウントを削除します。

```
aws chime delete-account --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「Amazon Chime 管理ガイド」の[「アカウントの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccount](#)」の「」を参照してください。

delete-phone-number

次の例は、delete-phone-number を使用する方法を説明しています。

AWS CLI

電話番号を削除するには

次のdelete-phone-number例では、指定された電話番号を削除キューに移動します。

```
aws chime delete-phone-number \  
  --phone-number-id "+12065550100"
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePhoneNumber](#)」の「」を参照してください。

delete-proxy-session

次のコード例は、delete-proxy-session を使用する方法を示しています。

AWS CLI

プロキシセッションを削除するには

次のdelete-proxy-session例では、指定されたプロキシセッションを削除します。

```
aws chime delete-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk56789l
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Chime デベロッパーガイド](#)」の「[プロキシフォンセッション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProxySession](#)」の「」を参照してください。

delete-room-membership

次の例は、delete-room-membership を使用する方法を説明しています。

AWS CLI

チャットルームのメンバーとしてユーザーを削除するには

次のdelete-room-membership例では、指定されたチャットルームから指定されたメンバーを削除します。

```
aws chime delete-room-membership \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRoomMembership](#)」の「」を参照してください。

delete-room

次の例は、delete-room を使用する方法を説明しています。

AWS CLI

チャットルームを削除するには

次のdelete-room例では、指定されたチャットルームを削除し、チャットルームのメンバーシップを削除します。

```
aws chime delete-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```



```
--room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRoom](#)」の「」を参照してください。

delete-voice-connector-group

次のコード例は、delete-voice-connector-group を使用する方法を示しています。

AWS CLI

title

次のdelete-voice-connector-group例では、指定された Amazon Chime Voice Connector グループを削除します。

```
aws chime delete-voice-connector-group \  
--voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector Groups の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVoiceConnectorGroup](#)」の「」を参照してください。

delete-voice-connector-origination

次のコード例は、delete-voice-connector-origination を使用する方法を示しています。

AWS CLI

送信元設定を削除するには

次のdelete-voice-connector-origination例では、指定された Amazon Chime Voice Connector から発信元ホスト、ポート、プロトコル、優先度、および重みを削除します。

```
aws chime delete-voice-connector-origination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Chime 管理ガイド](#)」の「[Amazon Chime Voice Connectors の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVoiceConnectorOrigination](#)」の「」を参照してください。

delete-voice-connector-proxy

次のコード例は、delete-voice-connector-proxy を使用する方法を示しています。

AWS CLI

プロキシ設定を削除するには

次のdelete-voice-connector-proxy例では、Amazon Chime Voice Connector からプロキシ設定を削除します。

```
aws chime delete-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Chime デベロッパーガイド](#)」の「[プロキシフォンセッション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVoiceConnectorProxy](#)」の「」を参照してください。

delete-voice-connector-streaming-configuration

次の例は、delete-voice-connector-streaming-configuration を使用する方法を説明しています。

AWS CLI

ストリーミング設定を削除するには

次のdelete-voice-connector-streaming-configuration例では、指定された Amazon Chime Voice Connector のストリーミング設定を削除します。

```
aws chime delete-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector データを Kinesis にストリーミングする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteVoiceConnectorStreamingConfiguration](#)」の「」を参照してください。

delete-voice-connector-termination-credentials

次の例は、delete-voice-connector-termination-credentials を使用する方法を説明しています。

AWS CLI

終了認証情報を削除するには

次のdelete-voice-connector-termination-credentials例では、指定されたユーザー名と Amazon Chime Voice Connector の終了認証情報を削除します。

```
aws chime delete-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --usernames "jdoe"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteVoiceConnectorTerminationCredentials](#)」の「」を参照してください。

delete-voice-connector-termination

次のコード例は、delete-voice-connector-termination を使用する方法を示しています。

AWS CLI

終了設定を削除するには

次のdelete-voice-connector-termination例では、指定された Amazon Chime Voice Connector の終了設定を削除します。

```
aws chime delete-voice-connector-termination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVoiceConnectorTermination](#)」の「」を参照してください。

delete-voice-connector

次の例は、delete-voice-connector を使用する方法を説明しています。

AWS CLI

Amazon Chime Voice Connector を削除するには

次のdelete-voice-connector例では、これを行います。

```
aws chime delete-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVoiceConnector](#)」の「」を参照してください。

disassociate-phone-number-from-user

次の例は、disassociate-phone-number-from-user を使用する方法を説明しています。

AWS CLI

ユーザーから電話番号の関連付けを解除するには

次のdisassociate-phone-number-from-user例では、指定されたユーザーから電話番号の関連付けを解除します。

```
aws chime disassociate-phone-number-from-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

このコマンドでは何も出力されません。

詳細については、「Amazon Chime 管理ガイド」の[「ユーザー電話番号の管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンスDisassociatePhoneNumberFromUser」の「」を参照してください。

disassociate-phone-numbers-from-voice-connector-group

次の例は、disassociate-phone-numbers-from-voice-connector-group を使用する方法を説明しています。

AWS CLI

Amazon Chime Voice Connector グループから電話番号の関連付けを解除するには

次のdisassociate-phone-numbers-from-voice-connector-group例では、Amazon Chime Voice Connector グループから指定された電話番号の関連付けを解除します。

```
aws chime disassociate-phone-numbers-from-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --e164-phone-numbers "+12065550100" "+12065550101"
```

出力:

```
{  
  "PhoneNumberErrors": []  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector Groups の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociatePhoneNumbersFromVoiceConnectorGroup](#)」の「」を参照してください。

disassociate-phone-numbers-from-voice-connector

次の例は、`disassociate-phone-numbers-from-voice-connector` を使用する方法を説明しています。

AWS CLI

Amazon Chime Voice Connector から電話番号の関連付けを解除するには

次の`disassociate-phone-numbers-from-voice-connector`例では、Amazon Chime Voice Connector から指定された電話番号の関連付けを解除します。

```
aws chime disassociate-phone-numbers-from-voice-connector \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --e164-phone-numbers " +12065550100" "+12065550101"
```

出力:

```
{
  "PhoneNumberErrors": []
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociatePhoneNumbersFromVoiceConnector](#)」の「」を参照してください。

disassociate-signin-delegate-groups-from-account

次の例は、`disassociate-signin-delegate-groups-from-account` を使用する方法を説明しています。

AWS CLI

サインイン委任グループの関連付けを解除するには

次のdisassociate-signin-delegate-groups-from-account例では、指定されたサインイン委任グループと指定された Amazon Chime アカウントとの関連付けを解除します。

```
aws chime disassociate-signin-delegate-groups-from-account \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --group-names "my_users"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Chime 管理ガイド」の[「ユーザーアクセスとアクセス許可の管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateSigninDelegateGroupsFromAccount](#)」の「」を参照してください。

get-account-settings

次の例は、get-account-settings を使用する方法を説明しています。

AWS CLI

アカウントの設定を取得するには

次のget-account-settings例では、指定されたアカウントのアカウント設定を取得します。

```
aws chime get-account-settings --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

出力:

```
{  
  "AccountSettings": {  
    "DisableRemoteControl": false,  
    "EnableDialOut": false  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」](#)の[「Amazon Chime アカウントの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAccountSettings](#)」の「」を参照してください。

get-account

次の例は、get-account を使用する方法を説明しています。

AWS CLI

アカウントの詳細を取得するには

次のget-account例では、指定された Amazon Chime アカウントの詳細を取得します。

```
aws chime get-account \  
--account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

出力:

```
{  
  "Account": {  
    "AwsAccountId": "111122223333",  
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "Name": "EnterpriseDirectory",  
    "AccountType": "EnterpriseDirectory",  
    "CreatedTimestamp": "2018-12-20T18:38:02.181Z",  
    "DefaultLicense": "Pro",  
    "SupportedLicenses": [  
      "Basic",  
      "Pro"  
    ],  
    "SigninDelegateGroups": [  
      {  
        "GroupName": "myGroup"  
      },  
    ]  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime アカウントの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAccount](#)」の「」を参照してください。

get-bot

次のコード例は、get-bot を使用する方法を示しています。

AWS CLI

ボットの詳細を取得するには

次のget-bot例では、指定されたボットの詳細を表示します。

```
aws chime get-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

出力:

```
{  
  "Bot": {  
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "DisplayName": "myBot (Bot)",  
    "BotType": "ChatBot",  
    "Disabled": false,  
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "BotEmail": "myBot-chimebot@example.com",  
    "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
  }  
}
```

詳細については、「Amazon Chime デベロッパーガイド」の「[チャットボットの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBot](#)」の「」を参照してください。

get-global-settings

次の例は、get-global-settings を使用する方法を説明しています。

AWS CLI

グローバル設定を取得するには

次の`get-global-settings`例では、管理者 AWS のアカウントに関連付けられた Amazon Chime Business Calling および Amazon Chime Voice Connectors の通話詳細レコードの保存に使用される S3 バケット名を取得します。

```
aws chime get-global-settings
```

出力:

```
{
  "BusinessCalling": {
    "CdrBucket": "s3bucket"
  },
  "VoiceConnector": {
    "CdrBucket": "s3bucket"
  }
}
```

詳細については、「Amazon Chime 管理ガイド」の「[グローバル設定の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGlobalSettings](#)」の「」を参照してください。

get-phone-number-order

次のコード例は、`get-phone-number-order` を使用する方法を示しています。

AWS CLI

電話番号の注文の詳細を取得するには

次の`get-phone-number-order`例では、指定された電話番号の順序の詳細を表示します。

```
aws chime get-phone-number-order \
  --phone-number-order-id abc12345-de67-89f0-123g-h45i678j9012
```

出力:

```
{
  "PhoneNumberOrder": {
    "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
```

```
"ProductType": "VoiceConnector",
"Status": "Partial",
"OrderedPhoneNumbers": [
  {
    "E164PhoneNumber": "+12065550100",
    "Status": "Acquired"
  },
  {
    "E164PhoneNumber": "+12065550101",
    "Status": "Acquired"
  },
  {
    "E164PhoneNumber": "+12065550102",
    "Status": "Failed"
  }
],
"CreatedTimestamp": "2019-08-09T21:35:21.427Z",
"UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetPhoneNumberOrder](#)」の「」を参照してください。

get-phone-number-settings

次のコード例は、get-phone-number-settings を使用する方法を示しています。

AWS CLI

アウトバウンド呼び出し名を取得するには

次のget-phone-number-settings例では、呼び出し元のユーザーの AWS アカウントのデフォルトのアウトバウンド呼び出し名を取得します。

```
aws chime get-phone-number-settings
```

このコマンドでは何も出力されません。出力:

```
{
```

```
"CallingName": "myName",
"CallingNameUpdatedTimestamp": "2019-10-28T18:56:42.911Z"
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPhoneNumberSettings](#)」の「」を参照してください。

get-phone-number

次のコード例は、get-phone-number を使用する方法を示しています。

AWS CLI

電話番号の詳細を取得するには

次のget-phone-number例では、指定された電話番号の詳細を表示します。

```
aws chime get-phone-number \
  --phone-number-id +12065550100
```

出力:

```
{
  "PhoneNumber": {
    "PhoneNumberId": "%2B12065550100",
    "E164PhoneNumber": "+12065550100",
    "Type": "Local",
    "ProductType": "VoiceConnector",
    "Status": "Unassigned",
    "Capabilities": {
      "InboundCall": true,
      "OutboundCall": true,
      "InboundSMS": true,
      "OutboundSMS": true,
      "InboundMMS": true,
      "OutboundMMS": true
    },
    "Associations": [
      {
        "Value": "abcdef1ghij2klmno3pqr4",
```

```

        "Name": "VoiceConnectorId",
        "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
      }
    ],
    "CallingNameStatus": "UpdateInProgress",
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
    "UpdatedTimestamp": "2019-08-09T21:35:31.745Z"
  }
}

```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPhoneNumber](#)」の「」を参照してください。

get-proxy-session

次のコード例は、get-proxy-session を使用する方法を示しています。

AWS CLI

プロキシセッションの詳細を取得するには

次のget-proxy-session例では、指定されたプロキシセッションの詳細を一覧表示します。

```

aws chime get-proxy-session \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891

```

出力:

```

{
  "ProxySession": {
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
    "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",
    "Status": "Open",
    "ExpiryMinutes": 60,
    "Capabilities": [
      "SMS",
      "Voice"
    ],
    "CreatedTimestamp": "2020-04-15T16:10:10.288Z",
  }
}

```

```
"UpdatedTimestamp": "2020-04-15T16:10:10.288Z",
"Participants": [
  {
    "PhoneNumber": "+12065550100",
    "ProxyPhoneNumber": "+19135550199"
  },
  {
    "PhoneNumber": "+14015550101",
    "ProxyPhoneNumber": "+19135550199"
  }
]
}
```

詳細については、「Amazon Chime デベロッパーガイド」の「[プロキシフォンセッション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetProxySession](#)」の「」を参照してください。

get-room

次の例は、get-room を使用する方法を説明しています。

AWS CLI

チャットルームの詳細を取得するには

次のget-room例では、指定されたチャットルームの詳細を表示します。

```
aws chime get-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

出力:

```
{
  "Room": {
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
    "Name": "chatRoom",
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",
```

```
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",  
    "UpdatedTimestamp": "2019-12-02T22:29:31.549Z"  
  }  
}
```

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRoom](#)」の「」を参照してください。

get-user-settings

次の例は、get-user-settings を使用する方法を説明しています。

AWS CLI

ユーザー設定を取得するには

次のget-user-settings例では、指定されたユーザー設定を表示します。

```
aws chime get-user-settings \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

出力:

```
{  
  "UserSettings": {  
    "Telephony": {  
      "InboundCalling": true,  
      "OutboundCalling": true,  
      "SMS": true  
    }  
  }  
}
```

詳細については、「Amazon Chime 管理ガイド」の[「ユーザー電話番号の管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUserSettings](#)」の「」を参照してください。

get-user

次のコード例は、get-user を使用する方法を示しています。

AWS CLI

ユーザーの詳細を取得するには

次のget-user例では、指定されたユーザーの詳細を取得します。

```
aws chime get-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

出力:

```
{  
  "User": {  
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "PrimaryEmail": "marthar@example.com",  
    "DisplayName": "Martha Rivera",  
    "LicenseType": "Pro",  
    "UserRegistrationStatus": "Registered",  
    "RegisteredOn": "2018-12-20T18:45:25.231Z",  
    "InvitedOn": "2018-12-20T18:45:25.231Z",  
    "AlexaForBusinessMetadata": {  
      "IsAlexaForBusinessEnabled": False,  
      "AlexaForBusinessRoomArn": "null"  
    },  
    "PersonalPIN": "XXXXXXXXXX"  
  }  
}
```

詳細については、「Amazon Chime 管理ガイド」の「[ユーザーの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUser](#)」の「」を参照してください。

get-voice-connector-group

次のコード例は、get-voice-connector-group を使用する方法を示しています。

AWS CLI

Amazon Chime Voice Connector グループの詳細を取得するには

次の`get-voice-connector-group`例では、指定された Amazon Chime Voice Connector グループの詳細を表示します。

```
aws chime get-voice-connector-group \  
--voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

出力:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
    "Name": "myGroup",  
    "VoiceConnectorItems": [],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector Groups の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVoiceConnectorGroup](#)」の「」を参照してください。

get-voice-connector-logging-configuration

次のコード例は、`get-voice-connector-logging-configuration` を使用する方法を示しています。

AWS CLI

ログ記録設定の詳細を取得するには

次の`get-voice-connector-logging-configuration`例では、指定された Amazon Chime Voice Connector のログ記録設定の詳細を取得します。

```
aws chime get-voice-connector-logging-configuration \  

```

```
--voice-connector-id abcdef1ghij2klmno3pqr4
```

出力:

```
{
  "LoggingConfiguration": {
    "EnableSIPLogs": true
  }
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector Media から Kinesis へのストリーミング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetVoiceConnectorLoggingConfiguration](#)」の「」を参照してください。

get-voice-connector-origination

次の例は、get-voice-connector-origination を使用する方法を説明しています。

AWS CLI

オリジン設定を取得するには

次のget-voice-connector-origination例では、指定された Amazon Chime Voice Connector のオリジンホスト、ポート、プロトコル、優先度、および重みを取得します。

```
aws chime get-voice-connector-origination \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

出力:

```
{
  "Origination": {
    "Routes": [
      {
        "Host": "10.24.34.0",
        "Port": 1234,
        "Protocol": "TCP",
        "Priority": 1,
        "Weight": 5
      }
    ]
  }
}
```

```
    ],  
    "Disabled": false  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVoiceConnectorOrigination](#)」の「」を参照してください。

get-voice-connector-proxy

次の例は、get-voice-connector-proxy を使用する方法を説明しています。

AWS CLI

プロキシ設定の詳細を取得するには

次のget-voice-connector-proxy例では、Amazon Chime Voice Connector のプロキシ設定の詳細を取得します。

```
aws chime get-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

出力:

```
{  
  "Proxy": {  
    "DefaultSessionExpiryMinutes": 60,  
    "Disabled": false,  
    "PhoneNumberCountries": [  
      "US"  
    ]  
  }  
}
```

詳細については、「Amazon Chime デベロッパーガイド」の[「プロキシフォンセッション」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVoiceConnectorProxy](#)」の「」を参照してください。

get-voice-connector-streaming-configuration

次のコード例は、`get-voice-connector-streaming-configuration` を使用する方法を示しています。

AWS CLI

ストリーミング設定の詳細を取得するには

次の`get-voice-connector-streaming-configuration`例では、指定された Amazon Chime Voice Connector のストリーミング設定の詳細を取得します。

```
aws chime get-voice-connector-streaming-configuration \  
--voice-connector-id abcdef1ghij2klmno3pqr4
```

出力:

```
{  
  "StreamingConfiguration": {  
    "DataRetentionInHours": 24,  
    "Disabled": false  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector データを Kinesis にストリーミングする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetVoiceConnectorStreamingConfiguration](#)」の「」を参照してください。

get-voice-connector-termination-health

次の例は、`get-voice-connector-termination-health` を使用する方法を説明しています。

AWS CLI

終了ヘルスの詳細を取得するには

次の`get-voice-connector-termination-health`例では、指定された Amazon Chime Voice Connector の終了ヘルスの詳細を取得します。

```
aws chime get-voice-connector-termination-health \  

```

```
--voice-connector-id abcdef1ghij2klmno3pqr4
```

出力:

```
{
  "TerminationHealth": {
    "Timestamp": "Fri Aug 23 16:45:55 UTC 2019",
    "Source": "10.24.34.0"
  }
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetVoiceConnectorTerminationHealth](#)」の「」を参照してください。

get-voice-connector-termination

次の例は、get-voice-connector-termination を使用する方法を説明しています。

AWS CLI

終了設定を取得するには

次のget-voice-connector-termination例では、指定された Amazon Chime Voice Connector の終了設定を取得します。

```
aws chime get-voice-connector-termination \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

このコマンドでは何も出力されません。出力:

```
{
  "Termination": {
    "CpsLimit": 1,
    "DefaultPhoneNumber": "+12065550100",
    "CallingRegions": [
      "US"
    ],
    "CidrAllowedList": [
      "10.24.34.0/23"
    ]
  }
}
```

```
    ],  
    "Disabled": false  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVoiceConnectorTermination](#)」の「」を参照してください。

get-voice-connector

次のコード例は、get-voice-connector を使用する方法を示しています。

AWS CLI

Amazon Chime Voice Connector の詳細を取得するには

次のget-voice-connector例は、指定された Amazon Chime Voice Connector の詳細を示しています。

```
aws chime get-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

出力:

```
{  
  "VoiceConnector": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "AwsRegion": "us-west-2",  
    "Name": "newVoiceConnector",  
    "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",  
    "RequireEncryption": true,  
    "CreatedTimestamp": "2019-09-18T20:34:01.352Z",  
    "UpdatedTimestamp": "2019-09-18T20:34:01.352Z"  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVoiceConnector](#)」の「」を参照してください。

invite-users

次の例は、invite-users を使用する方法を説明しています。

AWS CLI

Amazon Chime に参加するようにユーザーを招待するには

次のinvite-users例では、指定された Amazon Chime アカウントにユーザーを招待する E メールを送信します。

```
aws chime invite-users \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-email-list "alejandr@example.com" "janed@example.com"
```

出力:

```
{  
  "Invites": [  
    {  
      "InviteId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
      "Status": "Pending",  
      "EmailAddress": "alejandr@example.com",  
      "EmailStatus": "Sent"  
    }  
    {  
      "InviteId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
      "Status": "Pending",  
      "EmailAddress": "janed@example.com",  
      "EmailStatus": "Sent"  
    }  
  ]  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「ユーザーの招待と停止」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[InviteUsers](#)」の「」を参照してください。

list-accounts

次の例は、list-accounts を使用する方法を説明しています。

AWS CLI

アカウントのリストを取得するには

次のlist-accounts例では、管理者のアカウントの Amazon Chime AWS アカウントのリストを取得します。

```
aws chime list-accounts
```

出力:

```
{
  "Accounts": [
    {
      "AwsAccountId": "111122223333",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "Name": "First Chime Account",
      "AccountType": "EnterpriseDirectory",
      "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
      "DefaultLicense": "Pro",
      "SupportedLicenses": [
        "Basic",
        "Pro"
      ],
      "SigninDelegateGroups": [
        {
          "GroupName": "myGroup"
        }
      ]
    },
    {
      "AwsAccountId": "111122223333",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "Name": "Second Chime Account",
      "AccountType": "Team",
      "CreatedTimestamp": "2018-09-04T21:44:22.292Z",
      "DefaultLicense": "Pro",
      "SupportedLicenses": [
        "Basic",
```



```
        "Pro"
      ],
      "SignInDelegateGroups": [
        {
          "GroupName": "myGroup"
        },
      ]
    }
  ]
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime アカウントの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccounts](#)」の「」を参照してください。

list-bots

次のコード例は、list-bots を使用する方法を示しています。

AWS CLI

ボットのリストを取得するには

次のlist-bots例では、指定された Amazon Chime Enterprise アカウントに関連付けられているボットを一覧表示します。

```
aws chime list-bots \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45
```

出力:

```
{
  "Bot": {
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",
    "DisplayName": "myBot (Bot)",
    "BotType": "ChatBot",
    "Disabled": false,
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",
```

```
    "BotEmail": "myBot-chimebot@example.com",
    "SecurityToken": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
  }
}
```

詳細については、[「Amazon Chime デベロッパーガイド」の「Amazon Chime でチャットボットを使用する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBots](#)」の「」を参照してください。

list-phone-number-orders

次の例は、list-phone-number-orders を使用する方法を説明しています。

AWS CLI

電話番号の注文を一覧表示するには

次のlist-phone-number-orders例では、Amazon Chime 管理者のアカウントに関連付けられた電話番号の注文を一覧表示します。

```
aws chime list-phone-number-orders
```

出力:

```
{
  "PhoneNumberOrders": [
    {
      "PhoneNumberOrderId": "abc12345-de67-89f0-123g-h45i678j9012",
      "ProductType": "VoiceConnector",
      "Status": "Partial",
      "OrderedPhoneNumbers": [
        {
          "E164PhoneNumber": "+12065550100",
          "Status": "Acquired"
        },
        {
          "E164PhoneNumber": "+12065550101",
          "Status": "Acquired"
        },
        {
```

```
        "E164PhoneNumber": "+12065550102",
        "Status": "Failed"
      }
    ],
    "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
    "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
  }
  {
    "PhoneNumberOrderId": "cba54321-ed76-09f5-321g-h54i876j2109",
    "ProductType": "BusinessCalling",
    "Status": "Partial",
    "OrderedPhoneNumbers": [
      {
        "E164PhoneNumber": "+12065550103",
        "Status": "Acquired"
      },
      {
        "E164PhoneNumber": "+12065550104",
        "Status": "Acquired"
      },
      {
        "E164PhoneNumber": "+12065550105",
        "Status": "Failed"
      }
    ],
    "CreatedTimestamp": "2019-08-09T21:35:21.427Z",
    "UpdatedTimestamp": "2019-08-09T21:35:31.926Z"
  }
]
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPhoneNumberOrders](#)」の「」を参照してください。

list-phone-numbers

次の例は、list-phone-numbers を使用する方法を説明しています。

AWS CLI

Amazon Chime アカウントの電話番号を一覧表示するには

次のlist-phone-numbers例では、管理者の Amazon Chime アカウントに関連付けられている電話番号を一覧表示します。

```
aws chime list-phone-numbers
```

このコマンドでは何も出力されません。出力:

```
{
  "PhoneNumbers": [
    {
      "PhoneNumberId": "%2B12065550100",
      "E164PhoneNumber": "+12065550100",
      "Type": "Local",
      "ProductType": "VoiceConnector",
      "Status": "Assigned",
      "Capabilities": {
        "InboundCall": true,
        "OutboundCall": true,
        "InboundSMS": true,
        "OutboundSMS": true,
        "InboundMMS": true,
        "OutboundMMS": true
      },
      "Associations": [
        {
          "Value": "abcdef1ghij2klmno3pqr4",
          "Name": "VoiceConnectorId",
          "AssociatedTimestamp": "2019-10-28T18:40:37.453Z"
        }
      ],
      "CallingNameStatus": "UpdateInProgress",
      "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
      "UpdatedTimestamp": "2019-10-28T18:42:07.964Z"
    },
    {
      "PhoneNumberId": "%2B12065550101",
      "E164PhoneNumber": "+12065550101",
      "Type": "Local",
      "ProductType": "VoiceConnector",
      "Status": "Assigned",
      "Capabilities": {
        "InboundCall": true,
        "OutboundCall": true,
```

```
        "InboundSMS": true,
        "OutboundSMS": true,
        "InboundMMS": true,
        "OutboundMMS": true
    },
    "Associations": [
        {
            "Value": "abcdef1ghij2klmno3pqr4",
            "Name": "VoiceConnectorId",
            "AssociatedTimestamp": "2019-10-28T18:40:37.511Z"
        }
    ],
    "CallingNameStatus": "UpdateInProgress",
    "CreatedTimestamp": "2019-08-12T22:10:20.521Z",
    "UpdatedTimestamp": "2019-10-28T18:42:07.960Z"
}
]
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPhoneNumbers](#)」の「」を参照してください。

list-proxy-sessions

次のコード例は、list-proxy-sessions を使用する方法を示しています。

AWS CLI

プロキシセッションを一覧表示するには

次のlist-proxy-sessions例では、Amazon Chime Voice Connector のプロキシセッションを一覧表示します。

```
aws chime list-proxy-sessions \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

出力:

```
{
  "ProxySession": {
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
```

```
"ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",
"Status": "Open",
"ExpiryMinutes": 60,
"Capabilities": [
  "SMS",
  "Voice"
],
"CreatedTimestamp": "2020-04-15T16:10:10.288Z",
"UpdatedTimestamp": "2020-04-15T16:10:10.288Z",
"Participants": [
  {
    "PhoneNumber": "+12065550100",
    "ProxyPhoneNumber": "+19135550199"
  },
  {
    "PhoneNumber": "+14015550101",
    "ProxyPhoneNumber": "+19135550199"
  }
]
}
```

詳細については、「Amazon Chime デベロッパーガイド」の「[プロキシフォンセッション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProxySessions](#)」の「」を参照してください。

list-room-memberships

次のコード例は、list-room-memberships を使用する方法を示しています。

AWS CLI

ルームメンバーシップを一覧表示するには

次のlist-room-memberships例では、指定されたチャットルームのメンバーシップの詳細のリストを表示します。

```
aws chime list-room-memberships \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

出力:

```
{
  "RoomMemberships": [
    {
      "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
      "Member": {
        "MemberId": "2ab2345c-67de-8901-f23g-45h678901j2k",
        "MemberType": "User",
        "Email": "zhangw@example.com",
        "FullName": "Zhang Wei",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
      },
      "Role": "Member",
      "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
      "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
    },
    {
      "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",
      "Member": {
        "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",
        "MemberType": "User",
        "Email": "janed@example.com",
        "FullName": "Jane Doe",
        "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"
      },
      "Role": "Administrator",
      "InvitedBy": "arn:aws:iam::111122223333:user/alejandro",
      "UpdatedTimestamp": "2019-12-02T22:46:58.532Z"
    }
  ]
}
```

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRoomMemberships](#)」の「」を参照してください。

list-rooms

次のコード例は、list-rooms を使用する方法を示しています。

AWS CLI

チャットルームを一覧表示するには

次のlist-rooms例では、指定されたアカウントのチャットルームのリストを表示します。リストは、指定されたメンバーが属するチャットルームにのみフィルタリングされます。

```
aws chime list-rooms \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

出力:

```
{  
  "Room": {  
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",  
    "Name": "teamRoom",  
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",  
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",  
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",  
    "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"  
  }  
}
```

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRooms](#)」の「」を参照してください。

list-users

次のコード例は、list-users を使用する方法を示しています。

AWS CLI

アカウントのユーザーを一覧表示するには

次のlist-users例では、指定された Amazon Chime アカウントのユーザーを一覧表示します。

```
aws chime list-users --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```


出力:

```
{
  "Users": [
    {
      "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "PrimaryEmail": "mariag@example.com",
      "DisplayName": "Maria Garcia",
      "LicenseType": "Pro",
      "UserType": "PrivateUser",
      "UserRegistrationStatus": "Registered",
      "RegisteredOn": "2018-12-20T18:45:25.231Z"
      "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
      }
    },
    {
      "UserId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "PrimaryEmail": "richardr@example.com",
      "DisplayName": "Richard Roe",
      "LicenseType": "Pro",
      "UserType": "PrivateUser",
      "UserRegistrationStatus": "Registered",
      "RegisteredOn": "2018-12-20T18:45:45.415Z"
      "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
      }
    },
    {
      "UserId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
      "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "PrimaryEmail": "saanvis@example.com",
      "DisplayName": "Saanvi Sarkar",
      "LicenseType": "Basic",
      "UserType": "PrivateUser",
      "UserRegistrationStatus": "Registered",
      "RegisteredOn": "2018-12-20T18:46:57.747Z"
      "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
      }
    },
    {
```

```
    "UserId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "PrimaryEmail": "wxiulan@example.com",
    "DisplayName": "Wang Xiulan",
    "LicenseType": "Basic",
    "UserType": "PrivateUser",
    "UserRegistrationStatus": "Registered",
    "RegisteredOn": "2018-12-20T18:47:15.390Z"
    "AlexaForBusinessMetadata": {
        "IsAlexaForBusinessEnabled": false
    }
  }
]
```

詳細については、「Amazon Chime 管理ガイド」の「[ユーザーの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUsers](#)」の「」を参照してください。

list-voice-connector-groups

次の例は、list-voice-connector-groups を使用する方法を説明しています。

AWS CLI

Amazon Chime アカウントの Amazon Chime Voice Connector グループを一覧表示するには

次のlist-voice-connector-groups例では、管理者の Amazon Chime アカウントに関連付けられている Amazon Chime Voice Connector グループを一覧表示します。

```
aws chime list-voice-connector-groups
```

出力:

```
{
  "VoiceConnectorGroups": [
    {
      "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",
      "Name": "myGroup",
      "VoiceConnectorItems": [],
    }
  ]
}
```

```
        "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
        "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
    }
]
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector グループの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVoiceConnectorGroups](#)」の「」を参照してください。

list-voice-connector-termination-credentials

次のコード例は、`list-voice-connector-termination-credentials` を使用する方法を示しています。

AWS CLI

終了認証情報のリストを取得するには

次の`list-voice-connector-termination-credentials`例では、指定された Amazon Chime Voice Connector の終了認証情報のリストを取得します。

```
aws chime list-voice-connector-termination-credentials \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

このコマンドでは何も出力されません。出力:

```
{
  "Usernames": [
    "jdoe"
  ]
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVoiceConnectorTerminationCredentials](#)」の「」を参照してください。

list-voice-connectors

次のコード例は、list-voice-connectors を使用する方法を示しています。

AWS CLI

アカウントの Amazon Chime Voice Connectors を一覧表示するには

次のlist-voice-connectors例では、発信者のアカウントに関連付けられている Amazon Chime Voice Connectors を一覧表示します。

```
aws chime list-voice-connectors
```

出力:

```
{
  "VoiceConnectors": [
    {
      "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
      "AwsRegion": "us-east-1",
      "Name": "MyVoiceConnector",
      "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",
      "RequireEncryption": true,
      "CreatedTimestamp": "2019-06-04T18:46:56.508Z",
      "UpdatedTimestamp": "2019-09-18T16:33:00.806Z"
    },
    {
      "VoiceConnectorId": "cbadef1ghij2klmno3pqr5",
      "AwsRegion": "us-west-2",
      "Name": "newVoiceConnector",
      "OutboundHostName": "cbadef1ghij2klmno3pqr5.voiceconnector.chime.aws",
      "RequireEncryption": true,
      "CreatedTimestamp": "2019-09-18T20:34:01.352Z",
      "UpdatedTimestamp": "2019-09-18T20:34:01.352Z"
    }
  ]
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVoiceConnectors](#)」の「」を参照してください。

logout-user

次のコード例は、logout-user を使用する方法を示しています。

AWS CLI

ユーザーをログアウトするには

次のlogout-user例では、指定されたユーザーをログアウトします。

```
aws chime logout-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[LogoutUser](#)」の「」を参照してください。

put-voice-connector-logging-configuration

次のコード例は、put-voice-connector-logging-configuration を使用する方法を示しています。

AWS CLI

Amazon Chime Voice Connector のログ記録設定を追加するには

次のput-voice-connector-logging-configuration例では、指定された Amazon Chime Voice Connector のSIPログ記録設定をオンにします。

```
aws chime put-voice-connector-logging-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --logging-configuration EnableSIPLogs=true
```

出力:

```
{  
  "LoggingConfiguration": {  
    "EnableSIPLogs": true  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector Media から Kinesis へのストリーミング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutVoiceConnectorLoggingConfiguration](#)」の「」を参照してください。

put-voice-connector-origination

次のコード例は、put-voice-connector-origination を使用する方法を示しています。

AWS CLI

発信元設定を設定するには

次のput-voice-connector-origination例では、指定された Amazon Chime Voice Connector の発信元ホスト、ポート、プロトコル、優先度、および重みを設定します。

```
aws chime put-voice-connector-origination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --origination  
  Routes=[{Host="10.24.34.0",Port=1234,Protocol="TCP",Priority=1,Weight=5}],Disabled=false
```

出力:

```
{  
  "Origination": {  
    "Routes": [  
      {  
        "Host": "10.24.34.0",  
        "Port": 1234,  
        "Protocol": "TCP",  
        "Priority": 1,  
        "Weight": 5  
      }  
    ],  
    "Disabled": false  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutVoiceConnectorOrigination](#)」の「」を参照してください。

put-voice-connector-proxy

次のコード例は、put-voice-connector-proxy を使用する方法を示しています。

AWS CLI

プロキシ設定を配置するには

次のput-voice-connector-proxy例では、Amazon Chime Voice Connector にプロキシ設定を設定します。

```
aws chime put-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --default-session-expiry-minutes 60 \  
  --phone-number-pool-countries "US"
```

出力:

```
{  
  "Proxy": {  
    "DefaultSessionExpiryMinutes": 60,  
    "Disabled": false,  
    "PhoneNumberCountries": [  
      "US"  
    ]  
  }  
}
```

詳細については、「Amazon Chime デベロッパーガイド」の「[プロキシフォンセッション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutVoiceConnectorProxy](#)」の「」を参照してください。

put-voice-connector-streaming-configuration

次のコード例は、put-voice-connector-streaming-configuration を使用する方法を示しています。

AWS CLI

ストリーミング設定を作成するには

次のput-voice-connector-streaming-configuration例では、指定された Amazon Chime Voice Connector のストリーミング設定を作成します。Amazon Chime Voice Connector から Amazon Kinesis へのメディアストリーミングを有効にし、データ保持期間を 24 時間に設定します。

```
aws chime put-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --streaming-configuration DataRetentionInHours=24,Disabled=false
```

出力:

```
{  
  "StreamingConfiguration": {  
    "DataRetentionInHours": 24,  
    "Disabled": false  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector データを Kinesis にストリーミングする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutVoiceConnectorStreamingConfiguration](#)」の「」を参照してください。

put-voice-connector-termination-credentials

次の例は、put-voice-connector-termination-credentials を使用する方法を説明しています。

AWS CLI

終了認証情報を設定するには

次のput-voice-connector-termination-credentials例では、指定された Amazon Chime Voice Connector の終了認証情報を設定します。

```
aws chime put-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --termination-credentials ...
```



```
--credentials Username="jdoe",Password="XXXXXXXX"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutVoiceConnectorTerminationCredentials](#)」の「」を参照してください。

put-voice-connector-termination

次のコード例は、put-voice-connector-termination を使用する方法を示しています。

AWS CLI

終了設定を設定するには

次のput-voice-connector-termination例では、指定された Amazon Chime Voice Connector の呼び出しリージョンと許可された IP ホスト終了設定を設定します。

```
aws chime put-voice-connector-termination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --termination CallingRegions="US",CidrAllowedList="10.24.34.0/23",Disabled=false
```

出力:

```
{  
  "Termination": {  
    "CpsLimit": 0,  
    "CallingRegions": [  
      "US"  
    ],  
    "CidrAllowedList": [  
      "10.24.34.0/23"  
    ],  
    "Disabled": false  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutVoiceConnectorTermination](#)」の「」を参照してください。

regenerate-security-token

次のコード例は、regenerate-security-token を使用する方法を示しています。

AWS CLI

セキュリティトークンを再生成するには

次のregenerate-security-token例では、指定されたボットのセキュリティトークンを再生成します。

```
aws chime regenerate-security-token \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

出力:

```
{  
  "Bot": {  
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "DisplayName": "myBot (Bot)",  
    "BotType": "ChatBot",  
    "Disabled": false,  
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "BotEmail": "myBot-chimebot@example.com",  
    "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"  
  }  
}
```

詳細については、「Amazon Chime デベロッパーガイド」の「[チャットボットリクエストの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegenerateSecurityToken](#)」の「」を参照してください。

reset-personal-pin

次のコード例は、reset-personal-pin を使用する方法を示しています。

AWS CLI

ユーザーの個人会議をリセットするには PIN

次のreset-personal-pin例では、指定されたユーザーの個人会議 をリセットしますPIN。

```
aws chime reset-personal-pin \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

出力:

```
{  
  "User": {  
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "PrimaryEmail": "mateo@example.com",  
    "DisplayName": "Mateo Jackson",  
    "LicenseType": "Pro",  
    "UserType": "PrivateUser",  
    "UserRegistrationStatus": "Registered",  
    "RegisteredOn": "2018-12-20T18:45:25.231Z",  
    "AlexaForBusinessMetadata": {  
      "IsAlexaForBusinessEnabled": False,  
      "AlexaForBusinessRoomArn": "null"  
    },  
    "PersonalPIN": "XXXXXXXXXX"  
  }  
}
```

詳細については、「Amazon Chime 管理ガイド」の「[個人会議の変更PINs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetPersonalPin](#)」の「」を参照してください。

restore-phone-number

次の例は、restore-phone-number を使用する方法を説明しています。

AWS CLI

電話番号を復元するには

次の`restore-phone-number`例では、指定された電話番号を削除キューから復元します。

```
aws chime restore-phone-number \  
  --phone-number-id "+12065550100"
```

出力:

```
{  
  "PhoneNumber": {  
    "PhoneNumberId": "%2B12065550100",  
    "E164PhoneNumber": "+12065550100",  
    "Type": "Local",  
    "ProductType": "BusinessCalling",  
    "Status": "Unassigned",  
    "Capabilities": {  
      "InboundCall": true,  
      "OutboundCall": true,  
      "InboundSMS": true,  
      "OutboundSMS": true,  
      "InboundMMS": true,  
      "OutboundMMS": true  
    },  
    "Associations": [],  
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",  
    "UpdatedTimestamp": "2019-08-12T22:06:36.355Z"  
  }  
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestorePhoneNumber](#)」の「」を参照してください。

`search-available-phone-numbers`

次の例は、`search-available-phone-numbers` を使用する方法を説明しています。

AWS CLI

使用可能な電話番号を検索するには

次のsearch-available-phone-numbers例では、市外局番で使用可能な電話番号を検索します。

```
aws chime search-available-phone-numbers \  
  --area-code "206"
```

出力:

```
{  
  "E164PhoneNumbers": [  
    "+12065550100",  
    "+12065550101",  
    "+12065550102",  
    "+12065550103",  
    "+12065550104",  
    "+12065550105",  
    "+12065550106",  
    "+12065550107",  
    "+12065550108",  
    "+12065550109",  
  ]  
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスSearchAvailablePhoneNumbers](#)」の「」を参照してください。

update-account-settings

次の例は、update-account-settings を使用する方法を説明しています。

AWS CLI

アカウントの設定を更新するには

次のupdate-account-settings例では、指定された Amazon Chime アカウントの共有画面のリモートコントロールを無効にします。

```
aws chime update-account-settings \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --account-settings DisableRemoteControl=true
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAccountSettings](#)」の「」を参照してください。

update-account

次のコード例は、update-account を使用する方法を示しています。

AWS CLI

アカウントを更新するには

次のupdate-account例では、指定されたアカウント名を更新します。

```
aws chime update-account \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --name MyAccountName
```

出力:

```
{  
  "Account": {  
    "AwsAccountId": "111122223333",  
    "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "Name": "MyAccountName",  
    "AccountType": "Team",  
    "CreatedTimestamp": "2018-09-04T21:44:22.292Z",  
    "DefaultLicense": "Pro",  
    "SupportedLicenses": [  
      "Basic",  
      "Pro"  
    ],  
    "SigninDelegateGroups": [  
      {  
        "GroupName": "myGroup"  
      },  
    ]  
  }  
}
```

```
}  
}
```

詳細については、「Amazon Chime 管理ガイド」の[「アカウントの名前変更」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAccount](#)」の「」を参照してください。

update-bot

次のコード例は、update-bot を使用する方法を示しています。

AWS CLI

ボットを更新するには

次のupdate-bot例では、指定されたボットのステータスを更新して、実行を停止します。

```
aws chime update-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k \  
  --disabled
```

出力:

```
{  
  "Bot": {  
    "BotId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "UserId": "123abcd4-5ef6-789g-0h12-34j56789012k",  
    "DisplayName": "myBot (Bot)",  
    "BotType": "ChatBot",  
    "Disabled": true,  
    "CreatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "UpdatedTimestamp": "2019-09-09T18:05:56.749Z",  
    "BotEmail": "myBot-chimebot@example.com",  
    "SecurityToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"  
  }  
}
```

詳細については、「Amazon Chime デベロッパーガイド」の[「チャットボットの更新」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateBot](#)」の「」を参照してください。

update-global-settings

次の例は、update-global-settings を使用する方法を説明しています。

AWS CLI

グローバル設定を更新するには

次のupdate-global-settings例では、管理者 AWS のアカウントに関連付けられた Amazon Chime Business Calling および Amazon Chime Voice Connectors の通話詳細レコードの保存に使用される S3 バケットを更新します。

```
aws chime update-global-settings \  
  --business-calling CdrBucket="s3bucket" \  
  --voice-connector CdrBucket="s3bucket"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Chime 管理ガイド」の「[グローバル設定の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateGlobalSettings](#)」の「」を参照してください。

update-phone-number-settings

次のコード例は、update-phone-number-settings を使用する方法を示しています。

AWS CLI

アウトバウンド呼び出し名を更新するには

次のupdate-phone-number-settings例では、管理者の AWS アカウントのデフォルトのアウトバウンド呼び出し名を更新します。

```
aws chime update-phone-number-settings \  
  --calling-name "myName"
```


このコマンドでは何も出力されません。

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdatePhoneNumberSettings](#)」の「」を参照してください。

update-phone-number

次の例は、update-phone-number を使用する方法を説明しています。

AWS CLI

例 1: 電話番号の製品タイプを更新するには

次のupdate-phone-number例では、指定された電話番号の製品タイプを更新します。

```
aws chime update-phone-number \  
  --phone-number-id "+12065550100" \  
  --product-type "BusinessCalling"
```

出力:

```
{  
  "PhoneNumber": {  
    "PhoneNumberId": "%2B12065550100",  
    "E164PhoneNumber": "+12065550100",  
    "Type": "Local",  
    "ProductType": "BusinessCalling",  
    "Status": "Unassigned",  
    "Capabilities": {  
      "InboundCall": true,  
      "OutboundCall": true,  
      "InboundSMS": true,  
      "OutboundSMS": true,  
      "InboundMMS": true,  
      "OutboundMMS": true  
    },  
    "Associations": [],  
    "CallingName": "phonenumbers1",  
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",  
    "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"  
  }  
}
```

```
}
```

例 2: 電話番号のアウトバウンド呼び出し名を更新するには

次のupdate-phone-number例では、指定された電話番号のアウトバウンド呼び出し名を更新します。

```
aws チャイム update-phone-number --phone-number-id +12065550100 --calling-name 「phonenum2」
```

出力:

```
{
  "PhoneNumber": {
    "PhoneNumberId": "%2B12065550100",
    "E164PhoneNumber": "+12065550100",
    "Type": "Local",
    "ProductType": "BusinessCalling",
    "Status": "Unassigned",
    "Capabilities": {
      "InboundCall": true,
      "OutboundCall": true,
      "InboundSMS": true,
      "OutboundSMS": true,
      "InboundMMS": true,
      "OutboundMMS": true
    },
    "Associations": [],
    "CallingName": "phonenum2",
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
    "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"
  }
}
```

詳細については、「[Amazon Chime 管理ガイド](#)」の「[電話番号の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdatePhoneNumber](#)」の「」を参照してください。

update-proxy-session

次の例は、update-proxy-session を使用する方法を説明しています。

AWS CLI

プロキシセッションを更新するには

次のupdate-proxy-session例では、プロキシセッション機能を更新します。

```
aws chime update-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891 \  
  --capabilities "Voice"
```

出力:

```
{  
  "ProxySession": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk567891",  
    "Status": "Open",  
    "ExpiryMinutes": 60,  
    "Capabilities": [  
      "Voice"  
    ],  
    "CreatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "UpdatedTimestamp": "2020-04-15T16:10:10.288Z",  
    "Participants": [  
      {  
        "PhoneNumber": "+12065550100",  
        "ProxyPhoneNumber": "+19135550199"  
      },  
      {  
        "PhoneNumber": "+14015550101",  
        "ProxyPhoneNumber": "+19135550199"  
      }  
    ]  
  }  
}
```

詳細については、「Amazon Chime デベロッパーガイド」の「[プロキシフォンセッション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateProxySession](#)」の「」を参照してください。

update-room-membership

次のコード例は、update-room-membership を使用する方法を示しています。

AWS CLI

ルームメンバーシップを更新するには

次のupdate-room-membership例では、指定されたチャットルームメンバーのロールを に変更しますAdministrator。

```
aws chime update-room-membership \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k \  
  --role Administrator
```

出力:

```
{  
  "RoomMembership": {  
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",  
    "Member": {  
      "MemberId": "1ab2345c-67de-8901-f23g-45h678901j2k",  
      "MemberType": "User",  
      "Email": "sofiamartinez@example.com",  
      "FullName": "Sofia Martinez",  
      "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45"  
    },  
    "Role": "Administrator",  
    "InvitedBy": "arn:aws:iam::111122223333:user/admin",  
    "UpdatedTimestamp": "2019-12-02T22:40:22.931Z"  
  }  
}
```

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRoomMembership](#)」の「」を参照してください。

update-room

次の例は、update-room を使用する方法を説明しています。

AWS CLI

チャットルームを更新するには

次のupdate-room例では、指定されたチャットルームの名前を変更します。

```
aws chime update-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --name teamRoom
```

出力:

```
{  
  "Room": {  
    "RoomId": "abcd1e2d-3e45-6789-01f2-3g45h67i890j",  
    "Name": "teamRoom",  
    "AccountId": "12a3456b-7c89-012d-3456-78901e23fg45",  
    "CreatedBy": "arn:aws:iam::111122223333:user/alejandro",  
    "CreatedTimestamp": "2019-12-02T22:29:31.549Z",  
    "UpdatedTimestamp": "2019-12-02T22:33:19.310Z"  
  }  
}
```

詳細については、[「Amazon Chime ユーザーガイド」の「チャットルームの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRoom](#)」の「」を参照してください。

update-user-settings

次のコード例は、update-user-settings を使用する方法を示しています。

AWS CLI

ユーザー設定を更新するには

次のupdate-user-settings例では、指定されたユーザーがインバウンドコールとアウトバウンドコールを行い、SMSメッセージを送受信できるようにします。

```
aws chime update-user-settings \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k \  
  --user-settings "Telephony={InboundCalling=true,OutboundCalling=true,SMS=true}"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Chime 管理ガイド」の「[ユーザー電話番号の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserSettings](#)」の「」を参照してください。

update-user

次の例は、update-user を使用する方法を説明しています。

AWS CLI

ユーザーの詳細を更新するには

この例では、指定されたユーザーの指定された詳細を更新します。

コマンド:

```
aws chime update-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \  
  --license-type "Basic"
```

出力:

```
{  
  "User": {  
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUser](#)」の「」を参照してください。

update-voice-connector-group

次のコード例は、update-voice-connector-group を使用する方法を示しています。

AWS CLI

Amazon Chime Voice Connector グループの詳細を更新するには

次のupdate-voice-connector-group例では、指定された Amazon Chime Voice Connector グループの詳細を更新します。

```
aws chime update-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --name "newGroupName" \  
  --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=1
```

出力:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
    "Name": "newGroupName",  
    "VoiceConnectorItems": [  
      {  
        "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
        "Priority": 1  
      }  
    ],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-10-28T19:00:57.081Z"  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connector Groups の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVoiceConnectorGroup](#)」の「」を参照してください。

update-voice-connector

次の例は、`update-voice-connector` を使用する方法を説明しています。

AWS CLI

Amazon Chime Voice Connector の詳細を更新するには

次の`update-voice-connector`例では、指定された Amazon Chime Voice Connector の名前を更新します。

```
aws chime update-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --name newName \  
  --require-encryption
```

出力:

```
{  
  "VoiceConnector": {  
    "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",  
    "AwsRegion": "us-west-2",  
    "Name": "newName",  
    "OutboundHostName": "abcdef1ghij2klmno3pqr4.voiceconnector.chime.aws",  
    "RequireEncryption": true,  
    "CreatedTimestamp": "2019-09-18T20:34:01.352Z",  
    "UpdatedTimestamp": "2019-09-18T20:40:52.895Z"  
  }  
}
```

詳細については、[「Amazon Chime 管理ガイド」の「Amazon Chime Voice Connectors の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVoiceConnector](#)」の「」を参照してください。

を使用した Cloud Control APIの例 AWS CLI

次のコード例は、Cloud Control AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますAPI。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-resource

次のコード例は、create-resource を使用する方法を示しています。

AWS CLI

リソースを作成するには

次のcreate-resource例では、 という名前の `AWS::Kinesis::Stream` リソースを作成します。保持期間は 168 時間 ResourceExample、シャード数は 3 です。

```
aws cloudcontrol create-resource \  
  --type-name AWS::Kinesis::Stream \  
  --desired-state '{"Name": "ResourceExample","RetentionPeriodHours":168,  
  "ShardCount":3}'
```

出力:

```
{  
  "ProgressEvent": {  
    "EventTime": 1632506656.706,  
    "TypeName": "AWS::Kinesis::Stream",  
    "OperationStatus": "IN_PROGRESS",  
    "Operation": "CREATE",  
    "Identifier": "ResourceExample",  
    "RequestToken": "20999d87-e304-4725-ad84-832dcbfd7fc5"  
  }  
}
```

詳細については、「クラウドコントロールユーザーガイド」の「[リソースの作成](#)」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResource](#)」の「」を参照してください。

delete-resource

次の例は、delete-resource を使用する方法を説明しています。

AWS CLI

リソースを削除するには

次のdelete-resource例では、AWS アカウント ResourceExample から識別子を持つ AWS::Kinesis::Stream リソースを削除します。

```
aws cloudcontrol delete-resource \  
  --type-name AWS::Kinesis::Stream \  
  --identifier ResourceExample
```

出力:

```
{  
  "ProgressEvent": {  
    "TypeName": "AWS::Kinesis::Stream",  
    "Identifier": "ResourceExample",  
    "RequestToken": "e48f26ff-d0f9-4ab8-a878-120db1edf111",  
    "Operation": "DELETE",  
    "OperationStatus": "IN_PROGRESS",  
    "EventTime": 1632950300.14  
  }  
}
```

詳細については、「Cloud Control ユーザーガイド」の「[リソースの削除](#)」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResource](#)」の「」を参照してください。

get-resource-request-status

次のコード例は、`get-resource-request-status` を使用する方法を示しています。

AWS CLI

リソースリクエストのステータス情報を取得するには

次の`get-resource-request-status`例では、指定されたリソースリクエストに関するステータス情報を返します。

```
aws cloudcontrol get-resource-request-status \  
  --request-token "e1a6b86e-46bd-41ac-bfba-001234567890"
```

出力:

```
{  
  "ProgressEvent": {  
    "TypeName": "AWS::Kinesis::Stream",  
    "Identifier": "Demo",  
    "RequestToken": "e1a6b86e-46bd-41ac-bfba-001234567890",  
    "Operation": "CREATE",  
    "OperationStatus": "FAILED",  
    "EventTime": 1632950268.481,  
    "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with identifier  
'Demo' already exists.",  
    "ErrorCode": "AlreadyExists"  
  }  
}
```

詳細については、「クラウドコントロールユーザーガイド」の[「リソースオペレーションリクエストの管理」](#)を参照してください。API

- API 詳細については、「コマンドリファレンス[GetResourceRequestStatus](#)」の「」を参照してください。AWS CLI

get-resource

次のコード例は、`get-resource` を使用する方法を示しています。

AWS CLI

リソースの現在の状態を取得するには

次のget-resource例では、 という名前の AWS::Kinesis::Stream リソースの現在の状態を返します ResourceExample。

```
aws cloudcontrol get-resource \  
  --type-name AWS::Kinesis::Stream \  
  --identifier ResourceExample
```

出力:

```
{  
  "TypeName": "AWS::Kinesis::Stream",  
  "ResourceDescription": {  
    "Identifier": "ResourceExample",  
    "Properties": "{\"Arn\":\"arn:aws:kinesis:us-west-2:099908667365:stream/ResourceExample\", \"RetentionPeriodHours\":168, \"Name\":\"ResourceExample\", \"ShardCount\":3}"  
  }  
}
```

詳細については、「クラウドコントロールAPIユーザーガイド」の「[リソースの現在の状態の読み取り](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResource](#)」の「」を参照してください。

list-resource-requests

次の例は、list-resource-requests を使用する方法を説明しています。

AWS CLI

アクティブなリソースオペレーションリクエストを一覧表示するには

次のlist-resource-requests例では、AWS アカウントで失敗した CREATEおよび UPDATEオペレーションのリソースリクエストを一覧表示します。

```
aws cloudcontrol list-resource-requests \  
  --resource-request-status-filter Operations=CREATE,OperationStatuses=FAILED
```

出力:

```
{
  "ResourceRequestStatusSummaries": [
    {
      "TypeName": "AWS::Kinesis::Stream",
      "Identifier": "Demo",
      "RequestToken": "e1a6b86e-46bd-41ac-bfba-633abcdfdbd7",
      "Operation": "CREATE",
      "OperationStatus": "FAILED",
      "EventTime": 1632950268.481,
      "StatusMessage": "Resource of type 'AWS::Kinesis::Stream' with
identifier 'Demo' already exists.",
      "ErrorCode": "AlreadyExists"
    }
  ]
}
```

詳細については、「クラウドコントロールユーザーガイド」の「[リソースオペレーションリクエストの管理](#)」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceRequests](#)」の「」を参照してください。

list-resources

次のコード例は、list-resources を使用する方法を示しています。

AWS CLI

特定のタイプのリソースを一覧表示するには

次のlist-resources例では、AWS アカウントでプロビジョニングされたAWS::Kinesis::Stream リソースを一覧表示します。

```
aws cloudcontrol list-resources \  
  --type-name AWS::Kinesis::Stream
```

出力:

```
{
  "TypeName": "AWS::Kinesis::Stream",
  "ResourceDescriptions": [
```

```

    {
      "Identifier": "MyKinesisStream",
      "Properties": "{\"Name\":\"MyKinesisStream\"}"
    },
    {
      "Identifier": "AnotherStream",
      "Properties": "{\"Name\":\"AnotherStream\"}"
    }
  ]
}

```

詳細については、「クラウドコントロールAPIユーザーガイド」の「[リソースの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResources](#)」の「」を参照してください。

update-resource

次のコード例は、update-resource を使用する方法を示しています。

AWS CLI

既存のリソースのプロパティを更新するには

次のupdate-resource例では、という名前の AWS::Logs::LogGroup resource の保持ポリシー ExampleLogGroup を 90 日に更新します。

```

aws cloudcontrol update-resource \
  --type-name AWS::Logs::LogGroup \
  --identifier ExampleLogGroup \
  --patch-document "[{"op":"replace","path":"/RetentionInDays","value":90}]"

```

出力:

```

{
  "ProgressEvent": {
    "EventTime": "2021-08-09T18:17:15.219Z",
    "TypeName": "AWS::Logs::LogGroup",
    "OperationStatus": "IN_PROGRESS",
    "Operation": "UPDATE",
  }
}

```

```
"Identifier": "ExampleLogGroup",
  "RequestToken": "5f40c577-3534-4b20-9599-0b0123456789"
}
}
```

詳細については、「Cloud Control API ユーザーガイド」の「[リソースの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResource](#)」の「」を参照してください。

AWS Cloud Map を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Cloud Map。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-private-dns-namespace

次の例は、create-private-dns-namespace を使用する方法を説明しています。

AWS CLI

プライベートDNS名前空間を作成するには

次のcreate-private-dns-namespace例では、プライベートDNS名前空間を作成します。

```
aws servicediscovery create-private-dns-namespace \
  --name example.com \
```

```
--vpc vpc-1c56417b
```

出力:

```
{  
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"  
}
```

オペレーションが成功したことを確認するには、`aws get-operation` を実行できます。詳細については、「[get-operation](#)」を参照してください。

詳細については、AWS 「Cloud Map デベロッパーガイド」の「[名前空間の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePrivateDnsNamespace](#)」の「」を参照してください。

create-service

次のコード例は、`create-service` を使用する方法を示しています。

AWS CLI

サービスを作成するには

次の `create-service` 例では、サービスを作成します。

```
aws servicediscovery create-service \  
  --name myservice \  
  --namespace-id ns-ylexjili4cdxy3xm \  
  --dns-config "NamespaceId=ns-ylexjili4cdxy3xm, RoutingPolicy=MULTIVALUE, DnsRecords=[{Type=A, TTL=60}]"
```

出力:

```
{  
  "Service": {  
    "Id": "srv-p5zdwlg5uvvzjita",  
    "Arn": "arn:aws:servicediscovery:us-west-2:803642222207:service/srv-p5zdwlg5uvvzjita",  
  }  
}
```



```
    "Name": "myservice",
    "NamespaceId": "ns-ylexjili4cdxy3xm",
    "DnsConfig": {
      "NamespaceId": "ns-ylexjili4cdxy3xm",
      "RoutingPolicy": "MULTIVALUE",
      "DnsRecords": [
        {
          "Type": "A",
          "TTL": 60
        }
      ]
    },
    "CreateDate": 1587081768.334,
    "CreatorRequestId": "567c1193-6b00-4308-bd57-ad38a8822d25"
  }
}
```

詳細については、「Cloud Map デベロッパーガイド」の「[サービスの作成](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateService](#)」の「」を参照してください。

delete-namespace

次の例は、delete-namespace を使用する方法を説明しています。

AWS CLI

名前空間を削除するには

次のdelete-namespace例では、名前空間を削除します。

```
aws servicediscovery delete-namespace \  
  --id ns-ylexjili4cdxy3xm
```

出力:

```
{  
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6drk"  
}
```



```
--instance-id myservice-53
```

出力:

```
{
  "OperationId": "4yejorelbukcjzpnr6t1mrghsjwpngf4-k98rnaiq"
}
```

オペレーションが成功したことを確認するには、`aws servicediscovery get-operation` を実行できます。詳細については、「[get-operation](#)」を参照してください。

詳細については、AWS 「Cloud Map デベロッパーガイド」の「[サービスインスタンスの登録解除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterInstance](#)」の「`aws servicediscovery deregister-instance`」を参照してください。

discover-instances

次のコード例は、`discover-instances` を使用する方法を示しています。

AWS CLI

登録済みインスタンスを検出するには

次の`discover-instances`例では、登録されたインスタンスを検出します。

```
aws servicediscovery discover-instances \
  --namespace-name example.com \
  --service-name myservice \
  --max-results 10 \
  --health-status ALL
```

出力:

```
{
  "Instances": [
    {
      "InstanceId": "myservice-53",
      "NamespaceName": "example.com",
      "ServiceName": "myservice",
      "HealthStatus": "UNKNOWN",
    }
  ]
}
```

```
    "Attributes": {
      "AWS_INSTANCE_IPV4": "172.2.1.3",
      "AWS_INSTANCE_PORT": "808"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DiscoverInstances](#)」の「」を参照してください。

get-operation

次のコード例は、get-operation を使用する方法を示しています。

AWS CLI

オペレーションの結果を取得するには

次のget-operation例では、オペレーションの結果を取得します。

```
aws servicediscovery get-operation \
  --operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd
```

出力:

```
{
  "Operation": {
    "Id": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd",
    "Type": "CREATE_NAMESPACE",
    "Status": "SUCCESS",
    "CreateDate": 1587055860.121,
    "UpdateDate": 1587055900.469,
    "Targets": {
      "NAMESPACE": "ns-ylexjili4cdxy3xm"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOperation](#)」の「」を参照してください。

list-instances

次の例は、list-instances を使用する方法を説明しています。

AWS CLI

サービスインスタンスを一覧表示するには

次のlist-instances例では、サービスインスタンスを一覧表示します。

```
aws servicediscovery list-instances \  
  --service-id srv-qzpwvt2tfqcegapy
```

出力:

```
{  
  "Instances": [  
    {  
      "Id": "i-06bdabbae60f65a4e",  
      "Attributes": {  
        "AWS_INSTANCE_IPV4": "172.2.1.3",  
        "AWS_INSTANCE_PORT": "808"  
      }  
    }  
  ]  
}
```

詳細については、AWS「[Cloud Map デベロッパーガイド](#)」の「[サービスインスタンスのリストの表示](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスListInstances](#)」の「」を参照してください。

list-namespaces

次の例は、list-namespaces を使用する方法を説明しています。

AWS CLI

名前空間を一覧表示するには

次のlist-namespaces例では、名前空間を一覧表示します。

aws servicediscovery list-namespaces

出力:

```
{
  "Namespaces": [
    {
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-a3ccy2e7e3a7rile",
      "CreateDate": 1585354387.357,
      "Id": "ns-a3ccy2e7e3a7rile",
      "Name": "local",
      "Properties": {
        "DnsProperties": {
          "HostedZoneId": "Z06752353VBUDTC32S84S"
        },
        "HttpProperties": {
          "HttpName": "local"
        }
      },
      "Type": "DNS_PRIVATE"
    },
    {
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-pocfyjtrsmwtvcxx",
      "CreateDate": 1586468974.698,
      "Description": "My second namespace",
      "Id": "ns-pocfyjtrsmwtvcxx",
      "Name": "My-second-namespace",
      "Properties": {
        "DnsProperties": {},
        "HttpProperties": {
          "HttpName": "My-second-namespace"
        }
      },
      "Type": "HTTP"
    },
    {
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-ylexjili4cdxy3xm",
      "CreateDate": 1587055896.798,
      "Id": "ns-ylexjili4cdxy3xm",
      "Name": "example.com",
```

```

        "Properties": {
            "DnsProperties": {
                "HostedZoneId": "Z09983722P0QME1B3KC8I"
            },
            "HttpProperties": {
                "HttpName": "example.com"
            }
        },
        "Type": "DNS_PRIVATE"
    }
]
}

```

詳細については、AWS 「Cloud Map デベロッパーガイド」の [「名前空間のリストの表示」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListNamespaces](#)」の「」を参照してください。

list-services

次の例は、list-services を使用する方法を説明しています。

AWS CLI

サービスを一覧表示するには

次のlist-services例では、サービスの一覧を示しています。

```
aws servicediscovery list-services
```

出力:

```

{
  "Services": [
    {
      "Id": "srv-p5zdwlg5uvvzjita",
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:service/srv-p5zdwlg5uvvzjita",
      "Name": "myservice",
      "DnsConfig": {
        "RoutingPolicy": "MULTIVALUE",

```

```
    "DnsRecords": [
      {
        "Type": "A",
        "TTL": 60
      }
    ],
    "CreateDate": 1587081768.334
  }
]
```

詳細については、AWS「Cloud Map デベロッパーガイド」の「[サービスのリストの表示](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListServices](#)」の「」を参照してください。

register-instance

次の例は、register-instance を使用する方法を説明しています。

AWS CLI

サービスインスタンスを登録するには

次のregister-instance例では、サービスインスタンスを登録します。

```
aws servicediscovery register-instance \
  --service-id srv-p5zdwlg5uvvzjita \
  --instance-id myservice-53 \
  --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

出力:

```
{
  "OperationId": "4yejorelbukcjpnr6t1mrghsjwpngf4-k95yg2u7"
}
```

オペレーションが成功したことを確認するには、`get-operation` を実行できます。詳細については、「[get-operation](#)」を参照してください。

詳細については、AWS 「Cloud Map デベロッパーガイド」の「[インスタンスの登録](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterInstance](#)」の「」を参照してください。

AWS Cloud9 を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Cloud9。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-environment-ec2

次の例は、create-environment-ec2 を使用する方法を説明しています。

AWS CLI

AWS Cloud9 EC2開発環境を作成するには

次のcreate-environment-ec2例では、指定された設定で AWS Cloud9 開発環境を作成し、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを起動し、インスタンスから環境に接続します。

```
aws cloud9 create-environment-ec2 \  
  --name my-demo-env \  
  --description "My demonstration development environment." \  
  --instance-type t2.micro --image-id amazonlinux-2023-x86_64 \  
  --
```

```
--subnet-id subnet-1fab8aEX \  
--automatic-stop-time-minutes 60 \  
--owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

出力:

```
{  
  "environmentId": "8a34f51ce1e04a08882f1e811bd706EX"  
}
```

詳細については、「Cloud9 ユーザーガイド」の[EC2「環境の作成」](#)を参照してください。AWS Cloud9

- API 詳細については、「コマンドリファレンス」の[CreateEnvironmentEc「2」](#)を参照してください。AWS CLI

create-environment-membership

次の例は、create-environment-membership を使用する方法を説明しています。

AWS CLI

AWS Cloud9 開発環境に環境メンバーを追加するには

この例では、指定された環境メンバーを指定された AWS Cloud9 開発環境に追加します。

コマンド:

```
aws cloud9 create-environment-membership --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/  
AnotherDemoUser --permissions read-write
```

出力:

```
{  
  "membership": {  
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",  
    "userId": "AIDAJ3LOROMOUXTBSU6EX",  
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",  
    "permissions": "read-write"  
  }
```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEnvironmentMembership](#)」の「」を参照してください。

delete-environment-membership

次の例は、delete-environment-membership を使用する方法を説明しています。

AWS CLI

AWS Cloud9 開発環境から環境メンバーを削除するには

この例では、指定された AWS Cloud9 開発環境から指定された環境メンバーを削除します。

コマンド:

```
aws cloud9 delete-environment-membership --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/  
AnotherDemoUser
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEnvironmentMembership](#)」の「」を参照してください。

delete-environment

次の例は、delete-environment を使用する方法を説明しています。

AWS CLI

AWS Cloud9 開発環境を削除するには

この例では、指定された AWS Cloud9 開発環境を削除します。Amazon EC2 インスタンスが環境に接続されている場合、はインスタンスも終了します。

コマンド:

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEnvironment](#)」の「」を参照してください。

describe-environment-memberships

次の例は、describe-environment-memberships を使用する方法を説明しています。

AWS CLI

AWS Cloud9 開発環境の環境メンバーに関する情報を取得するには

この例では、指定された AWS Cloud9 開発環境の環境メンバーに関する情報を取得します。

コマンド:

```
aws cloud9 describe-environment-memberships --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

出力:

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJ3LOROMOUCTBSUGEX",
      "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
      "permissions": "read-write"
    },
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
```

```
    "permissions": "owner"
  }
]
}
```

AWS Cloud9 開発環境の所有者に関する情報を取得するには

この例では、指定された AWS Cloud9 開発環境の所有者に関する情報を取得します。

コマンド:

```
aws cloud9 describe-environment-memberships --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --permissions owner
```

出力:

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

複数の AWS Cloud9 開発環境の環境メンバーに関する情報を取得するには

この例では、複数の AWS Cloud9 開発環境の指定された環境メンバーに関する情報を取得します。

コマンド:

```
aws cloud9 describe-environment-memberships --user-  
arn arn:aws:iam::123456789012:user/MyDemoUser
```

出力:

```
{
```

```
"memberships": [  
  {  
    "environmentId": "10a75714bd494714929e7f5ec4125aEX",  
    "lastAccess": 1516213427.0,  
    "userId": "AIDAJNUEDQAQWFELJDLEX",  
    "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",  
    "permissions": "owner"  
  },  
  {  
    "environmentId": "1980b80e5f584920801c09086667f0EX",  
    "lastAccess": 1516144884.0,  
    "userId": "AIDAJNUEDQAQWFELJDLEX",  
    "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",  
    "permissions": "owner"  
  }  
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEnvironmentMemberships](#)」の「」を参照してください。

describe-environment-status

次の例は、describe-environment-status を使用する方法を説明しています。

AWS CLI

AWS Cloud9 開発環境のステータス情報を取得するには

この例では、指定された AWS Cloud9 開発環境のステータス情報を取得します。

コマンド:

```
aws cloud9 describe-environment-status --environment-  
id 685f892f431b45c2b28cb69eadcdb0EX
```

出力:

```
{  
  "status": "ready",  
  "message": "Environment is ready to use"
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEnvironmentStatus](#)」の「」を参照してください。

describe-environments

次の例は、describe-environments を使用する方法を説明しています。

AWS CLI

AWS Cloud9 開発環境に関する情報を取得するには

この例では、指定された AWS Cloud9 開発環境に関する情報を取得します。

コマンド:

```
aws cloud9 describe-environments --environment-ids 685f892f431b45c2b28cb69eadcdb0EX 349c86d4579e4e7298d500ff57a6b2EX
```

出力:

```
{
  "environments": [
    {
      "id": "685f892f431b45c2b28cb69eadcdb0EX",
      "name": "my-demo-ec2-env",
      "description": "Created from CodeStar.",
      "type": "ec2",
      "arn": "arn:aws:cloud9:us-east-1:123456789012:environment:685f892f431b45c2b28cb69eadcdb0EX",
      "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "lifecycle": {
        "status": "CREATED"
      }
    },
    {
      "id": "349c86d4579e4e7298d500ff57a6b2EX",
      "name": "my-demo-ssh-env",
      "description": "",
      "type": "ssh",
      "arn": "arn:aws:cloud9:us-east-1:123456789012:environment:349c86d4579e4e7298d500ff57a6b2EX",

```

```
    "ownerArn": "arn:aws:iam::123456789012:user/MyDemoUser",
    "lifecycle": {
      "status": "CREATED"
    }
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEnvironments](#)」の「」を参照してください。

list-environments

次の例は、list-environments を使用する方法を説明しています。

AWS CLI

使用可能な AWS Cloud9 開発環境識別子のリストを取得するには

この例では、使用可能な AWS Cloud9 開発環境識別子のリストを取得します。

コマンド:

```
aws cloud9 list-environments
```

出力:

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEnvironments](#)」の「」を参照してください。

update-environment-membership

次の例は、update-environment-membership を使用する方法を説明しています。

AWS CLI

AWS Cloud9 開発環境の既存の環境メンバーの設定を変更するには

この例では、指定された AWS Cloud9 開発環境の指定された既存の環境メンバーの設定を変更します。

コマンド:

```
aws cloud9 update-environment-membership --environment-id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/AnotherDemoUser --permissions read-only
```

出力:

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUCTBSUGEX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-only"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateEnvironmentMembership](#)」の「」を参照してください。

update-environment

次の例は、update-environment を使用する方法を説明しています。

AWS CLI

既存の AWS Cloud9 開発環境の設定を変更するには

この例では、指定された既存の AWS Cloud9 開発環境の指定された設定を変更します。

コマンド:

```
aws cloud9 update-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX --name my-changed-demo-env --description "My changed demonstration development environment."
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateEnvironment](#)」の「」を参照してください。

AWS CloudFormation を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS CloudFormation。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

activate-type

次のコード例は、activate-type を使用する方法を示しています。

AWS CLI

タイプをアクティブ化するには

次のactivate-type例では、パブリックサードパーティー拡張機能をアクティブ化し、スタックテンプレートで使用できるようにします。

```
aws cloudformation activate-type \  
  --region us-west-2 \  
  --type RESOURCE \  
  --
```

```
--type-name Example::Test::1234567890abcdef0 \  
--type-name-alias Example::Test::Alias
```

出力:

```
{  
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Example-  
Test-Alias"  
}
```

詳細については、AWS CloudFormation 「[ユーザーガイド](#)」の [AWS CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス ActivateType](#)」の「」を参照してください。

batch-describe-type-configurations

次のコード例は、batch-describe-type-configurations を使用する方法を示しています。

AWS CLI

タイプ設定をバッチ記述するには

次のbatch-describe-type-configurations例では、タイプのデータを設定します。

```
aws cloudformation batch-describe-type-configurations \  
  --region us-west-2 \  
  --type-configuration-identifiers TypeArn="arn:aws:cloudformation:us-  
west-2:123456789012:type/resource/Example-Test-  
Type,TypeConfigurationAlias=MyConfiguration"
```

出力:

```
{  
  "Errors": [],  
  "UnprocessedTypeConfigurations": [],  
  "TypeConfigurations": [  
    {  
      "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/  
Example-Test-Type",
```

```
        "Alias": "MyConfiguration",
        "Configuration": "{\n          \"Example\": {\n            \"ApiKey\":\n\n            \"examplekey\", \n            \"ApplicationKey\": \"examplekey1\", \n            \"ApiURL\": \"exampleurl\"\n          }\n        },\n        \"LastUpdated\": \"2021-10-01T15:25:46.210000+00:00\",
        \"TypeArn\": \"arn:aws:cloudformation:us-east-1:123456789012:type/resource/\nExample-Test-Type\"\n      }\n    ]\n  }\n}
```

詳細については、AWS CloudFormation [「ユーザーガイド」の AWS CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスBatchDescribeTypeConfigurations」](#) の「」を参照してください。

cancel-update-stack

次の例は、cancel-update-stack を使用する方法を説明しています。

AWS CLI

進行中のスタック更新をキャンセルする

以下の cancel-update-stack コマンドは、myteststack スタックで行われているスタック更新をキャンセルします。

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- API 詳細については、AWS CLI [「コマンドリファレンスCancelUpdateStack」](#) の「」を参照してください。

continue-update-rollback

次の例は、continue-update-rollback を使用する方法を説明しています。

AWS CLI

更新のロールバックを再試行する

以下の `continue-update-rollback` 例は、前回失敗したスタック更新からのロールバック操作を再開します。

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [ContinueUpdateRollback](#)」の「」を参照してください。

create-change-set

次のコード例は、`create-change-set` を使用する方法を示しています。

AWS CLI

変更セットを作成するには

次の `create-change-set` 例では、`CAPABILITY_IAM` 機能を持つ変更セットを作成します。ファイルは、IAM リソースを含むスタックを定義する現在のフォルダ内の AWS CloudFormation テンプレート `template.yaml` です。

```
aws cloudformation create-change-set \  
  --stack-name my-application \  
  --change-set-name my-change-set \  
  --template-body file://template.yaml \  
  --capabilities CAPABILITY_IAM
```

出力:

```
{  
  "Id": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/  
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",  
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-application/  
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateChangeSet](#)」の「」を参照してください。

create-stack-instances

次の例は、create-stack-instances を使用する方法を説明しています。

AWS CLI

スタックインスタンスを作成するには

次のcreate-stack-instances例では、2つのアカウントと4つのリージョンにスタックセットのインスタンスを作成します。耐障害性の設定により、一部のスタックを作成できない場合でも、すべてのアカウントとリージョンで更新が試行されます。

```
aws cloudformation create-stack-instances \  
  --stack-set-name my-stack-set \  
  --accounts 123456789012 223456789012 \  
  --regions us-east-1 us-east-2 us-west-1 us-west-2 \  
  --operation-preferences FailureToleranceCount=7
```

出力:

```
{  
  "OperationId": "d7995c31-83c2-xmpl-a3d4-e9ca2811563f"  
}
```

スタックセットを作成するには、create-stack-set コマンドを使用します。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStackInstances](#)」の「」を参照してください。

create-stack-set

次のコード例は、create-stack-set を使用する方法を示しています。

AWS CLI

スタックセットを作成するには

次のcreate-stack-set例では、指定されたYAMLファイルテンプレートを使用してスタックセットを作成します。template.yamlは、スタックを定義する現在のフォルダ内のAWS CloudFormation テンプレートです。

```
aws cloudformation create-stack-set \  

```

```
--stack-set-name my-stack-set \  
--template-body file://template.yaml \  
--description "SNS topic"
```

出力:

```
{  
  "StackSetId": "my-stack-set:8d0f160b-d157-xmpl-a8e6-c0ce8e5d8cc1"  
}
```

スタックセットにスタックインスタンスを追加するには、`create-stack-instances` コマンドを使用します。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateStackSet](#)」の「」を参照してください。

create-stack

次のコード例は、`create-stack` を使用する方法を示しています。

AWS CLI

AWS CloudFormation スタックを作成するには

以下の `create-stacks` コマンドは、`sampletemplate.json` テンプレートを使用して `myteststack` という名前のスタックを作成します。

```
aws cloudformation create-stack --stack-name myteststack --template-body file://  
sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey  
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

出力:

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

詳細については、AWS CloudFormation 「ユーザーガイド」の「スタック」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStack](#)」の「」を参照してください。

deactivate-type

次の例は、deactivate-type を使用する方法を説明しています。

AWS CLI

タイプを非アクティブ化するには

次のdeactivate-type例では、このアカウントとリージョンで以前にアクティブ化されたパブリック拡張機能を非アクティブ化します。

```
aws cloudformation deactivate-type \  
  --region us-west-2 \  
  --type MODULE \  
  --type-name Example::Test::Type::MODULE
```

このコマンドでは何も出力されません。

詳細については、AWS CloudFormation 「[ユーザーガイド](#)」の [AWS CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeactivateType](#)」の「」を参照してください。

delete-change-set

次の例は、delete-change-set を使用する方法を説明しています。

AWS CLI

変更セットを削除するには

次のdelete-change-set例では、変更セット名とスタック名を指定して変更セットを削除します。

```
aws cloudformation delete-change-set \  
  --stack-name my-stack \  
  --change-set-name my-change-set
```



```
--change-set-name my-change-set
```

このコマンドでは何も出力されません。

次のdelete-change-set例では、変更セットARN全体を指定して変更セットを削除します。

```
aws cloudformation delete-change-set \  
  --change-set-name arn:aws:cloudformation:us-east-2:123456789012:changeSet/my-  
change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteChangeSet](#)」の「」を参照してください。

delete-stack-instances

次の例は、delete-stack-instances を使用する方法を説明しています。

AWS CLI

スタックインスタンスを削除するには

次のdelete-stack-instances例では、2つのリージョンの2つのアカウントに設定されたスタックのインスタンスを削除し、スタックを終了します。

```
aws cloudformation delete-stack-instances \  
  --stack-set-name my-stack-set \  
  --accounts 123456789012 567890123456 \  
  --regions us-east-1 us-west-1 \  
  --no-retain-stacks
```

出力:

```
{  
  "OperationId": "ad49f10c-fd1d-413f-a20a-8de6e2fa8f27"  
}
```

空のスタックセットを削除するには、delete-stack-set コマンドを使用します。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStackInstances](#)」の「」を参照してください。

delete-stack-set

次の例は、delete-stack-set を使用する方法を説明しています。

AWS CLI

スタックセットを削除するには

次のコマンドは、指定された空のスタックセットを削除します。スタックセットは空である必要があります。

```
aws cloudformation delete-stack-set \  
  --stack-set-name my-stack-set
```

このコマンドでは何も出力されません。

スタックセットからインスタンスを削除するには、delete-stack-instances コマンドを使用します。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStackSet](#)」の「」を参照してください。

delete-stack

次の例は、delete-stack を使用する方法を説明しています。

AWS CLI

スタックを削除する

以下の delete-stack 例は、指定されたスタックを削除します。

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStack](#)」の「」を参照してください。

deploy

次の例は、deploy を使用する方法を説明しています。

AWS CLI

次のコマンドは、`template.json` という名前のテンプレートを `my-new-stack` という名前のスタックにデプロイします。

```
aws cloudformation deploy --template-file /path_to_template/template.json
--stack-name my-new-stack --parameter-overrides Key1=Value1 Key2=Value2 --
tags Key1=Value1 Key2=Value2
```

- API 詳細については、「コマンドリファレンス」の[「デプロイ」](#)を参照してください。AWS CLI

deregister-type

次の例は、deregister-type を使用する方法を説明しています。

AWS CLI

タイプバージョンの登録を解除するには

次のderegister-type例では、指定されたタイプバージョンを CloudFormation レジストリでのアクティブな使用から削除し、CloudFormation オペレーションで使用できなくなります。

```
aws cloudformation deregister-type \
--type RESOURCE \
--type-name My::Logs::LogGroup \
--version-id 00000002
```

このコマンドでは何も出力されません。

詳細については、AWS CloudFormation 「ユーザーガイド」の[CloudFormation 「レジストリの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterType](#)」の「」を参照してください。

describe-account-limits

次のコード例は、describe-account-limits を使用する方法を示しています。

AWS CLI

アカウント制限に関する情報を取得するには

次のコマンドは、現在のアカウントのリージョン制限のリストを取得します。

```
aws cloudformation describe-account-limits
```

出力:

```
{
  "AccountLimits": [
    {
      "Name": "StackLimit",
      "Value": 200
    },
    {
      "Name": "StackOutputsLimit",
      "Value": 60
    },
    {
      "Name": "ConcurrentResourcesLimit",
      "Value": 2500
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountLimits](#)」の「」を参照してください。

describe-change-set

次のコード例は、describe-change-set を使用する方法を示しています。

AWS CLI

変更セットに関する情報を取得するには

次のdescribe-change-set例では、変更セット名とスタック名で指定された変更セットの詳細を表示します。

```
aws cloudformation describe-change-set \  
  --change-set-name my-change-set \  
  --stack-name my-stack
```

次のdescribe-change-set例は、変更セットARNのフルで指定された変更セットの詳細を表示します。

```
aws cloudformation describe-change-set \  
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

出力:

```
{  
  "Changes": [  
    {  
      "Type": "Resource",  
      "ResourceChange": {  
        "Action": "Modify",  
        "LogicalResourceId": "function",  
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",  
        "ResourceType": "AWS::Lambda::Function",  
        "Replacement": "False",  
        "Scope": [  
          "Properties"  
        ],  
        "Details": [  
          {  
            "Target": {  
              "Attribute": "Properties",  
              "Name": "Timeout",  
              "RequiresRecreation": "Never"  
            },  
            "Evaluation": "Static",
```

```

        "ChangeSource": "DirectModification"
      }
    ]
  }
},
"ChangeSetName": "my-change-set",
"ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
"StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
"StackName": "my-stack",
"Description": null,
"Parameters": null,
"CreationTime": "2019-10-02T05:20:56.651Z",
"ExecutionStatus": "AVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": null,
"NotificationARNs": [],
"RollbackConfiguration": {},
"Capabilities": [
  "CAPABILITY_IAM"
],
"Tags": null
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeChangeSet](#)」の「」を参照してください。

describe-publisher

次のコード例は、describe-publisher を使用する方法を示しています。

AWS CLI

パブリッシャーを記述するには

次のdescribe-publisher例では、パブリッシャーの情報を設定します。

```

aws cloudformation describe-publisher \
  --region us-west-2 \
  --publisher-id 000q6TfUovXsEMmgKowxDZLLwqr2QUsh

```

出力:

```
{
  "PublisherId": "000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c",
  "PublisherStatus": "VERIFIED",
  "IdentityProvider": "AWS_Marketplace",
  "PublisherProfile": "https://aws.amazon.com/marketplace/seller-profile?id=2c5dc1f0-17cd-4259-8e46-822a83gdtegd"
}
```

詳細については、AWS CloudFormation [「ユーザーガイド」](#) の [AWS CloudFormation「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスDescribePublisher」](#) の「」を参照してください。

describe-stack-drift-detection-status

次のコード例は、describe-stack-drift-detection-status を使用する方法を示しています。

AWS CLI

ドリフト検出オペレーションのステータスを確認するには

次のdescribe-stack-drift-detection-status例では、ドリフト検出オペレーションのステータスを表示します。detect-stack-drift コマンドを実行する ID で を取得します。

```
aws cloudformation describe-stack-drift-detection-status \
  --stack-drift-detection-id 1a229160-e4d9-xmpl-ab67-0a4f93df83d4
```

出力:

```
{
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
  "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4",
  "StackDriftStatus": "DRIFTED",
  "DetectionStatus": "DETECTION_COMPLETE",
  "DriftedStackResourceCount": 1,
  "Timestamp": "2019-10-02T05:54:30.902Z"
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStackDriftDetectionStatus](#)」の「」を参照してください。

describe-stack-events

次のコード例は、describe-stack-events を使用する方法を示しています。

AWS CLI

スタックイベントを記述する

以下の describe-stack-events 例は、指定されたスタックの最新イベントを 2 件表示します。

```
aws cloudformation describe-stack-events \  
  --stack-name my-stack \  
  --max-items 2  
  
{  
  "StackEvents": [  
    {  
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-  
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
      "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",  
      "StackName": "my-stack",  
      "LogicalResourceId": "my-stack",  
      "PhysicalResourceId": "arn:aws:cloudformation:us-  
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
      "ResourceType": "AWS::CloudFormation::Stack",  
      "Timestamp": "2019-10-02T05:34:29.556Z",  
      "ResourceStatus": "UPDATE_COMPLETE"  
    },  
    {  
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-  
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
      "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",  
      "StackName": "my-stack",  
      "LogicalResourceId": "my-stack",  
      "PhysicalResourceId": "arn:aws:cloudformation:us-  
west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
      "ResourceType": "AWS::CloudFormation::Stack",
```



```

    "Timestamp": "2019-10-02T05:34:29.127Z",
    "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
  },
  ],
  "NextToken": "eyJ0ZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStackEvents](#)」の「」を参照してください。

describe-stack-instance

次の例は、describe-stack-instance を使用する方法を説明しています。

AWS CLI

スタックインスタンスを記述するには

次のコマンドは、指定されたアカウントとリージョンで指定されたスタックセットのインスタンスについて説明します。スタックセットは現在のリージョンとアカウントにあり、インスタンスはアカウントの us-west-2 リージョンにあります 123456789012。

```

aws cloudformation describe-stack-instance \
  --stack-set-name my-stack-set \
  --stack-instance-account 123456789012 \
  --stack-instance-region us-west-2

```

出力:

```

{
  "StackInstance": {
    "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
    "Region": "us-west-2",
    "Account": "123456789012",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/4287f9a0-e615-xmpl-894a-12b31d3117be",
    "ParameterOverrides": [],
    "Status": "OUTDATED",
    "StatusReason": "ResourceLogicalId:ConfigBucket, Resource Type:AWS::S3::Bucket, ResourceStatusReason:You have attempted to create

```

```

more buckets than allowed (Service: Amazon S3; Status Code: 400; Error Code:
TooManyBuckets; Request ID: F7F21CXMPL580224; S3 Extended Request ID: egd/
Fdt89BXMPLYiqbMNljVvk55Yqqvi3NYW2nKLUVWhUGEhNfCmZdyj9671hriaG/dWMobS040o=)."
    }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStackInstance](#)」の「」を参照してください。

describe-stack-resource-drifts

次の例は、describe-stack-resource-drifts を使用する方法を説明しています。

AWS CLI

スタック定義からドリフトしたリソースに関する情報を取得するには

次のコマンドは、指定されたスタックのドリフトされたリソースに関する情報を表示します。ドリフト検出を開始するには、detect-stack-drift コマンドを使用します。

```

aws cloudformation describe-stack-resource-drifts \
  --stack-name my-stack

```

出力には、変更された AWS Lambda 関数が表示されます out-of-band。

```

{
  "StackResourceDrifts": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
      "LogicalResourceId": "function",
      "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
      "ResourceType": "AWS::Lambda::Function",
      "ExpectedProperties": "{\"Description\":\"Write a file to S3.\",
\\\"Environment\\\":{\\\"Variables\\\":{\\\"bucket\\\":\\\"my-stack-bucket-1vc62xmplgguf\\\"}}},\\\"Handler\\\":\\\"index.handler\\\",\\\"MemorySize\\\":128,\\\"Role\\\":\\\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLE0M9E\\\",\\\"Runtime\\\":\\\"nodejs10.x\\\",\\\"Tags\\\":[{\\\"Key\\\":\\\"lambda:createdBy\\\",\\\"Value\\\":\\\"SAM\\\"}],\\\"Timeout\\\":900,\\\"TracingConfig\\\":{\\\"Mode\\\":\\\"Active\\\"}}\",
      "ActualProperties": "{\"Description\":\"Write a file to S3.\",
\\\"Environment\\\":{\\\"Variables\\\":{\\\"bucket\\\":\\\"my-stack-bucket-1vc62xmplgguf\\\"}}},\\\"Handler\\\":\\\"index.handler\\\",\\\"MemorySize\\\":256,\\\"Role\\\":

```

```

\arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\", \"Runtime\":
\nodejs10.x\", \"Tags\": [{\"Key\": \"lambda:createdBy\", \"Value\": \"SAM\"}], \"Timeout
\":22, \"TracingConfig\": {\"Mode\": \"Active\"}},
  \"PropertyDifferences\": [
    {
      \"PropertyPath\": \"/MemorySize\",
      \"ExpectedValue\": \"128\",
      \"ActualValue\": \"256\",
      \"DifferenceType\": \"NOT_EQUAL\"
    },
    {
      \"PropertyPath\": \"/Timeout\",
      \"ExpectedValue\": \"900\",
      \"ActualValue\": \"22\",
      \"DifferenceType\": \"NOT_EQUAL\"
    }
  ],
  \"StackResourceDriftStatus\": \"MODIFIED\",
  \"Timestamp\": \"2019-10-02T05:54:44.064Z\"
}
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStackResourceDrifts](#)」の「」を参照してください。

describe-stack-resource

次の例は、describe-stack-resource を使用する方法を説明しています。

AWS CLI

リソースに関する情報を取得する

以下の describe-stack-resource 例は、指定されたスタック内にある MyFunction という名前のリソースに関する詳細情報を表示します。

```

aws cloudformation describe-stack-resource \
  --stack-name MyStack \
  --logical-resource-id MyFunction

```

出力:

```
{
  "StackResourceDetail": {
    "StackName": "MyStack",
    "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "MyFunction",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
    "ResourceStatus": "UPDATE_COMPLETE",
    "Metadata": "{}",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStackResource](#)」の「」を参照してください。

describe-stack-resources

次のコード例は、describe-stack-resources を使用する方法を示しています。

AWS CLI

リソースに関する情報を取得する

次の describe-stack-resources 例は、指定されたスタック内にあるリソースの詳細情報を表示します。

```
aws cloudformation describe-stack-resources \
  --stack-name my-stack
```

出力:

```
{
  "StackResources": [
    {
      "StackName": "my-stack",
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```

    "LogicalResourceId": "bucket",
    "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
    "ResourceType": "AWS::S3::Bucket",
    "Timestamp": "2019-10-02T04:34:11.345Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  },
  {
    "StackName": "my-stack",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "function",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "Timestamp": "2019-10-02T05:34:27.989Z",
    "ResourceStatus": "UPDATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  },
  {
    "StackName": "my-stack",
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "functionRole",
    "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",
    "ResourceType": "AWS::IAM::Role",
    "Timestamp": "2019-10-02T04:34:06.350Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStackResources](#)」の「」を参照してください。

describe-stack-set-operation

次のコード例は、describe-stack-set-operation を使用する方法を示しています。

AWS CLI

スタックセットオペレーションに関する情報を取得するには

次の describe-stack-set-operation` 例は、指定されたスタックセットの更新オペレーションの詳細を表示します。

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name enable-config \  
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

出力:

```
{  
  "StackSetOperation": {  
    "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",  
    "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",  
    "Action": "UPDATE",  
    "Status": "SUCCEEDED",  
    "OperationPreferences": {  
      "RegionOrder": [  
        "us-east-1",  
        "us-west-2",  
        "eu-west-1",  
        "us-west-1"  
      ],  
      "FailureToleranceCount": 7,  
      "MaxConcurrentCount": 2  
    },  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
AWSCloudFormationStackSetAdministrationRole",  
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",  
    "CreationTimestamp": "2019-10-03T16:28:44.377Z",  
    "EndTimestamp": "2019-10-03T16:42:08.607Z"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStackSetOperation](#)」の「」を参照してください。

describe-stack-set

次の例は、describe-stack-set を使用する方法を説明しています。

AWS CLI

スタックセットに関する情報を取得するには

次の describe-stack-set の例は、指定されたスタックセットの詳細を表示します。

```
aws cloudformation describe-stack-set \  
  --stack-set-name my-stack-set
```

出力:

```
{  
  "StackSet": {  
    "StackSetName": "my-stack-set",  
    "StackSetId": "my-stack-set:296a3360-xmpl-40af-be78-9341e95bf743",  
    "Description": "Create an Amazon SNS topic",  
    "Status": "ACTIVE",  
    "TemplateBody": "AWSTemplateFormatVersion: '2010-09-09'\nDescription: An AWS  
SNS topic\nResources:\n  topic:\n    Type: AWS::SNS::Topic",  
    "Parameters": [],  
    "Capabilities": [],  
    "Tags": [],  
    "StackSetARN": "arn:aws:cloudformation:us-west-2:123456789012:stackset/  
enable-config:296a3360-xmpl-40af-be78-9341e95bf743",  
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/  
AWSCloudFormationStackSetAdministrationRole",  
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStackSet](#)」の「」を参照してください。

describe-stacks

次のコード例は、describe-stacks を使用する方法を示しています。

AWS CLI

AWS CloudFormation スタックを記述するには

以下の describe-stacks 例は、myteststack スタックの概要情報を表示します。

```
aws cloudformation describe-stacks --stack-name myteststack
```

出力:

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
      "Tags": [],
      "Outputs": [
        {
          "Description": "Name of S3 bucket to hold website content",
          "OutputKey": "BucketName",
          "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
        }
      ],
      "StackStatusReason": null,
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "CREATE_COMPLETE",
      "DisableRollback": false
    }
  ]
}
```

詳細については、AWS CloudFormation 「ユーザーガイド」の「スタック」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStacks](#)」の「」を参照してください。

describe-type-registration

次の例は、describe-type-registration を使用する方法を説明しています。

AWS CLI

タイプ登録情報を表示するには

次のdescribe-type-registration例では、タイプのステータス、タイプ、バージョンなど、指定されたタイプ登録に関する情報を表示します。

```
aws cloudformation describe-type-registration \  
  --registration-token a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "ProgressStatus": "COMPLETE",  
  "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup",  
  "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",  
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup/00000001"  
}
```

詳細については、AWS CloudFormation「[ユーザーガイド](#)」の [CloudFormation「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeTypeRegistration](#)」の「」を参照してください。

describe-type

次の例は、describe-type を使用する方法を説明しています。

AWS CLI

タイプ情報を表示するには

次のdescribe-type例では、指定されたタイプの情報を表示します。

```
aws cloudformation describe-type \  
  --type-name My::Logs::LogGroup \  
  --
```

```
--type RESOURCE
```

出力:

```
{
  "SourceUrl": "https://github.com/aws-cloudformation/aws-cloudformation-resource-providers-logs.git",
  "Description": "Customized resource derived from AWS::Logs::LogGroup",
  "TimeCreated": "2019-12-03T23:29:33.321Z",
  "Visibility": "PRIVATE",
  "TypeName": "My::Logs::LogGroup",
  "LastUpdated": "2019-12-03T23:29:33.321Z",
  "DeprecatedStatus": "LIVE",
  "ProvisioningType": "FULLY_MUTABLE",
  "Type": "RESOURCE",
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Logs-LogGroup/00000001",
  "Schema": "[details omitted]"
}
```

詳細については、AWS CloudFormation「[ユーザーガイド](#)」の [CloudFormation「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス DescribeType](#)」の「」を参照してください。

detect-stack-drift

次の例は、detect-stack-drift を使用する方法を説明しています。

AWS CLI

ドリフトしたリソースを検出するには

次のdetect-stack-drift例では、指定されたスタックのドリフト検出を開始します。

```
aws cloudformation detect-stack-drift \  
  --stack-name my-stack
```

出力:

```
{
```

```
"StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4"
}
```

その後、`describe-stack-resource-drifts` コマンドでこの ID を使用してドリフトされたリソースを記述できます。

- API 詳細については、AWS CLI 「コマンドリファレンス [DetectStackDrift](#)」の「」を参照してください。

detect-stack-resource-drift

次の例は、`detect-stack-resource-drift` を使用する方法を説明しています。

AWS CLI

リソースのドリフトを検出するには

次の`detect-stack-resource-drift`例では、ドリフトMyStackの という名前のスタックMyFunctionで という名前のリソースをチェックします。

```
aws cloudformation detect-stack-resource-drift \
  --stack-name MyStack \
  --logical-resource-id MyFunction
```

出力には、変更された AWS Lambda 関数が表示されます out-of-band。

```
{
  "StackResourceDrift": {
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/MyStack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
    "LogicalResourceId": "MyFunction",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "ExpectedProperties": "{\"Description\":\"Write a file to S3.\",
  \"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf\"}},\"Handler\":\"index.handler\", \"MemorySize\":128, \"Role\":\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLE0M9E\", \"Runtime\":\"nodejs10.x\", \"Tags\":[{\"Key\":\"lambda:createdBy\", \"Value\":\"SAM\"}], \"Timeout\":900, \"TracingConfig\":{\"Mode\":\"Active\"}}",
    "ActualProperties": "{\"Description\":\"Write a file to S3.\", \"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf\"}}, \"Handler\":\"index.handler\", \"MemorySize\":256, \"Role\":\"arn:aws:iam::123456789012:role/
```

```

my-functionRole-HIZXMPLEOM9E\", \"Runtime\": \"nodejs10.x\", \"Tags\": [{\"Key\":
\"lambda:createdBy\", \"Value\": \"SAM\"}], \"Timeout\": 22, \"TracingConfig\": {\"Mode\":
\"Active\"}}\",
  \"PropertyDifferences\": [
    {
      \"PropertyPath\": \"/MemorySize\",
      \"ExpectedValue\": \"128\",
      \"ActualValue\": \"256\",
      \"DifferenceType\": \"NOT_EQUAL\"
    },
    {
      \"PropertyPath\": \"/Timeout\",
      \"ExpectedValue\": \"900\",
      \"ActualValue\": \"22\",
      \"DifferenceType\": \"NOT_EQUAL\"
    }
  ],
  \"StackResourceDriftStatus\": \"MODIFIED\",
  \"Timestamp\": \"2019-10-02T05:58:47.433Z\"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DetectStackResourceDrift](#)」の「」を参照してください。

detect-stack-set-drift

次のコード例は、detect-stack-set-drift を使用する方法を示しています。

AWS CLI

スタックセットおよび関連するすべてのスタックインスタンスのドリフトを検出するには

次のdetect-stack-set-drift例では、指定されたスタックセットに対して、そのスタックセットに関連付けられたすべてのスタックインスタンスを含むドリフト検出オペレーションを開始し、ドリフトオペレーションのステータスを追跡するために使用できるオペレーション ID を返します。

```

aws cloudformation detect-stack-set-drift \
  --stack-set-name stack-set-drift-example

```

出力:

```
{
  "OperationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

詳細については、AWS CloudFormation 「[ユーザーガイド](#)」の「[スタックセットで管理されていない設定変更を検出する](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDetectStackSetDrift](#)」の「」を参照してください。

estimate-template-cost

次の例は、estimate-template-cost を使用する方法を説明しています。

AWS CLI

テンプレートコストを見積もる

以下の estimate-template-cost 例は、現在のフォルダ内にある template.yaml という名前のテンプレートに関するコスト見積もりを生成します。

```
aws cloudformation estimate-template-cost \
  --template-body file://template.yaml
```

出力:

```
{
  "Url": "http://calculator.s3.amazonaws.com/calc5.html?
key=cloudformation/7870825a-xmpl-4def-92e7-c4f8dd360cca"
}
```

- API 詳細については、AWS CLI 「[コマンドリファレンスEstimateTemplateCost](#)」の「」を参照してください。

execute-change-set

次のコード例は、execute-change-set を使用する方法を示しています。

AWS CLI

変更セットを実行するには

次のexecute-change-set例では、変更セット名とスタック名で指定された変更セットを実行します。

```
aws cloudformation execute-change-set \  
  --change-set-name my-change-set \  
  --stack-name my-stack
```

次のexecute-change-set例では、変更セットARNのフルで指定された変更セットを実行します。

```
aws cloudformation execute-change-set \  
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ExecuteChangeSet](#)」の「」を参照してください。

get-stack-policy

次のコード例は、get-stack-policy を使用する方法を示しています。

AWS CLI

スタックポリシーを表示するには

次のget-stack-policy例では、指定されたスタックのスタックポリシーを表示します。スタックにポリシーをアタッチするには、set-stack-policy コマンドを使用します。

```
aws cloudformation get-stack-policy \  
  --stack-name my-stack
```

出力:

```
{  
  "StackPolicyBody": "{  
    \"Statement\" : [  
      {  
        \"Effect\" :  
        \"Allow\",  
        \"Action\" : \"Update:*\",  
        \"Principal\" : \"*\",  
        \"Resource\" : \"*\"  
      },  
      {  
        \"Effect\" : \"Deny\",  
        \"Action\" : \"Update:*\",  
        \"Principal\" : \"*\",  
        \"Resource\" :  
        \"LogicalResourceId/bucket\"  
      }  
    ]  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStackPolicy](#)」の「」を参照してください。

get-template-summary

次のコード例は、get-template-summary を使用する方法を示しています。

AWS CLI

テンプレートの概要を表示するには

次のコマンドは、指定されたテンプレートファイルのリソースとメタデータに関する概要情報を表示します。

```
aws cloudformation get-template-summary \  
  --template-body file://template.yaml
```

出力:

```
{  
  "Parameters": [],  
  "Description": "A VPC and subnets.",  
  "ResourceTypes": [  
    "AWS::EC2::VPC",  
    "AWS::EC2::Subnet",  
    "AWS::EC2::Subnet",  
    "AWS::EC2::RouteTable",  
    "AWS::EC2::VPCEndpoint",  
    "AWS::EC2::SubnetRouteTableAssociation",  
    "AWS::EC2::SubnetRouteTableAssociation",  
    "AWS::EC2::VPCEndpoint"  
  ],  
  "Version": "2010-09-09"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTemplateSummary](#)」の「」を参照してください。

get-template

次のコード例は、get-template を使用する方法を示しています。

AWS CLI

AWS CloudFormation スタックのテンプレート本文を表示するには

以下の `get-template` 例は、`myteststack` スタックのテンプレートを表示します。

```
aws cloudformation get-template --stack-name myteststack
```

出力:

```
{
  "TemplateBody": {
    "AWSTemplateFormatVersion": "2010-09-09",
    "Outputs": {
      "BucketName": {
        "Description": "Name of S3 bucket to hold website content",
        "Value": {
          "Ref": "S3Bucket"
        }
      }
    },
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
    "Resources": {
      "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
          "AccessControl": "PublicRead"
        }
      }
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTemplate](#)」の「」を参照してください。

list-change-sets

次のコード例は、`list-change-sets` を使用する方法を示しています。

AWS CLI

変更セットを一覧表示するには

次のlist-change-sets例では、指定されたスタックの保留中の変更セットのリストを表示します。

```
aws cloudformation list-change-sets \  
  --stack-name my-stack
```

出力:

```
{  
  "Summaries": [  
    {  
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-  
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
      "StackName": "my-stack",  
      "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/  
my-change-set/70160340-7914-xmpl-bcbf-128a1fa78b5d",  
      "ChangeSetName": "my-change-set",  
      "ExecutionStatus": "AVAILABLE",  
      "Status": "CREATE_COMPLETE",  
      "CreationTime": "2019-10-02T05:38:54.297Z"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListChangeSets](#)」の「」を参照してください。

list-exports

次の例は、list-exports を使用する方法を説明しています。

AWS CLI

エクスポートを一覧表示するには

次のlist-exports例では、現在のリージョンのスタックからのエクスポートのリストを表示します。

aws cloudformation list-exports

出力:

```
{
  "Exports": [
    {
      "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-a",
      "Value": "subnet-07b410xmplddcfa03"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-b",
      "Value": "subnet-075ed3xmpllebd2fb1"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-vpcid",
      "Value": "vpc-011d7xmpl1100e9841"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListExports](#)」の「」を参照してください。

list-imports

次の例は、list-imports を使用する方法を説明しています。

AWS CLI

インポートを一覧表示するには

次のlist-imports例では、指定されたエクスポートをインポートするスタックを一覧表示します。使用可能なエクスポートのリストを取得するには、list-exports コマンドを使用します。

```
aws cloudformation list-imports \  
  --export-name private-vpc-vpcid
```

出力:

```
{  
  "Imports": [  
    "my-database-stack"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListImports](#)」の「」を参照してください。

list-stack-instances

次の例は、list-stack-instances を使用する方法を説明しています。

AWS CLI

スタックのインスタスを一覧表示するには

次のlist-stack-instances例では、指定されたスタックセットから作成されたインスタスを一覧表示します。

```
aws cloudformation list-stack-instances \  
  --stack-set-name enable-config
```

出力例には、エラーが原因で更新に失敗したスタックに関する詳細が含まれます。

```
{  
  "Summaries": [  
    {  
      "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",  
      "Region": "us-west-2",  
      "Account": "123456789012",  
      "StackId": "arn:aws:cloudformation:ap-northeast-1:123456789012:stack/  
StackSet-enable-config-35a6ac50-d9f8-4084-86e4-7da34d5de4c4/a1631cd0-e5fb-xmpl-  
b474-0aa20f14f06e",  
      "Status": "CURRENT"  
    },  
  ],  
}
```

```
{
  "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
  "Region": "us-west-2",
  "Account": "123456789012",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/eab53680-e5fa-xmpl-
ba14-0a522351f81e",
  "Status": "OUTDATED",
  "StatusReason": "ResourceLogicalId:ConfigDeliveryChannel,
ResourceType:AWS::Config::DeliveryChannel, ResourceStatusReason:Failed to put
delivery channel 'StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532-
ConfigDeliveryChannel-10JWJ7XD59WR0' because the maximum number of delivery
channels: 1 is reached. (Service: AmazonConfig; Status Code: 400; Error Code:
MaxNumberOfDeliveryChannelsExceededException; Request ID: d14b34a0-ef7c-xmpl-
acf8-8a864370ae56)."
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStackInstances](#)」の「」を参照してください。

list-stack-resources

次の例は、list-stack-resources を使用する方法を説明しています。

AWS CLI

スタック内のリソースを一覧表示する

以下のコマンドは、指定されたスタック内にあるリソースのリストを表示します。

```
aws cloudformation list-stack-resources \
  --stack-name my-stack
```

出力:

```
{
  "StackResourceSummaries": [
    {
      "LogicalResourceId": "bucket",
      "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
```

```

    "ResourceType": "AWS::S3::Bucket",
    "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  },
  {
    "LogicalResourceId": "function",
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
    "ResourceType": "AWS::Lambda::Function",
    "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
    "ResourceStatus": "UPDATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  },
  {
    "LogicalResourceId": "functionRole",
    "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",
    "ResourceType": "AWS::IAM::Role",
    "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStackResources](#)」の「」を参照してください。

list-stack-set-operation-results

次の例は、list-stack-set-operation-results を使用する方法を説明しています。

AWS CLI

スタックセットオペレーション結果を一覧表示するには

次のコマンドは、指定されたスタックセット内のインスタンスに対する更新オペレーションの結果を表示します。

```
aws cloudformation list-stack-set-operation-results \  
  --stack-set-name enable-config \  
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

出力:

```
{  
  "Summaries": [  
    {  
      "Account": "223456789012",  
      "Region": "us-west-2",  
      "Status": "SUCCEEDED",  
      "AccountGateResult": {  
        "Status": "SKIPPED",  
        "StatusReason": "Function not found: arn:aws:lambda:eu-west-1:223456789012:function:AWSCloudFormationStackSetAccountGate"  
      }  
    },  
    {  
      "Account": "223456789012",  
      "Region": "ap-south-1",  
      "Status": "CANCELLED",  
      "StatusReason": "Cancelled since failure tolerance has exceeded"  
    }  
  ]  
}
```

注：アカウントゲート関数を作成しない限り、SKIPPEDのステータスAccountGateResultは正常なオペレーションに期待されます。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStackSetOperationResults](#)」の「」を参照してください。

list-stack-set-operations

次の例は、list-stack-set-operations を使用する方法を説明しています。

AWS CLI

スタックセットオペレーションを一覧表示するには

次のlist-stack-set-operations例では、指定されたスタックセットの最新のオペレーションのリストを表示します。

```
aws cloudformation list-stack-set-operations \  
  --stack-set-name my-stack-set
```

出力:

```
{  
  "Summaries": [  
    {  
      "OperationId": "35d45ebc-ed88-xmpl-ab59-0197a1fc83a0",  
      "Action": "UPDATE",  
      "Status": "SUCCEEDED",  
      "CreationTimestamp": "2019-10-03T16:28:44.377Z",  
      "EndTimestamp": "2019-10-03T16:42:08.607Z"  
    },  
    {  
      "OperationId": "891aa98f-7118-xmpl-00b2-00954d1dd0d6",  
      "Action": "UPDATE",  
      "Status": "FAILED",  
      "CreationTimestamp": "2019-10-03T15:43:53.916Z",  
      "EndTimestamp": "2019-10-03T15:45:58.925Z"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStackSetOperations](#)」の「」を参照してください。

list-stack-sets

次のコード例は、list-stack-sets を使用する方法を示しています。

AWS CLI

スタックセットを一覧表示するには

次のlist-stack-sets例では、現在のリージョンとアカウントのスタックセットのリストを表示します。

```
aws cloudformation list-stack-sets
```

出力:

```
{
  "Summaries": [
    {
      "StackSetName": "enable-config",
      "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
      "Description": "Enable AWS Config",
      "Status": "ACTIVE"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStackSets](#)」の「」を参照してください。

list-stacks

次の例は、list-stacks を使用する方法を説明しています。

AWS CLI

AWS CloudFormation スタックを一覧表示するには

以下の list-stacks コマンドは、ステータスが CREATE_COMPLETE になっているすべてのスタックの概要を表示します。

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

出力:

```
[
  {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
    "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket: Sample template showing how to create a publicly accessible S3 bucket. **WARNING** This template creates an S3 bucket. You will be billed for the AWS resources used if you create a stack from this template.",
  }
]
```



```
    "StackStatusReason": null,  
    "CreationTime": "2013-08-26T03:27:10.190Z",  
    "StackName": "myteststack",  
    "StackStatus": "CREATE_COMPLETE"  
  }  
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStacks](#)」の「」を参照してください。

list-type-registrations

次の例は、list-type-registrations を使用する方法を説明しています。

AWS CLI

タイプの完了済み登録を一覧表示するには

次のlist-type-registrations例では、指定されたタイプの完了したタイプ登録のリストを表示します。

```
aws cloudformation list-type-registrations \  
  --type RESOURCE \  
  --type-name My::Logs::LogGroup \  
  --registration-status-filter COMPLETE
```

出力:

```
{  
  "RegistrationTokenList": [  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
  ]  
}
```

詳細については、AWS CloudFormation 「ユーザーガイド」の [CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTypeRegistrations](#)」の「」を参照してください。

list-type-versions

次の例は、list-type-versions を使用する方法を説明しています。

AWS CLI

拡張機能のバージョンを一覧表示するには

次のlist-type-versions例では、拡張機能のバージョンに関する概要情報を返します。

```
aws cloudformation list-type-versions \  
  --endpoint https://example.com \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name My::Resource::Example \  
  --publisher-id 123456789012
```

このコマンドでは何も出力されません。

詳細については、AWS CloudFormation 「[ユーザーガイド](#)」の [AWS CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListTypeVersions](#)」の「」を参照してください。

list-types

次のコード例は、list-types を使用する方法を示しています。

AWS CLI

アカウントのプライベートリソースタイプを一覧表示するには

次のlist-types例では、現在の AWS アカウントで現在登録されているプライベートリソースタイプのリストを表示します。

```
aws cloudformation list-types
```

出力:

```
{  
  "TypeSummaries": [  
    ...  
  ]  
}
```

```
{
  "Description": "WordPress blog resource for internal use",
  "LastUpdated": "2019-12-04T18:28:15.059Z",
  "TypeName": "My::WordPress::BlogExample",
  "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-WordPress-BlogExample",
  "DefaultVersionId": "00000005",
  "Type": "RESOURCE"
},
{
  "Description": "Customized resource derived from AWS::Logs::LogGroup",
  "LastUpdated": "2019-12-04T18:28:15.059Z",
  "TypeName": "My::Logs::LogGroup",
  "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
My-Logs-LogGroup",
  "DefaultVersionId": "00000003",
  "Type": "RESOURCE"
}
]
```

詳細については、AWS CloudFormation [「ユーザーガイド」の CloudFormation 「レジストリの使用」](#)を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスListTypes」](#)の「」を参照してください。

package

次のコード例は、package を使用する方法を示しています。

AWS CLI

次のコマンドは、ローカルアーティファクトを S3 バケットにアップロードtemplate.jsonしてという名前のテンプレートをエクスポートbucket-nameし、エクスポートされたテンプレートをに書き込みますpackaged-template.json。

```
aws cloudformation package --template-file /path_to_template/template.json --s3-
bucket bucket-name --output-template-file packaged-template.json --use-json
```

- API 詳細については、「[コマンドリファレンス](#)」の[「パッケージ」](#)を参照してください。

AWS CLI

publish-type

次のコード例は、publish-type を使用する方法を示しています。

AWS CLI

拡張機能を発行するには

次のpublish-type例では、指定された拡張機能をこのリージョンのパブリック拡張機能として CloudFormation レジストリに公開します。

```
aws cloudformation publish-type \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name Example::Test::1234567890abcdef0
```

出力:

```
{  
  "PublicTypeArn": "arn:aws:cloudformation:us-west-2::type/  
resource/000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c/Example-  
Test-1234567890abcdef0/1.0.0"  
}
```

詳細については、AWS CloudFormation [「ユーザーガイド」の AWS CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスPublishType」](#) の「」を参照してください。

register-publisher

次の例は、register-publisher を使用する方法を説明しています。

AWS CLI

パブリッシャーを登録するには

次のregister-publisher例では、パブリッシャーを登録し、条件パラメータを受け入れません。

```
aws cloudformation register-publisher \  

```

```
--region us-west-2 \  
--accept-terms-and-conditions
```

出力:

```
{  
  "PublisherId": "000q6TfUovXsEMmgKowxDZLlWqr2QUshd2e75c8c"  
}
```

詳細については、AWS CloudFormation 「[ユーザーガイド](#)」の [AWS CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスRegisterPublisher](#)」の「」を参照してください。

register-type

次の例は、register-type を使用する方法を説明しています。

AWS CLI

リソースタイプを登録するには

次のregister-type例では、指定されたリソースタイプをユーザーのアカウントにプライベートリソースタイプとして登録します。

```
aws cloudformation register-type \  
  --type-name My::Organization::ResourceName \  
  --schema-handler-package s3://bucket_name/my-organization-resource_name.zip \  
  --type RESOURCE
```

出力:

```
{  
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1f1example"  
}
```

詳細については、「[タイプ開発のコマンドラインインターフェイスユーザーガイド](#)」の「[リソースプロバイダーの登録](#)」を参照してください。 CloudFormation

- API 詳細については、AWS CLI 「[コマンドリファレンスRegisterType](#)」の「」を参照してください。

set-stack-policy

次の例は、set-stack-policy を使用する方法を説明しています。

AWS CLI

スタックポリシーを適用するには

次のset-stack-policy例では、指定されたスタック内の指定されたリソースの更新を無効にします。stack-policy.jsonは、スタック内のリソースで許可されるオペレーションを定義するJSONドキュメントです。

```
aws cloudformation set-stack-policy \  
  --stack-name my-stack \  
  --stack-policy-body file://stack-policy.json
```

出力:

```
{  
  "Statement" : [  
    {  
      "Effect" : "Allow",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "*"   
    },  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "LogicalResourceId/bucket"   
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetStackPolicy](#)」の「」を参照してください。

set-type-configuration

次のコード例は、set-type-configuration を使用する方法を示しています。

AWS CLI

データを設定するには

次のset-type-configuration例では、特定のアカウントとリージョンに登録されたCloudFormation 拡張機能の設定データを指定します。

```
aws cloudformation set-type-configuration \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name Example::Test::Type \  
  --configuration-alias default \  
  --configuration "{\"CredentialKey\": \"testUserCredential\"}"
```

出力:

```
{  
  "ConfigurationArn": "arn:aws:cloudformation:us-west-2:123456789012:type-  
configuration/resource/Example-Test-Type/default"  
}
```

詳細については、AWS CloudFormation 「[ユーザーガイド](#)」の [AWS CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスSetTypeConfiguration](#)」の「」を参照してください。

set-type-default-version

次の例は、set-type-default-version を使用する方法を説明しています。

AWS CLI

タイプのデフォルトバージョンを設定するには

次のset-type-default-version例では、このタイプのデフォルトとして使用するよう指定されたタイプバージョンを設定します。

```
aws cloudformation set-type-default-version \  
  --type RESOURCE \  
  --type-name My::Logs::LogGroup \  
  --version 1
```

```
--version-id 00000003
```

このコマンドでは何も出力されません。

詳細については、AWS CloudFormation 「ユーザーガイド」の [CloudFormation 「レジストリの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetTypeDefaultVersion](#)」の「」を参照してください。

signal-resource

次のコード例は、signal-resource を使用する方法を示しています。

AWS CLI

リソースにシグナルを送信するには

次のsignal-resource例では、 という名前のスタックMyWaitConditionで という名前の待機条件を満たすsuccessようにシグナルを送りますmy-stack。

```
aws cloudformation signal-resource \  
  --stack-name my-stack \  
  --logical-resource-id MyWaitCondition \  
  --unique-id 1234 \  
  --status SUCCESS
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [SignalResource](#)」の「」を参照してください。

stop-stack-set-operation

次のコード例は、stop-stack-set-operation を使用する方法を示しています。

AWS CLI

スタックセットオペレーションを停止するには

次のstop-stack-set-operation例では、指定されたスタックセットのインポートゲーション更新オペレーションを停止します。


```
aws cloudformation stop-stack-set-operation \  
  --stack-set-name my-stack-set \  
  --operation-id 1261cd27-490b-xmpl-ab42-793a896c69e6
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopStackSetOperation](#)」の「」を参照してください。

test-type

次の例は、test-type を使用する方法を説明しています。

AWS CLI

拡張機能をテストするには

次のtest-type例では、登録された拡張機能をテストして、CloudFormation レジストリで公開するために必要なすべての要件を満たしていることを確認します。

```
aws cloudformation test-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-Resource123/00000001
```

出力:

```
{  
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/  
Sample-Test-Resource123/00000001"  
}
```

詳細については、AWS CloudFormation 「ユーザーガイド」の[AWS CloudFormation 「レジストリの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TestType](#)」の「」を参照してください。

update-stack-instances

次の例は、update-stack-instances を使用する方法を説明しています。

AWS CLI

スタックインスタンスを更新するには

次のupdate-stack-instances例では、最新の設定で2つのリージョンの2つのアカウントのスタックインスタンスの更新を再試行します。指定された耐障害性設定により、一部のスタックを更新できない場合でも、すべてのアカウントとリージョンで更新が試行されます。

```
aws cloudformation update-stack-instances \  
  --stack-set-name my-stack-set \  
  --accounts 123456789012 567890123456 \  
  --regions us-east-1 us-west-2 \  
  --operation-preferences FailureToleranceCount=3
```

出力:

```
{  
  "OperationId": "103ebdf2-21ea-xmpl-8892-de5e30733132"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateStackInstances](#)」の「」を参照してください。

update-stack-set

次の例は、update-stack-set を使用する方法を説明しています。

AWS CLI

スタックセットを更新するには

次のupdate-stack-set例では、指定されたスタックセットのスタックインスタンスITにキー名Ownerと の値を持つタグを追加します。

```
aws cloudformation update-stack-set \  
  --stack-set-name my-stack-set \  
  --use-previous-template \  
  --tags Key=Owner, Value=IT
```

出力:

```
{
  "OperationId": "e2b60321-6cab-xmpl-bde7-530c6f47950e"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateStackSet](#)」の「」を参照してください。

update-stack

次の例は、update-stack を使用する方法を説明しています。

AWS CLI

AWS CloudFormation スタックを更新するには

以下の update-stack コマンドは、mystack スタックのテンプレートと入力パラメータを更新します。

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template --
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

以下の update-stack コマンドは、mystack スタックの SubnetIDs パラメータ値のみを更新します。パラメータ値を指定しない場合は、テンプレートで指定されているデフォルト値が使用されます。

```
aws cloudformation update-stack --stack-name mystack --
template-url https://s3.amazonaws.com/sample/updated.template
--parameters ParameterKey=KeyPairName,UsePreviousValue=true
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

以下の update-stack コマンドは、mystack スタックに 2 つのスタック通知トピックを追加します。

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --
notification-arns "arn:aws:sns:use-east-1:123456789012:mytopic1" "arn:aws:sns:us-
east-1:123456789012:mytopic2"
```

詳細については、「ユーザーガイド」の[AWS CloudFormation 「スタックの更新」](#)を参照してください。AWS CloudFormation

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateStack](#)」の「」を参照してください。

update-termination-protection

次の例は、update-termination-protection を使用する方法を説明しています。

AWS CLI

終了保護を有効にするには

次のupdate-termination-protection例では、指定されたスタックで終了保護を有効にします。

```
aws cloudformation update-termination-protection \  
  --stack-name my-stack \  
  --enable-termination-protection
```

出力:

```
{  
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/  
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTerminationProtection](#)」の「」を参照してください。

validate-template

次のコード例は、validate-template を使用する方法を示しています。

AWS CLI

AWS CloudFormation テンプレートを検証するには

以下の validate-template コマンドは、sampletemplate.json テンプレートを検証します。

```
aws cloudformation validate-template --template-body file://sampltemplate.json
```

出力:

```
{
  "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
  showing how to create a publicly accessible S3 bucket. **WARNING** This template
  creates an S3 bucket. You will be billed for the AWS resources used if you create a
  stack from this template.",
  "Parameters": [],
  "Capabilities": []
}
```

詳細については、AWS CloudFormation 「ユーザーガイド」の AWS CloudFormation 「テンプレートの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ValidateTemplate](#)」の「」を参照してください。

CloudFront を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CloudFront。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-cloud-front-origin-access-identity

次のコード例は、create-cloud-front-origin-access-identity を使用する方法を示しています。

AWS CLI

CloudFront オリジンアクセス ID を作成するには

次の例では、コマンドライン引数としてOAI設定を指定してオ CloudFront リジンアクセスアイデンティティ (OAI) を作成します。

```
aws cloudfront create-cloud-front-origin-access-identity \  
  --cloud-front-origin-access-identity-config \  
    CallerReference="cli-example",Comment="Example OAI"
```

次の例に示すように、JSON ファイルでOAI設定を指定することで、同じことを実行できます。

```
aws cloudfront create-cloud-front-origin-access-identity \  
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

OAI-config.json ファイルは、現在のディレクトリにあるJSONドキュメントで、以下が含まれています。

```
{  
  "CallerReference": "cli-example",  
  "Comment": "Example OAI"  
}
```

OAI 設定にコマンドライン引数とJSONファイルのどちらを指定しても、出力は同じです。

```
{  
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/origin-access-identity/  
cloudfront/E74FTE3AEXAMPLE",  
  "ETag": "E2QWRUHEXAMPLE",  
  "CloudFrontOriginAccessIdentity": {  
    "Id": "E74FTE3AEXAMPLE",  
    "S3CanonicalUserId":  
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",  
    "CloudFrontOriginAccessIdentityConfig": {  
      "CallerReference": "cli-example",  
      "Comment": "Example OAI"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateCloudFrontOriginAccessIdentity](#)」の「」を参照してください。

create-distribution-with-tags

次のコード例は、create-distribution-with-tags を使用する方法を示しています。

AWS CLI

タグを使用して CloudFront デイストリビューションを作成するには

次の例では、という名前のJSONファイルにデイストリビューション設定とタグを指定して、2つのタグを持つデイストリビューションを作成しますdist-config-with-tags.json。

```
aws cloudfront create-distribution-with-tags \  
  --distribution-config-with-tags file://dist-config-with-tags.json
```

dist-config-with-tags.json ファイルは、現在のフォルダにあるJSONドキュメントで、以下が含まれています。ファイルの上部にあるTagsオブジェクトに、次の2つのタグがあります。

Name = ExampleDistributionProject = ExampleProject

```
{  
  "Tags": {  
    "Items": [  
      {  
        "Key": "Name",  
        "Value": "ExampleDistribution"  
      },  
      {  
        "Key": "Project",  
        "Value": "ExampleProject"  
      }  
    ]  
  },  
  "DistributionConfig": {  
    "CallerReference": "cli-example",  
    "Aliases": {  
      "Quantity": 0  
    },  
    "DefaultRootObject": "index.html",
```

```
"Origins": {
  "Quantity": 1,
  "Items": [
    {
      "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
      "DomainName": "awsexamplebucket.s3.amazonaws.com",
      "OriginPath": "",
      "CustomHeaders": {
        "Quantity": 0
      },
      "S3OriginConfig": {
        "OriginAccessIdentity": ""
      }
    }
  ]
},
"OriginGroups": {
  "Quantity": 0
},
"DefaultCacheBehavior": {
  "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
  "ForwardedValues": {
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    },
  },
  "Headers": {
    "Quantity": 0
  },
  "QueryStringCacheKeys": {
    "Quantity": 0
  }
},
"TrustedSigners": {
  "Enabled": false,
  "Quantity": 0
},
"ViewerProtocolPolicy": "allow-all",
"MinTTL": 0,
"AllowedMethods": {
  "Quantity": 2,
  "Items": [
    "HEAD",
    "GET"
  ]
}
```



```
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
},
"Comment": "",
"Logging": {
  "Enabled": false,
  "IncludeCookies": false,
  "Bucket": "",
  "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
  "CloudFrontDefaultCertificate": true,
  "MinimumProtocolVersion": "TLSv1",
  "CertificateSource": "cloudfront"
},
"Restrictions": {
  "GeoRestriction": {
    "RestrictionType": "none",
    "Quantity": 0
  }
},
},
```

```
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}
```

出力:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/EDFDVBD6EXAMPLE",
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    "DistributionConfig": {
      "CallerReference": "cli-example",
      "Aliases": {
        "Quantity": 0
      },
      "DefaultRootObject": "index.html",
      "Origins": {
        "Quantity": 1,
        "Items": [
          {
            "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
            "DomainName": "awsexamplebucket.s3.amazonaws.com",
            "OriginPath": "",
            "CustomHeaders": {
              "Quantity": 0
            },
            "S3OriginConfig": {
              "OriginAccessIdentity": ""
            }
          }
        ]
      }
    }
  }
}
```

```
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      },
      "Headers": {
        "Quantity": 0
      },
      "QueryStringCacheKeys": {
        "Quantity": 0
      }
    },
    "TrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ],
      "CachedMethods": {
        "Quantity": 2,
        "Items": [
          "HEAD",
          "GET"
        ]
      }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
```

```
        "Quantity": 0
      },
      "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
      "Quantity": 0
    },
    "CustomErrorResponses": {
      "Quantity": 0
    },
    "Comment": "",
    "Logging": {
      "Enabled": false,
      "IncludeCookies": false,
      "Bucket": "",
      "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDistributionWithTags](#)」の「」を参照してください。

create-distribution

次の例は、create-distribution を使用する方法を説明しています。

AWS CLI

CloudFront デистриビューションを作成するには

以下の例では、`awsexamplebucket` という名前の S3 バケットのデистриビューションを作成し、コマンドライン引数を使用してデフォルトのルートオブジェクトとして `index.html` を指定しています。

```
aws cloudfront create-distribution \  
  --origin-domain-name awsexamplebucket.s3.amazonaws.com \  
  --default-root-object index.html
```

コマンドライン引数を使用する代わりに、次の例に示すように、JSON ファイルにデистриビューション設定を指定できます。

```
aws cloudfront create-distribution \  
  --distribution-config file://dist-config.json
```

`dist-config.json` ファイルは、現在のフォルダ内の JSON ドキュメントで、以下が含まれています。

```
{  
  "CallerReference": "cli-example",  
  "Aliases": {  
    "Quantity": 0  
  },  
  "DefaultRootObject": "index.html",  
  "Origins": {  
    "Quantity": 1,  
    "Items": [  
      {  
        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",  
        "DomainName": "awsexamplebucket.s3.amazonaws.com",  
        "OriginPath": "",  
        "CustomHeaders": {  
          "Quantity": 0  
        },  
        "S3OriginConfig": {  
          "OriginAccessIdentity": ""  
        }  
      }  
    ]  
  }  
}
```

```
},
"OriginGroups": {
  "Quantity": 0
},
"DefaultCacheBehavior": {
  "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
  "ForwardedValues": {
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  }
}
```

```

    },
    "FieldLevelEncryptionId": ""
  },
  "CacheBehaviors": {
    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "",
  "Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
  },
  "PriceClass": "PriceClass_All",
  "Enabled": true,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  },
  "WebACLId": "",
  "HttpVersion": "http2",
  "IsIPV6Enabled": true
}

```

ディストリビューション情報をコマンドライン引数またはJSONファイルのどちらで指定しても、出力は同じです。

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EMLARXS9EXAMPLE",
  "ETag": "E9LHASXEXAMPLE",
  "Distribution": {
    "Id": "EMLARXS9EXAMPLE",

```

```
"ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
"Status": "InProgress",
"LastModifiedTime": "2019-11-22T00:55:15.705Z",
"InProgressInvalidationBatches": 0,
"DomainName": "d1111111abcdef8.cloudfront.net",
"ActiveTrustedSigners": {
  "Enabled": false,
  "Quantity": 0
},
"DistributionConfig": {
  "CallerReference": "cli-example",
  "Aliases": {
    "Quantity": 0
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        }
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      }
    },
    "Headers": {
      "Quantity": 0
    }
  }
}
```



```
        "QueryStringCacheKeys": {
            "Quantity": 0
        },
    },
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
```

```
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDistribution](#)」の「」を参照してください。

create-field-level-encryption-config

次の例は、create-field-level-encryption-config を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化設定を作成するには

次の例では、 という名前のJSONファイルに設定パラメータを指定して、フィールドレベルの暗号化設定を作成しますfle-config.json。フィールドレベルの暗号化設定を作成する前に、フィールドレベルの暗号化プロファイルが必要です。プロファイルを作成するには、 create-field-level-encryption-profile コマンドを参照してください。

CloudFront フィールドレベルの暗号化の詳細については、「Amazon CloudFront デベロッパーガイド」の「[フィールドレベルの暗号化を使用して機密データを保護する](#)」を参照してください。

```
aws cloudfront create-field-level-encryption-config \
```

```
--field-level-encryption-config file://fle-config.json
```

fle-config.json ファイルは、現在のフォルダ内のJSONドキュメントで、以下が含まれています。

```
{
  "CallerReference": "cli-example",
  "Comment": "Example FLE configuration",
  "QueryArgProfileConfig": {
    "ForwardWhenQueryArgProfileIsUnknown": true,
    "QueryArgProfiles": {
      "Quantity": 0
    }
  },
  "ContentTypeProfileConfig": {
    "ForwardWhenContentTypeIsUnknown": true,
    "ContentTypeProfiles": {
      "Quantity": 1,
      "Items": [
        {
          "Format": "URLEncoded",
          "ProfileId": "P280MFCLSY0CVU",
          "ContentType": "application/x-www-form-urlencoded"
        }
      ]
    }
  }
}
```

出力:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption/C3KM2WVD605UAY",
  "ETag": "E2P4Z4VU7TY5SG",
  "FieldLevelEncryption": {
    "Id": "C3KM2WVD605UAY",
    "LastModifiedTime": "2019-12-10T21:30:18.974Z",
    "FieldLevelEncryptionConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example FLE configuration",
      "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
```

```

    "QueryArgProfiles": {
      "Quantity": 0,
      "Items": []
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSYOCVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateFieldLevelEncryptionConfig](#)」の「」を参照してください。

create-field-level-encryption-profile

次の例は、create-field-level-encryption-profile を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化プロファイルを作成するには

次の例では、という名前のJSONファイルにパラメータを指定して、フィールドレベルの暗号化プロファイルを作成しますfle-profile-config.json。フィールドレベルの暗号化プロファイルを作成する前に、CloudFront パブリックキーが必要です。CloudFront パブリックキーを作成するには、コマンドを参照してください create-public-key。

CloudFront フィールドレベルの暗号化の詳細については、「Amazon CloudFront デベロップャーガイド」の「[フィールドレベルの暗号化を使用した機密データの保護](#)」を参照してください。

```
aws cloudfront create-field-level-encryption-profile \
```

```
--field-level-encryption-profile-config file://fle-profile-config.json
```

fle-profile-config.json ファイルは、現在のフォルダにあるJSONドキュメントで、以下が含まれています。

```
{
  "Name": "ExampleFLEProfile",
  "CallerReference": "cli-example",
  "Comment": "FLE profile for AWS CLI example",
  "EncryptionEntities": {
    "Quantity": 1,
    "Items": [
      {
        "PublicKeyId": "K2K8NC4HVFE3M0",
        "ProviderId": "ExampleFLEProvider",
        "FieldPatterns": {
          "Quantity": 1,
          "Items": [
            "ExampleSensitiveField"
          ]
        }
      }
    ]
  }
}
```

出力:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption-profile/PPK0U0SIF5WSV",
  "ETag": "E2QWRUHEXAMPLE",
  "FieldLevelEncryptionProfile": {
    "Id": "PPK0U0SIF5WSV",
    "LastModifiedTime": "2019-12-10T01:03:16.537Z",
    "FieldLevelEncryptionProfileConfig": {
      "Name": "ExampleFLEProfile",
      "CallerReference": "cli-example",
      "Comment": "FLE profile for AWS CLI example",
      "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
          {

```

```
{
  "PublicKeyId": "K2K8NC4HVFE3M0",
  "ProviderId": "ExampleFLEProvider",
  "FieldPatterns": {
    "Quantity": 1,
    "Items": [
      "ExampleSensitiveField"
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFieldLevelEncryptionProfile](#)」の「」を参照してください。

create-invalidation

次の例は、create-invalidation を使用する方法を説明しています。

AWS CLI

CloudFront デистриビューションの無効化を作成するには

次のcreate-invalidation例では、指定された CloudFront デистриビューション内の指定されたファイルの無効化を作成します。

```
aws cloudfront create-invalidation \  
  --distribution-id EDFDVBD6EXAMPLE \  
  --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

出力:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/  
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",
  "Invalidation": {
    "Id": "I1JLWSDAP8FU89",
    "Status": "InProgress",
    "CreateTime": "2019-12-05T18:24:51.407Z",
```

```

    "InvalidationBatch": {
      "Paths": {
        "Quantity": 2,
        "Items": [
          "/example-path/example-file2.png",
          "/example-path/example-file.jpg"
        ]
      },
      "CallerReference": "cli-1575570291-670203"
    }
  }
}

```

前の例では、AWS CLIはランダムな を自動的に生成しましたCallerReference。独自の を指定するかCallerReference、無効化パラメータをコマンドライン引数として渡さないようにするには、JSON ファイルを使用できます。次の例では、 という名前のファイルに無効化パラメータを指定することで、2つのJSONファイルの無効化を作成しますinv-batch.json。

```

aws cloudfront create-invalidation \
  --distribution-id EDFDVBD6EXAMPLE \
  --invalidation-batch file://inv-batch.json

```

inv-batch.json の内容:

```

{
  "Paths": {
    "Quantity": 2,
    "Items": [
      "/example-path/example-file.jpg",
      "/example-path/example-file2.png"
    ]
  },
  "CallerReference": "cli-example"
}

```

出力:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUY0IK",
  "Invalidation": {
    "Id": "I2J0I21PCUY0IK",

```

```

    "Status": "InProgress",
    "CreateTime": "2019-12-05T18:40:49.413Z",
    "InvalidationBatch": {
      "Paths": {
        "Quantity": 2,
        "Items": [
          "/example-path/example-file.jpg",
          "/example-path/example-file2.png"
        ]
      },
      "CallerReference": "cli-example"
    }
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateInvalidation](#)」の「」を参照してください。

create-public-key

次のコード例は、create-public-key を使用する方法を示しています。

AWS CLI

CloudFront パブリックキーを作成するには

次の例では、という名前のJSONファイルにパラメータを指定して CloudFront パブリックキーを作成します pub-key-config.json。このコマンドを使用する前に、PEMエンコードされたパブリックキーが必要です。詳細については、「Amazon [デベロッパーガイド](#)」のRSA「[キーペアの作成](#)」を参照してください。 CloudFront

```

aws cloudfront create-public-key \
  --public-key-config file://pub-key-config.json

```

pub-key-config.json ファイルは、現在のフォルダにあるJSONドキュメントで、以下が含まれています。パブリックキーは PEM 形式でエンコードされることに注意してください。

```

{
  "CallerReference": "cli-example",
  "Name": "ExampleKey",
  "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPmBCA2Ks01nd7IR+3pw

```



```

\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWnWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLK0X5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwiDAQAB\n-----END
PUBLIC KEY-----\n",
  "Comment": "example public key"
}

```

出力:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/public-key/
KDFB19YGCR002",
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKey": {
    "Id": "KDFB19YGCR002",
    "CreatedTime": "2019-12-05T18:51:43.781Z",
    "PublicKeyConfig": {
      "CallerReference": "cli-example",
      "Name": "ExampleKey",
      "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBCGKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWnWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLK0X5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwiDAQAB\n-----END
PUBLIC KEY-----\n",
      "Comment": "example public key"
    }
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePublicKey](#)」の「」を参照してください。

delete-cloud-front-origin-access-identity

次のコード例は、delete-cloud-front-origin-access-identity を使用する方法を示しています。

AWS CLI

CloudFront オリジンアクセス ID を削除するには

次の例では、ID を持つオリジンアクセス ID (OAI) を削除しますE74FTE3AEXAMPLE。を削除するにはOAI、 OAIの ID と が必要ですETag。OAI ID は、create-cloud-front-origin-access-identity コマンドと list-cloud-front-origin-access-identities コマンドの出力で返されます。を取得するにはETag、get-cloud-front-origin-access-identity または get-cloud-front-origin-access-identity-config command を使用します。--if-match オプションを使用して、 OAIの を指定しますETag。

```
aws cloudfront delete-cloud-front-origin-access-identity \  
  --id E74FTE3AEXAMPLE \  
  --if-match E2QWRUHEXAMPLE
```

成功した場合は、コマンドの出力はありません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteCloudFrontOriginAccessIdentity](#)」の「」を参照してください。

delete-distribution

次の例は、delete-distribution を使用する方法を説明しています。

AWS CLI

CloudFront デイストリビューションを削除するには

次の例では、ID が の CloudFront デイストリビューションを削除しますEDFDVBD6EXAMPLE。デイストリビューションを削除する前に、デイストリビューションを無効にする必要があります。デイストリビューションを無効にするには、update-distribution コマンドを使用します。詳細については、update-distribution の例を参照してください。

デイストリビューションを無効にすると、デイストリビューションを削除できます。デイストリビューションを削除するには、--if-match オプションを使用してデイストリビューション のETag を指定する必要があります。を取得するにはETag、get-distribution または get-distribution-config コマンドを使用します。

```
aws cloudfront delete-distribution \  
  --id EDFDVBD6EXAMPLE \  
  --if-match E2QWRUHEXAMPLE
```

成功した場合は、コマンドの出力はありません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDistribution](#)」の「」を参照してください。

delete-field-level-encryption-config

次のコード例は、delete-field-level-encryption-config を使用する方法を示しています。

AWS CLI

CloudFront フィールドレベルの暗号化設定を削除するには

次の例では、ID を持つ CloudFront フィールドレベルの暗号化設定を削除します。C3KM2WVD605UAY。フィールドレベルの暗号化設定を削除するには、その ID と ETag が必要です。ID は create-field-level-encryption-config コマンドと list-field-level-encryption-configs コマンドの出力で返されます。ETag を取得するには get-field-level-encryption または get-field-level-encryption-config コマンドを使用します。--if-match オプションを使用して、設定の ETag を指定します。

```
aws cloudfront delete-field-level-encryption-config \  
  --id C3KM2WVD605UAY \  
  --if-match E26M4BIAV81ZF6
```

成功した場合は、コマンドの出力はありません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFieldLevelEncryptionConfig](#)」の「」を参照してください。

delete-field-level-encryption-profile

次の例は、delete-field-level-encryption-profile を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化プロファイルを削除するには

次の例では、ID を持つ CloudFront フィールドレベルの暗号化プロファイルを削除します。PPK0UOSIF5WSV。フィールドレベルの暗号化プロファイルを削除するには、その ID と ETag が必要です。ID は create-field-level-encryption-profile コマンドと list-field-level-encryption-

profiles コマンドの出力で返されます。を取得するにはETag、get-field-level-encryption-profile コマンドまたは get-field-level-encryption-profile-config コマンドを使用します。--if-match オプションを使用して、プロファイルの を指定しますETag。

```
aws cloudfront delete-field-level-encryption-profile \  
  --id PPK0UOSIF5WSV \  
  --if-match EJETYFJ9CL66D
```

成功した場合は、コマンドの出力はありません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFieldLevelEncryptionProfile](#)」の「」を参照してください。

delete-public-key

次のコード例は、delete-public-key を使用する方法を示しています。

AWS CLI

CloudFront パブリックキーを削除するには

次の例では、ID が の CloudFront パブリックキーを削除しますKDFB19YGCR002。パブリックキーを削除するには、その ID と が必要ですETag。ID は、create-public-key および list-public-keys コマンドの出力で返されます。を取得するにはETag、get-public-key または get-public-key-config コマンドを使用します。--if-match オプションを使用して、パブリックキーの を指定しますETag。

```
aws cloudfront delete-public-key \  
  --id KDFB19YGCR002 \  
  --if-match E2QWRUHEXAMPLE
```

成功した場合は、コマンドの出力はありません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePublicKey](#)」の「」を参照してください。

get-cloud-front-origin-access-identity-config

次の例は、get-cloud-front-origin-access-identity-config を使用する方法を説明しています。

AWS CLI

CloudFront オリジンアクセス ID 設定を取得するには

次の例では、E74FTE3AEXAMPLEを含む ID を持つ CloudFront オリジンアクセス ID (OAI) に関するメタデータを取得しますETag。OAI ID は、create-cloud-front-origin-access-identity コマンドと list-cloud-front-origin-access-identities コマンドの出力で返されます。

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

出力:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentityConfig": {
    "CallerReference": "cli-example",
    "Comment": "Example OAI"
  }
}
```

- API 詳細については、AWS CLI 「[コマンドリファレンスGetCloudFrontOriginAccessIdentityConfig](#)」の「」を参照してください。

get-cloud-front-origin-access-identity

次の例は、get-cloud-front-origin-access-identity を使用する方法を説明しています。

AWS CLI

CloudFront オリジンアクセス ID を取得するには

次の例では、ID ETagと関連付けられた S3 正規 ID を含むE74FTE3AEXAMPLE、ID を持つ CloudFront オリジンアクセス ID (OAI) を取得します。OAI ID は、create-cloud-front-origin-access-identity コマンドと list-cloud-front-origin-access-identities コマンドの出力で返されます。

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

出力:

```
{
  "ETag": "E2QWRUHEXAMPLE",
```

```
"CloudFrontOriginAccessIdentity": {
  "Id": "E74FTE3AEXAMPLE",
  "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
  "CloudFrontOriginAccessIdentityConfig": {
    "CallerReference": "cli-example",
    "Comment": "Example OAI"
  }
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCloudFrontOriginAccessIdentity](#)」の「」を参照してください。

get-distribution-config

次の例は、get-distribution-config を使用する方法を説明しています。

AWS CLI

CloudFront デイストリビューション設定を取得するには

次の例ではEDFDVBD6EXAMPLE、を含む ID を持つ CloudFront デイストリビューションに関するメタデータを取得しますETag。デイストリビューション ID は create-distribution コマンドと list-distributions コマンドで返されます。

```
aws cloudfront get-distribution-config --id EDFDVBD6EXAMPLE
```

出力:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
```

```
        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        }
    }
]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        },
    },
    "Headers": {
        "Quantity": 0
    },
    "QueryStringCacheKeys": {
        "Quantity": 0
    }
},
"TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
},
"ViewerProtocolPolicy": "allow-all",
"MinTTL": 0,
"AllowedMethods": {
    "Quantity": 2,
    "Items": [
        "HEAD",
        "GET"
    ],
},
"CachedMethods": {
    "Quantity": 2,
    "Items": [
```

```
        "HEAD",
        "GET"
    ]
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
},
"Comment": "",
"Logging": {
  "Enabled": false,
  "IncludeCookies": false,
  "Bucket": "",
  "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
  "CloudFrontDefaultCertificate": true,
  "MinimumProtocolVersion": "TLSv1",
  "CertificateSource": "cloudfront"
},
"Restrictions": {
  "GeoRestriction": {
    "RestrictionType": "none",
    "Quantity": 0
  }
},
"WebACLId": "",
"HttpVersion": "http2",
"IsIPV6Enabled": true
}
```



```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDistributionConfig](#)」の「」を参照してください。

get-distribution

次のコード例は、get-distribution を使用する方法を示しています。

AWS CLI

CloudFront デイストリビューションを取得するには

次の例では、EDFDVBD6EXAMPLEを含む ID で CloudFront デイストリビューションを取得します ETag。デイストリビューション ID は create-distribution コマンドと list-distributions コマンドで返されます。

```
aws cloudfront get-distribution --id EDFDVBD6EXAMPLE
```

出力:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "Deployed",
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
  },
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
```

```
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        }
      }
    ],
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      },
      "Headers": {
        "Quantity": 0
      },
      "QueryStringCacheKeys": {
        "Quantity": 0
      }
    },
    "TrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ],
      "CachedMethods": {
```

```
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ]
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
},
"Restrictions": {
    "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
    }
},
"WebACLId": "",
"HttpVersion": "http2",
```

```
        "IsIPV6Enabled": true
      }
    }
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDistribution](#)」の「」を参照してください。

get-field-level-encryption-config

次のコード例は、get-field-level-encryption-config を使用する方法を示しています。

AWS CLI

CloudFront フィールドレベルの暗号化設定に関するメタデータを取得するには

次の例ではC3KM2WVD605UAY、を含む ID を持つ CloudFront フィールドレベルの暗号化設定に関するメタデータを取得しますETag。

```
aws cloudfront get-field-level-encryption-config --id C3KM2WVD605UAY
```

出力:

```
{
  "ETag": "E2P4Z4VU7TY5SG",
  "FieldLevelEncryptionConfig": {
    "CallerReference": "cli-example",
    "Comment": "Example FLE configuration",
    "QueryArgProfileConfig": {
      "ForwardWhenQueryArgProfileIsUnknown": true,
      "QueryArgProfiles": {
        "Quantity": 0,
        "Items": []
      }
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
```

```

        "ProfileId": "P280MFCLSY0CVU",
        "ContentType": "application/x-www-form-urlencoded"
    }
  ]
}
}
}
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetFieldLevelEncryptionConfig](#)」の「」を参照してください。

get-field-level-encryption-profile-config

次のコード例は、get-field-level-encryption-profile-config を使用する方法を示しています。

AWS CLI

CloudFront フィールドレベルの暗号化プロファイル設定を取得するには

次の例では、を含む ID PPK0UOSIF5WSV を持つ CloudFront フィールドレベルの暗号化プロファイルに関するメタデータを取得しますETag。

```
aws cloudfront get-field-level-encryption-profile-config --id PPK0UOSIF5WSV
```

出力:

```

{
  "ETag": "E1QQG65FS2L2GC",
  "FieldLevelEncryptionProfileConfig": {
    "Name": "ExampleFLEProfile",
    "CallerReference": "cli-example",
    "Comment": "FLE profile for AWS CLI example",
    "EncryptionEntities": {
      "Quantity": 1,
      "Items": [
        {
          "PublicKeyId": "K2K8NC4HVFE3M0",
          "ProviderId": "ExampleFLEProvider",
          "FieldPatterns": {
            "Quantity": 1,

```

```
        "Items": [  
            "ExampleSensitiveField"  
        ]  
    }  
}  
]  
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetFieldLevelEncryptionProfileConfig](#)」の「」を参照してください。

get-field-level-encryption-profile

次の例は、`get-field-level-encryption-profile` を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化プロファイルを取得するには

次の例では、を含む ID `PPK0U0SIF5WSV` で CloudFront フィールドレベルの暗号化プロファイルを取得します ETag。

```
aws cloudfront get-field-level-encryption-profile --id PPK0U0SIF5WSV
```

出力:

```
{  
  "ETag": "E1QQG65FS2L2GC",  
  "FieldLevelEncryptionProfile": {  
    "Id": "PPK0U0SIF5WSV",  
    "LastModifiedTime": "2019-12-10T01:03:16.537Z",  
    "FieldLevelEncryptionProfileConfig": {  
      "Name": "ExampleFLEProfile",  
      "CallerReference": "cli-example",  
      "Comment": "FLE profile for AWS CLI example",  
      "EncryptionEntities": {  
        "Quantity": 1,  
        "Items": [  
          {  
            "PublicKeyId": "K2K8NC4HVFE3M0",
```

```
      "ProviderId": "ExampleFLEProvider",
      "FieldPatterns": {
        "Quantity": 1,
        "Items": [
          "ExampleSensitiveField"
        ]
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFieldLevelEncryptionProfile](#)」の「」を参照してください。

get-field-level-encryption

次の例は、get-field-level-encryption を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化設定を取得するには

次の例では、C3KM2WVD605UAYを含む ID で CloudFront フィールドレベルの暗号化設定を取得しますETag。

```
aws cloudfront get-field-level-encryption --id C3KM2WVD605UAY
```

出力:

```
{
  "ETag": "E2P4Z4VU7TY5SG",
  "FieldLevelEncryption": {
    "Id": "C3KM2WVD605UAY",
    "LastModifiedTime": "2019-12-10T21:30:18.974Z",
    "FieldLevelEncryptionConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example FLE configuration",
      "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,

```

```

    "QueryArgProfiles": {
      "Quantity": 0,
      "Items": []
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSYOCVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFieldLevelEncryption](#)」の「」を参照してください。

get-invalidation

次のコード例は、get-invalidation を使用する方法を示しています。

AWS CLI

CloudFront 無効化を取得するには

次の例では、ID を持つ I2J0I21PCUY0IK CloudFront デイストリビューションの ID を使用して無効化を取得します EDFDVBD6EXAMPLE。

```
aws cloudfront get-invalidation --id I2J0I21PCUY0IK --distribution-id EDFDVBD6EXAMPLE
```

出力:

```
{
```



```

    "Invalidation": {
      "Status": "Completed",
      "InvalidationBatch": {
        "Paths": {
          "Items": [
            "/example-path/example-file.jpg",
            "/example-path/example-file-2.jpg"
          ],
          "Quantity": 2
        },
        "CallerReference": "cli-example"
      },
      "Id": "I2J0I21PCUY0IK",
      "CreateTime": "2019-12-05T18:40:49.413Z"
    }
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInvalidation](#)」の「」を参照してください。

get-public-key-config

次のコード例は、get-public-key-config を使用する方法を示しています。

AWS CLI

CloudFront パブリックキー設定を取得するには

次の例ではKDFB19YGCR002、を含む ID を持つ CloudFront パブリックキーに関するメタデータを取得しますETag。パブリックキー ID は create-public-key および list-public-keys コマンドで返されます。

```
aws cloudfront get-public-key-config --id KDFB19YGCR002
```

出力:

```

{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKeyConfig": {
    "CallerReference": "cli-example",
    "Name": "ExampleKey",

```

```

    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBCgKCAQEAxPmBCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBaz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfyTBzVZYG2tSesplc0kjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
    "Comment": "example public key"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPublicKeyConfig](#)」の「」を参照してください。

get-public-key

次のコード例は、get-public-key を使用する方法を示しています。

AWS CLI

CloudFront パブリックキーを取得するには

次の例ではKDFB19YGCR002、を含む ID を持つ CloudFront パブリックキーを取得しますETag。パブリックキー ID は create-public-key および list-public-keys コマンドで返されます。

```
aws cloudfront get-public-key --id KDFB19YGCR002
```

出力:

```

{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKey": {
    "Id": "KDFB19YGCR002",
    "CreatedTime": "2019-12-05T18:51:43.781Z",
    "PublicKeyConfig": {
      "CallerReference": "cli-example",
      "Name": "ExampleKey",
      "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBCgKCAQEAxPmBCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBaz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb

```

```

\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLK0X5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwlIDAQAB\n-----END
PUBLIC KEY-----\n",
    "Comment": "example public key"
  }
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPublicKey](#)」の「」を参照してください。

list-cloud-front-origin-access-identities

次の例は、list-cloud-front-origin-access-identities を使用する方法を説明しています。

AWS CLI

CloudFront オリジンアクセス ID を一覧表示するには

次の例では、AWS アカウントの CloudFront オリジンアクセス ID (OAI) のリストを取得します。

```
aws cloudfront list-cloud-front-origin-access-identities
```

出力:

```

{
  "CloudFrontOriginAccessIdentityList": {
    "Items": [
      {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "Comment": "Example OAI"
      },
      {
        "Id": "EH1HDMBEXAMPLE",
        "S3CanonicalUserId":
"1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
        "Comment": "Test OAI"
      }
    ]
  }
}

```

```
    },
    {
      "Id": "E2X2C9EXAMPLE",
      "S3CanonicalUserId":
"cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
      "Comment": "Example OAI #2"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListCloudFrontOriginAccessIdentities](#)」の「」を参照してください。

list-distributions

次の例は、list-distributions を使用する方法を説明しています。

AWS CLI

CloudFront デイストリビューションを一覧表示するには

次の例では、AWS アカウントの CloudFront デイストリビューションのリストを取得します。

```
aws cloudfront list-distributions
```

出力:

```
{
  "DistributionList": {
    "Items": [
      {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/
EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-11-22T00:55:15.705Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d1111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
          "Enabled": false,
          "Quantity": 0
        }
      }
    ]
  }
}
```

```
    },
    "DistributionConfig": {
      "CallerReference": "cli-example",
      "Aliases": {
        "Quantity": 0
      },
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "awsexamplebucket.s3.amazonaws.com-cli-
example",
          "DomainName": "awsexamplebucket.s3.amazonaws.com",
          "OriginPath": "",
          "CustomHeaders": {
            "Quantity": 0
          },
          "S3OriginConfig": {
            "OriginAccessIdentity": ""
          }
        }
      ]
    },
    "OriginGroups": {
      "Quantity": 0
    },
    "DefaultCacheBehavior": {
      "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-
example",
      "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
          "Forward": "none"
        },
        "Headers": {
          "Quantity": 0
        },
        "QueryStringCacheKeys": {
          "Quantity": 0
        }
      },
      "TrustedSigners": {
        "Enabled": false,
```

```
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
```

```

        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
},
{
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/
EDFDVBD6EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d930174dauwrn8.cloudfront.net",
    "ActiveTrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "awsexamplebucket1.s3.amazonaws.com-cli-
example",
                    "DomainName": "awsexamplebucket1.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {

```

```
        "OriginAccessIdentity": ""
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket1.s3.amazonaws.com-cli-
example",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      },
      "Headers": {
        "Quantity": 0
      },
      "QueryStringCacheKeys": {
        "Quantity": 0
      }
    },
    "TrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ],
      "CachedMethods": {
        "Quantity": 2,
        "Items": [
          "HEAD",
          "GET"
        ]
      }
    },
    "SmoothStreaming": false,
```



```
        "DefaultTTL": 86400,
        "MaxTTL": 31536000,
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": ""
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
}
},
{
    "Id": "E1X5IZQEXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/
E1X5IZQEXAMPLE",
    "Status": "Deployed",
```

```
"LastModifiedTime": "2019-11-06T21:31:48.864Z",
"DomainName": "d2e04y12345678.cloudfront.net",
"Aliases": {
  "Quantity": 0
},
"Origins": {
  "Quantity": 1,
  "Items": [
    {
      "Id": "awsexamplebucket2",
      "DomainName": "awsexamplebucket2.s3.us-
west-2.amazonaws.com",
      "OriginPath": "",
      "CustomHeaders": {
        "Quantity": 0
      },
      "S3OriginConfig": {
        "OriginAccessIdentity": ""
      }
    }
  ]
},
"OriginGroups": {
  "Quantity": 0
},
"DefaultCacheBehavior": {
  "TargetOriginId": "awsexamplebucket2",
  "ForwardedValues": {
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    }
  },
  "Headers": {
    "Quantity": 0
  },
  "QueryStringCacheKeys": {
    "Quantity": 0
  }
},
"TrustedSigners": {
  "Enabled": false,
  "Quantity": 0
},
"ViewerProtocolPolicy": "allow-all",
```

```
    "MinTTL": 0,
    "AllowedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ],
      "CachedMethods": {
        "Quantity": 2,
        "Items": [
          "HEAD",
          "GET"
        ]
      }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
      "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
  },
  "CacheBehaviors": {
    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "",
  "PriceClass": "PriceClass_All",
  "Enabled": true,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  }
},
```

```

        "WebACLId": "",
        "HttpVersion": "HTTP1_1",
        "IsIPV6Enabled": true
    }
]
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDistributions](#)」の「」を参照してください。

list-field-level-encryption-configs

次の例は、list-field-level-encryption-configs を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化設定を一覧表示するには

次の例では、AWS アカウントの CloudFront フィールドレベルの暗号化設定のリストを取得します。

```
aws cloudfront list-field-level-encryption-configs
```

出力:

```

{
  "FieldLevelEncryptionList": {
    "MaxItems": 100,
    "Quantity": 1,
    "Items": [
      {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T21:30:18.974Z",
        "Comment": "Example FLE configuration",
        "QueryArgProfileConfig": {
          "ForwardWhenQueryArgProfileIsUnknown": true,
          "QueryArgProfiles": {
            "Quantity": 0,
            "Items": []
          }
        }
      }
    ],
  },
}

```

```

    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSY0CVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFieldLevelEncryptionConfigs](#)」の「」を参照してください。

list-field-level-encryption-profiles

次の例は、list-field-level-encryption-profiles を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化プロファイルを一覧表示するには

次の例では、AWS アカウントの CloudFront フィールドレベルの暗号化プロファイルのリストを取得します。

```
aws cloudfront list-field-level-encryption-profiles
```

出力:

```

{
  "FieldLevelEncryptionProfileList": {
    "MaxItems": 100,
    "Quantity": 2,
    "Items": [
      {

```

```
    "Id": "P280MFCLSY0CVU",
    "LastModifiedTime": "2019-12-05T01:05:39.896Z",
    "Name": "ExampleFLEProfile",
    "EncryptionEntities": {
      "Quantity": 1,
      "Items": [
        {
          "PublicKeyId": "K2K8NC4HVFE3M0",
          "ProviderId": "ExampleFLEProvider",
          "FieldPatterns": {
            "Quantity": 1,
            "Items": [
              "ExampleSensitiveField"
            ]
          }
        }
      ]
    },
    "Comment": "FLE profile for AWS CLI example"
  },
  {
    "Id": "PPK0U0SIF5WSV",
    "LastModifiedTime": "2019-12-10T01:03:16.537Z",
    "Name": "ExampleFLEProfile2",
    "EncryptionEntities": {
      "Quantity": 1,
      "Items": [
        {
          "PublicKeyId": "K2ABC10EXAMPLE",
          "ProviderId": "ExampleFLEProvider2",
          "FieldPatterns": {
            "Quantity": 1,
            "Items": [
              "ExampleSensitiveField2"
            ]
          }
        }
      ]
    },
    "Comment": "FLE profile #2 for AWS CLI example"
  }
]
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFieldLevelEncryptionProfiles](#)」の「」を参照してください。

list-invalidations

次の例は、list-invalidations を使用する方法を説明しています。

AWS CLI

CloudFront 無効化を一覧表示するには

次の例では、ID を持つディストリビューションの無効化 CloudFront のリストを取得しますEDFDVBD6EXAMPLE。

```
aws cloudfront list-invalidations --distribution-id EDFDVBD6EXAMPLE
```

出力:

```
{
  "InvalidationList": {
    "Marker": "",
    "Items": [
      {
        "Status": "Completed",
        "Id": "YNY2LI2BVJ4NJU",
        "CreateTime": "2019-08-31T21:15:52.042Z"
      }
    ],
    "IsTruncated": false,
    "MaxItems": 100,
    "Quantity": 1
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInvalidations](#)」の「」を参照してください。

list-public-keys

次の例は、list-public-keys を使用する方法を説明しています。

AWS CLI

CloudFront パブリックキーを一覧表示するには

次の例では、AWS アカウントの CloudFront パブリックキーのリストを取得します。

```
aws cloudfront list-public-keys
```

出力:

```
{
  "PublicKeyList": {
    "MaxItems": 100,
    "Quantity": 2,
    "Items": [
      {
        "Id": "K2K8NC4HVFE3M0",
        "Name": "ExampleKey",
        "CreatedTime": "2019-12-05T01:04:28.818Z",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAxPmbCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjm3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
        "Comment": "example public key"
      },
      {
        "Id": "K1S0LWQ2L5HTBU",
        "Name": "ExampleKey2",
        "CreatedTime": "2019-12-09T23:28:11.110Z",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAp0CAG88A8+f4dujn9Izt
\n26LxtgAkn2opGgo/NKpMiaisyw5qlg3f1gol17FV6pYN178iJg3E08JBbwt1H
+cR9\nLGSf60NDeVhm760c39Np/vWg0dsGQcRbi9WmKZeS0DqjQGzVZWqPmito3FzWV6b
\nfVY5N36U/RdbVAJm95Km+qaMY1bIdF40t72bi3IkKYV5h1B2XoDjlQ9F6ajQKyTB
\nMHa3SN8q+3ZjQ4sJJ7D1V6r4wR8jDcFVD5NckWJmngIVnk0QM37NYeoDnka0uTpu\nnha/
```



```
+3b8t0b2z3LBVHPkp85zJRA0XacSwf5rZtPYKBNFsixTa2n55k2r218m0kMC4\nUwIDAQAB\n-----END
PUBLIC KEY-----",
    "Comment": "example public key #2"
  }
]
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPublicKeys](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

CloudFront デイストリビューションのタグを一覧表示するには

次の例では、CloudFront デイストリビューションのタグのリストを取得します。

```
aws cloudfront list-tags-for-resource \
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE
```

出力:

```
{
  "Tags": {
    "Items": [
      {
        "Key": "DateCreated",
        "Value": "2019-12-04"
      },
      {
        "Key": "Name",
        "Value": "Example name"
      },
      {
        "Key": "Project",
        "Value": "Example project"
      }
    ]
  }
}
```

```
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListTagsForResource](#)」の「」を参照してください。

sign

次の例は、sign を使用する方法を説明しています。

AWS CLI

に署名するには CloudFront URL

次の例は に署名します CloudFront URL。に署名するにはURL、キーペア ID (AWS マネジメントコンソールでアクセスキー ID と呼ばれます) と、信頼できる署名者のキーペアのプライベート CloudFront キーが必要です。署名付き の詳細についてはURLs、[「Amazon デベロッパーガイド」の「署名付き Cookie URLsを使用したプライベートコンテンツの配信」](#)を参照してください。 CloudFront

```
aws cloudfront sign \
  --url https://d111111abcdef8.cloudfront.net/private-content/private-file.html \
  --key-pair-id APKAEIBAERJR2EXAMPLE \
  --private-key file://cf-signer-priv-key.pem \
  --date-less-than 2020-01-01
```

出力:

```
https://d111111abcdef8.cloudfront.net/private-content/private-
file.html?Expires=1577836800&Signature=nEXK7Kby47XKeZQKvc6pwkif6oZc-
JWSpDkH0UH7EBGGqvgurkecCbgL5VfUAXyLQuJxFwRQWscz-
owcq9KpmewCXrXQbPaJZNi9XSNwf4YKurPDQYaRQawKoeenH0GFteRf9ELK-
Bs3n1jTLjtbgzIUt7QJNKXcWr8AuUYikzGdJ4-qzx6WnxXfH~fxg4-
GGl6l2kgCpXUB6Jx6K~Y3kpV0dzUP0IqFLHANJojbhxrVejomZZ2XrquDvNUCCIbePGnR3d24UPaLXG4FK0qNEaWDIB
GNvjRJxqWf93uMobeM0iVYahb-e0KIItiQewGcm0eLZQ__&Key-Pair-Id=APKAEIBAERJR2EXAMPLE
```

- API 詳細については、「コマンドAWS CLI リファレンス で[サインイン](#)」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

CloudFront ディストリビューションにタグを付けるには

次のtag-resource例では、指定された CloudFront ディストリビューションに 2 つのタグを追加します。

```
aws cloudfront tag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tags 'Items=[{Key=Name,Value="Example name"},{Key=Project,Value="Example project"}]'
```

コマンドライン引数を使用する代わりに、次の例に示すように、JSONファイルにタグを指定できます。

```
aws cloudfront tag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tags file://tags.json
```

tags.json の内容:

```
{  
  "Items": [  
    {  
      "Key": "Name",  
      "Value": "Example name"  
    },  
    {  
      "Key": "Project",  
      "Value": "Example project"  
    }  
  ]  
}
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、`untag-resource` を使用する方法を示しています。

AWS CLI

CloudFront デイストリビューションからタグを削除するには

次の例では、コマンドライン引数を使用してデイス CloudFront トリビューションから 2 つのタグを削除します。

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys Items=Name,Project
```

コマンドライン引数を使用する代わりに、次の例に示すように、JSONファイル内のタグキーを指定できます。

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys file://tag-keys.json
```

`tag-keys.json` ファイルは、現在のフォルダにあるJSONドキュメントで、以下が含まれています。

```
{  
  "Items": [  
    "Name",  
    "Project"  
  ]  
}
```

成功した場合は、コマンドの出力はありません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-cloud-front-origin-access-identity

次のコード例は、`update-cloud-front-origin-access-identity` を使用する方法を示しています。

AWS CLI

CloudFront オリジンアクセス ID を更新するには

次の例では、オリジンアクセス ID (OAI) を ID で更新しますE74FTE3AEXAMPLE。更新できるフィールドは、OAIののみですComment。

を更新するにはOAI、OAIの ID と が必要ですETag。OAI ID は、create-cloud-front-origin-access-identity コマンドと list-cloud-front-origin-access-identities コマンドの出力で返されます。を取得するにはETag、get-cloud-front-origin-access-identity または get-cloud-front-origin-access-identity-config command を使用します。--if-match オプションを使用して、OAIの を指定しますETag。

```
aws cloudfront update-cloud-front-origin-access-identity \  
  --id E74FTE3AEXAMPLE \  
  --if-match E2QWRUHEXAMPLE \  
  --cloud-front-origin-access-identity-config \  
    CallerReference=cli-example,Comment="Example OAI Updated"
```

次の例に示すように、JSON ファイルでOAI設定を指定することで、同じことを実行できます。

```
aws cloudfront update-cloud-front-origin-access-identity \  
  --id E74FTE3AEXAMPLE \  
  --if-match E2QWRUHEXAMPLE \  
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

OAI-config.json ファイルは、現在のディレクトリにあるJSONドキュメントで、以下が含まれています。

```
{  
  "CallerReference": "cli-example",  
  "Comment": "Example OAI Updated"  
}
```

OAI 設定にコマンドライン引数とJSONファイルのどちらを指定しても、出力は同じです。

```
{  
  "ETag": "E9LHASXEXAMPLE",  
  "CloudFrontOriginAccessIdentity": {  
    "Id": "E74FTE3AEXAMPLE",
```

```
    "S3CanonicalUserId":
      "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
      "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI Updated"
      }
    }
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateCloudFrontOriginAccessIdentity](#)」の「」を参照してください。

update-distribution

次のコード例は、update-distribution を使用する方法を示しています。

AWS CLI

CloudFront デイストリビューションのデフォルトのルートオブジェクトを更新するには

次の例では、デイスCloudFront トリビューションindex.htmlのデフォルトのルートオブジェクトを ID でに更新しますEDFDVBD6EXAMPLE。

```
aws cloudfront update-distribution --id EDFDVBD6EXAMPLE \
  --default-root-object index.html
```

出力:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:55:39.870Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    "DistributionConfig": {
```

```
"CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
"Aliases": {
  "Quantity": 0
},
"DefaultRootObject": "index.html",
"Origins": {
  "Quantity": 1,
  "Items": [
    {
      "Id": "example-website",
      "DomainName": "www.example.com",
      "OriginPath": "",
      "CustomHeaders": {
        "Quantity": 0
      },
      "CustomOriginConfig": {
        "HTTPPort": 80,
        "HTTPSPort": 443,
        "OriginProtocolPolicy": "match-viewer",
        "OriginSslProtocols": {
          "Quantity": 2,
          "Items": [
            "SSLv3",
            "TLSv1"
          ]
        },
        "OriginReadTimeout": 30,
        "OriginKeepaliveTimeout": 5
      }
    }
  ]
},
"OriginGroups": {
  "Quantity": 0
},
"DefaultCacheBehavior": {
  "TargetOriginId": "example-website",
  "ForwardedValues": {
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    }
  },
  "Headers": {
    "Quantity": 1,
```

```
        "Items": [
            "*"
        ]
    },
    "QueryStringCacheKeys": {
        "Quantity": 0
    }
},
"TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
},
"ViewerProtocolPolicy": "allow-all",
"MinTTL": 0,
"AllowedMethods": {
    "Quantity": 2,
    "Items": [
        "HEAD",
        "GET"
    ],
    "CachedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ]
    }
},
"SmoothStreaming": false,
"DefaultTTL": 86400,
"MaxTTL": 31536000,
"Compress": false,
"LambdaFunctionAssociations": {
    "Quantity": 0
},
"FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponse": {
    "Quantity": 0
},
"Comment": ""
```



```
    "Logging": {
      "Enabled": false,
      "IncludeCookies": false,
      "Bucket": "",
      "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http1.1",
    "IsIPV6Enabled": true
  }
}
```

CloudFront デイストリビューションを更新するには

次の例では、という名前のJSONファイルに CloudFront デイストリビューション設定を指定EMLARXS9EXAMPLEすることで、ID を持つデイストリビューションを無効にしますdist-config-disable.json。デイストリビューションを更新するには、--if-match オプションを使用してデイストリビューションの ETag を指定する必要があります。を取得するにはETag、get-distribution または get-distribution-config コマンドを使用します。

次の例を使用してデイストリビューションを無効にした後は、delete-distribution コマンドを使用してデイストリビューションを削除できます。

```
aws cloudfront update-distribution \  
  --id EMLARXS9EXAMPLE \  
  --if-match E2QWRUHEXAMPLE \  
  --distribution-config file://dist-config-disable.json
```

dist-config-disable.json ファイルは、現在のフォルダにあるJSONドキュメントで、以下が含まれています。Enabled フィールドが false に設定されていることに注意してください。

```
{
  "CallerReference": "cli-1574382155-496510",
  "Aliases": {
    "Quantity": 0
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        }
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      },
      "Headers": {
        "Quantity": 0
      },
      "QueryStringCacheKeys": {
        "Quantity": 0
      }
    },
    "TrustedSigners": {
```

```
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": false,
"ViewerCertificate": {
```

```

    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  },
  "WebACLIId": "",
  "HttpVersion": "http2",
  "IsIPV6Enabled": true
}

```

出力:

```

{
  "ETag": "E9LHASXEXAMPLE",
  "Distribution": {
    "Id": "EMLARXS9EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:32:35.553Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  },
  "DistributionConfig": {
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
      "Quantity": 0
    }
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",

```

```
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        }
    ]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId":
"awsexamplebucket.s3.amazonaws.com-1574382155-273939",
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        },
        "Headers": {
            "Quantity": 0
        },
        "QueryStringCacheKeys": {
            "Quantity": 0
        }
    },
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    }
}
```

```
        ]
      }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
      "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
  },
  "CacheBehaviors": {
    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "",
  "Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
  },
  "PriceClass": "PriceClass_All",
  "Enabled": false,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  },
  "WebACLId": "",
  "HttpVersion": "http2",
  "IsIPV6Enabled": true
}
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDistribution](#)」の「」を参照してください。

update-field-level-encryption-config

次の例は、update-field-level-encryption-config を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化設定を更新するには

次の例では、JSONファイル内のパラメータを指定C3KM2WVD605UAYすることで、フィールドレベルの暗号化設定のCommentフィールドを ID で更新します。

フィールドレベルの暗号化設定を更新するには、設定の ID と `--if-match` が必要ですETag。ID は create-field-level-encryption-config コマンドと list-field-level-encryption-configs コマンドの出力で返されます。を取得するにはETag、 get-field-level-encryption または get-field-level-encryption-config コマンドを使用します。--if-match オプションを使用して、設定の を指定しますETag。

```
aws cloudfront update-field-level-encryption-config \  
  --id C3KM2WVD605UAY \  
  --if-match E2P4Z4VU7TY5SG \  
  --field-level-encryption-config file://fle-config.json
```

fle-config.json ファイルは、現在のディレクトリにあるJSONドキュメントで、以下が含まれています。

```
{  
  "CallerReference": "cli-example",  
  "Comment": "Updated example FLE configuration",  
  "QueryArgProfileConfig": {  
    "ForwardWhenQueryArgProfileIsUnknown": true,  
    "QueryArgProfiles": {  
      "Quantity": 0  
    }  
  },  
  "ContentTypeProfileConfig": {  
    "ForwardWhenContentTypeIsUnknown": true,  
    "ContentTypeProfiles": {
```

```

    "Quantity": 1,
    "Items": [
      {
        "Format": "URLEncoded",
        "ProfileId": "P280MFCLSYOCVU",
        "ContentType": "application/x-www-form-urlencoded"
      }
    ]
  }
}

```

出力:

```

{
  "ETag": "E26M4BIAV81ZF6",
  "FieldLevelEncryption": {
    "Id": "C3KM2WVD605UAY",
    "LastModifiedTime": "2019-12-10T22:26:26.170Z",
    "FieldLevelEncryptionConfig": {
      "CallerReference": "cli-example",
      "Comment": "Updated example FLE configuration",
      "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
          "Quantity": 0,
          "Items": []
        }
      }
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSYOCVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}

```



```
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateFieldLevelEncryptionConfig](#)」の「」を参照してください。

update-field-level-encryption-profile

次の例は、update-field-level-encryption-profile を使用する方法を説明しています。

AWS CLI

CloudFront フィールドレベルの暗号化プロファイルを更新するには

次の例では、フィールドレベルの暗号化プロファイルを更新します PPK0U0SIF5WSV。この例では Comment、プロファイル Name と を更新し、JSON ファイルにパラメータを指定して 2 番目の FieldPatterns 項目を追加します。

フィールドレベルの暗号化プロファイルを更新するには、プロファイルの ID と が必要で ETag。ID は create-field-level-encryption-profile コマンドと list-field-level-encryption-profiles コマンドの出力で返されます。を取得するには ETag、get-field-level-encryption-profile コマンドまたは get-field-level-encryption-profile-config コマンドを使用します。--if-match オプションを使用して、プロファイルの を指定します ETag。

```
aws cloudfront update-field-level-encryption-profile \
  --id PPK0U0SIF5WSV \
  --if-match E1QQG65FS2L2GC \
  --field-level-encryption-profile-config file://fle-profile-config.json
```

file-profile-config.json ファイルは、現在のディレクトリにある JSON ドキュメントで、以下が含まれています。

```
{
  "Name": "ExampleFLEProfileUpdated",
  "CallerReference": "cli-example",
  "Comment": "Updated FLE profile for AWS CLI example",
  "EncryptionEntities": {
    "Quantity": 1,
    "Items": [
      {
        "PublicKeyId": "K2K8NC4HVFE3M0",
```

```

        "ProviderId": "ExampleFLEProvider",
        "FieldPatterns": {
            "Quantity": 2,
            "Items": [
                "ExampleSensitiveField",
                "SecondExampleSensitiveField"
            ]
        }
    ]
}

```

出力:

```

{
  "ETag": "EJETYFJ9CL66D",
  "FieldLevelEncryptionProfile": {
    "Id": "PPK0UOSIF5WSV",
    "LastModifiedTime": "2019-12-10T19:05:58.296Z",
    "FieldLevelEncryptionProfileConfig": {
      "Name": "ExampleFLEProfileUpdated",
      "CallerReference": "cli-example",
      "Comment": "Updated FLE profile for AWS CLI example",
      "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
          {
            "PublicKeyId": "K2K8NC4HVFE3M0",
            "ProviderId": "ExampleFLEProvider",
            "FieldPatterns": {
              "Quantity": 2,
              "Items": [
                "ExampleSensitiveField",
                "SecondExampleSensitiveField"
              ]
            }
          }
        ]
      }
    }
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFieldLevelEncryptionProfile](#)」の「」を参照してください。

を使用した Amazon CloudSearch の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CloudSearch。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

upload-documents

次の例は、upload-documents を使用する方法を説明しています。

AWS CLI

次のupload-documentsコマンドは、JSONドキュメントのバッチを Amazon CloudSearch ドメインにアップロードします。

```
aws cloudsearchdomain upload-documents --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com --content-type application/json --documents document-batch.json
```

出力:

```
{
  "status": "success",
  "adds": 5000,
```

```
"deletes": 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadDocuments](#)」の「」を参照してください。

CloudTrail を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CloudTrail。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags

次のコード例は、add-tags を使用する方法を示しています。

AWS CLI

証跡にタグを追加するには

次のadd-tagsコマンドは、のタグを追加しますTrail1。

```
aws cloudtrail add-tags --resource-id arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --tags-list Key=name,Value=Alice Key=location,Value=us
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTags](#)」の「」を参照してください。

create-subscription

次の例は、create-subscription を使用する方法を説明しています。

AWS CLI

証跡の AWS リソースを作成して設定するには

次のcreate-subscriptionコマンドは、の新しい S3 バケットとSNSトピックを作成しますTrail1。

```
aws cloudtrail create-subscription --name Trail1 --s3-new-bucket my-bucket --sns-new-topic my-topic
```

出力:

```
Setting up new S3 bucket my-bucket...
Setting up new SNS topic my-topic...
Creating/updating CloudTrail configuration...
CloudTrail configuration:
{
  "trailList": [
    {
      "IncludeGlobalServiceEvents": true,
      "Name": "Trail1",
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
      "LogFileValidationEnabled": false,
      "IsMultiRegionTrail": false,
      "S3BucketName": "my-bucket",
      "SnsTopicName": "my-topic",
      "HomeRegion": "us-east-1"
    }
  ],
  "ResponseMetadata": {
    "HTTPStatusCode": 200,
    "RequestId": "f39e51f6-c615-11e5-85bd-d35ca21ee3e2"
  }
}
Starting CloudTrail service...
Logs will be delivered to my-bucket
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubscription](#)」の「」を参照してください。

create-trail

次の例は、create-trail を使用する方法を説明しています。

AWS CLI

証跡を作成するには

次のcreate-trailコマンドは、 という名前のマルチリージョン証跡を作成しTrail1、S3 バケットを指定します。

```
aws cloudtrail create-trail --name Trail1 --s3-bucket-name my-bucket --is-multi-region-trail
```

出力:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[CreateTrail](#)」の「」を参照してください。

delete-trail

次の例は、delete-trail を使用する方法を説明しています。

AWS CLI

証跡を削除するには

次のdelete-trailコマンドは、 という名前の証跡を削除しますTrail1。

```
aws cloudtrail delete-trail --name Trail1
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[DeleteTrail](#)」の「」を参照してください。

describe-trails

次の例は、describe-trails を使用する方法を説明しています。

AWS CLI

証跡を記述するには

次のdescribe-trailsコマンドは、Trail1と の設定を返しますTrail2。

```
aws cloudtrail describe-trails --trail-name-list Trail1 Trail2
```

出力:

```
{
  "trailList": [
    {
      "IncludeGlobalServiceEvents": true,
      "Name": "Trail1",
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
      "LogFileValidationEnabled": false,
      "IsMultiRegionTrail": false,
      "S3BucketName": "my-bucket",
      "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/
CloudTrail_CloudWatchLogs_Role",
      "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-
group:CloudTrail:*",
      "SnsTopicName": "my-topic",
      "HomeRegion": "us-east-1"
    },
    {
      "IncludeGlobalServiceEvents": true,
      "Name": "Trail2",
      "S3KeyPrefix": "my-prefix",
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
      "LogFileValidationEnabled": false,
      "IsMultiRegionTrail": false,
      "S3BucketName": "my-bucket",
      "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/4c5ae5ac-3c13-421e-8335-c7868ef6a769",
      "HomeRegion": "us-east-1"
    }
  ]
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrails](#)」の「」を参照してください。

get-event-selectors

次のコード例は、get-event-selectors を使用する方法を示しています。

AWS CLI

証跡のイベントセレクタ設定を表示するには

次のget-event-selectorsコマンドは、 の設定を返しますTrail1。

```
aws cloudtrail get-event-selectors --trail-name Trail1
```

出力:

```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [],
      "ReadWriteType": "All"
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEventSelectors](#)」の「」を参照してください。

get-trail-status

次の例は、get-trail-status を使用する方法を説明しています。

AWS CLI

証跡のステータスを取得するには

次のget-trail-statusコマンドは、 の配信とログ記録の詳細を返しますTrail1。

```
aws cloudtrail get-trail-status --name Trail1
```

出力:

```
{
  "LatestNotificationTime": 1454022144.869,
  "LatestNotificationAttemptSucceeded": "2016-01-28T23:02:24Z",
  "LatestDeliveryAttemptTime": "2016-01-28T23:02:24Z",
  "LatestDeliveryTime": 1454022144.869,
  "TimeLoggingStarted": "2015-11-06T18:36:38Z",
  "LatestDeliveryAttemptSucceeded": "2016-01-28T23:02:24Z",
  "IsLogging": true,
  "LatestCloudWatchLogsDeliveryTime": 1454022144.918,
  "StartLoggingTime": 1446834998.695,
  "StopLoggingTime": 1446834996.933,
  "LatestNotificationAttemptTime": "2016-01-28T23:02:24Z",
  "TimeLoggingStopped": "2015-11-06T18:36:36Z"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTrailStatus](#)」の「」を参照してください。

list-public-keys

次のコード例は、list-public-keys を使用する方法を示しています。

AWS CLI

証跡のすべてのパブリックキーを一覧表示するには

次のlist-public-keysコマンドは、指定された時間範囲内でダイジェストファイルに署名するためにプライベートキーが使用されたすべてのパブリックキーを返します。

```
aws cloudtrail list-public-keys --start-time 2016-01-01T20:30:00.000Z
```

出力:

```
{
  "PublicKeyList": [
```

```

    {
      "ValidityStartTime": 1453076702.0,
      "ValidityEndTime": 1455668702.0,
      "Value": "MIIBCgKCAQEAlSS3cl92HDycr/MTj0mo0has8habjrraXw+Kz1WF0axSI2tcF
+3iJ9BKQAVSKxGwxwu3m0wG3J
+kU11xboEcEPHYoIYMbgfSw7KGnuDKwkLzsQWhUJ0cIb0HASox1vv/5fNXkrHhGbDCHeVXm804c83nvHUEFYThr1PfyP
+4WGDk+BGH5m9iuiAKkipEHWmU18/P7XpfpWQuk4h8g3pXZ0rNXr081bh4d39svj7Uqdhv0XoBISp9t/
EXYuePGEtBdrKD9Dz+VHwyUPtBQvYr9BnkF88qBnaPNhS44rzwIDAQAB",
      "Fingerprint": "7f3f401420072e50a65a141430817ab3"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListPublicKeys](#)」の「」を参照してください。

list-tags

次の例は、list-tags を使用する方法を説明しています。

AWS CLI

証跡のタグを一覧表示するには

次のlist-tagsコマンドは、Trail1と のタグを一覧表示しますTrail2。

```

aws cloudtrail list-tags --resource-id-list arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2

```

出力:

```

{
  "ResourceTagList": [
    {
      "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",
      "TagsList": [
        {
          "Value": "Alice",
          "Key": "name"
        },
        {

```

```
        "Value": "us",
        "Key": "location"
      }
    ],
  },
  {
    "ResourceId": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail2",
    "TagsList": [
      {
        "Value": "Bob",
        "Key": "name"
      }
    ]
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTags](#)」の「」を参照してください。

lookup-events

次の例は、lookup-events を使用する方法を説明しています。

AWS CLI

証跡のイベントを検索するには

次のlookup-eventsコマンドは、属性 でAPIアクティビティイベントを検索しますEventName。

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=ConsoleLogin
```

出力:

```
{
  "Events": [
    {
      "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
      "Username": "my-session-name",
      "EventTime": "2021-11-18T09:41:02-08:00",
```

```

    "CloudTrailEvent": "{ \"eventVersion\": \"1.02\", \"userIdentity\": { \"type\": \"AssumedRole\", \"principalId\": \"AROAJIKPFTA72SWU4L7T4:my-session-name\", \"arn\": \"arn:aws:sts::123456789012:assumed-role/my-role/my-session-name\", \"accountId\": \"123456789012\", \"sessionContext\": { \"attributes\": { \"mfaAuthenticated\": \"false\", \"creationDate\": \"2016-01-26T21:42:12Z\" }, \"sessionIssuer\": { \"type\": \"Role\", \"principalId\": \"AROAJIKPFTA72SWU4L7T4\", \"arn\": \"arn:aws:iam::123456789012:role/my-role\", \"accountId\": \"123456789012\", \"userName\": \"my-role\" } } }, \"eventTime\": \"2016-01-26T21:42:12Z\", \"eventSource\": \"signin.amazonaws.com\", \"eventName\": \"ConsoleLogin\", \"awsRegion\": \"us-east-1\", \"sourceIPAddress\": \"72.21.198.70\", \"userAgent\": \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36\", \"requestParameters\": null, \"responseElements\": { \"ConsoleLogin\": \"Success\" }, \"additionalEventData\": { \"MobileVersion\": \"No\", \"MFAUsed\": \"No\" }, \"eventID\": \"654ccbc0-ba0d-486a-9076-dbf7274677a7\", \"eventType\": \"AwsConsoleSignIn\", \"recipientAccountId\": \"123456789012\" }",
    "EventName": "ConsoleLogin",
    "Resources": []
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [LookupEvents](#)」の「」を参照してください。

put-event-selectors

次のコード例は、put-event-selectors を使用する方法を示しています。

AWS CLI

例 1: アドバンスイベントセレクタを使用して管理イベントとデータイベントを記録する証跡を設定する

アドバンスイベントセレクタとアドバンスイベントセレクタの条件を追加でき、証跡のすべての条件とセレクタに最大 500 個の値を追加できます。アドバンスイベントセレクタを使用して、使用可能なすべてのデータイベントタイプをログに記録できます。高度なイベントセレクターまたは基本的なイベントセレクターのいずれかを使用できますが、両方を使用することはできません。高度なイベントセレクターを証跡に適用すると、既存の基本的なイベントセレクターは上書きされます。

次の例では、という名前の証跡のアドバンスイベントセレクタを作成し、すべての管理イベント myTrail をログに記録し、S3 PutObject を DeleteObject API ログに記録し、1 つの S3 バケッ

トを除くすべての を呼び出し、 という名前の Lambda 関数のデータAPI呼び出しをログに記録しmyFunction、 という名前のSNSトピックのAPI呼び出しを公開しますmyTopic。

```
aws cloudtrail put-event-selectors \
  --trail-name myTrail \
  --advanced-event-selectors '[{"Name": "Log all management events",
  "FieldSelectors": [{"Field": "eventCategory", "Equals": ["Management"]} ] },
{"Name": "Log PutObject and DeleteObject events for all but one
bucket", "FieldSelectors": [{"Field": "eventCategory", "Equals": ["Data"]} ],
{ "Field": "resources.type", "Equals": ["AWS::S3::Object"]} ],{ "Field":
"eventName", "Equals": ["PutObject","DeleteObject"]} ],{ "Field": "resources.ARN",
"NotStartsWith": ["arn:aws:s3:::sample_bucket_name/"]} ]}],{"Name": "Log
data events for a specific Lambda function", "FieldSelectors": [{"Field":
"eventCategory", "Equals": ["Data"]} ],{ "Field": "resources.type",
"Equals": ["AWS::Lambda::Function"]} ],{ "Field": "resources.ARN", "Equals":
["arn:aws:lambda:us-east-1:123456789012:function:myFunction"]} ]}],{"Name":
"Log all Publish API calls on a specific SNS topic", "FieldSelectors":
[{"Field": "eventCategory", "Equals": ["Data"]} ],{ "Field": "resources.type",
"Equals": ["AWS::SNS::Topic"]} ],{ "Field": "eventName", "Equals":
["Publish"]} ],{ "Field": "resources.ARN", "Equals": ["arn:aws:sns:us-
east-1:123456789012:myTopic.fifo"]} ]}]'
```

出力:

```
{
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/myTrail",
  "AdvancedEventSelectors": [
    {
      "Name": "Log all management events",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
          "Equals": [
            "Management"
          ]
        }
      ]
    },
    {
      "Name": "Log PutObject and DeleteObject events for all but one bucket",
      "FieldSelectors": [
        {
          "Field": "eventCategory",
```

```
        "Equals": [
            "Data"
        ]
    },
    {
        "Field": "resources.type",
        "Equals": [
            "AWS::S3::Object"
        ]
    },
    {
        "Field": "eventName",
        "Equals": [
            "PutObject",
            "DeleteObject"
        ]
    },
    {
        "Field": "resources.ARN",
        "NotStartsWith": [
            "arn:aws:s3:::sample_bucket_name/"
        ]
    }
]
},
{
    "Name": "Log data events for a specific Lambda function",
    "FieldSelectors": [
        {
            "Field": "eventCategory",
            "Equals": [
                "Data"
            ]
        },
        {
            "Field": "resources.type",
            "Equals": [
                "AWS::Lambda::Function"
            ]
        },
        {
            "Field": "resources.ARN",
            "Equals": [
                "arn:aws:lambda:us-east-1:123456789012:function:myFunction"
            ]
        }
    ]
}
```

```

    ]
  }
]
},
{
  "Name": "Log all Publish API calls on a specific SNS topic",
  "FieldSelectors": [
    {
      "Field": "eventCategory",
      "Equals": [
        "Data"
      ]
    },
    {
      "Field": "resources.type",
      "Equals": [
        "AWS::SNS::Topic"
      ]
    },
    {
      "Field": "eventName",
      "Equals": [
        "Publish"
      ]
    },
    {
      "Field": "resources.ARN",
      "Equals": [
        "arn:aws:sns:us-east-1:123456789012:myTopic.fifo"
      ]
    }
  ]
}
]
}

```

詳細については、AWS CloudTrail 「ユーザーガイド」の [「アドバンスドイベントセレクタを使用してイベントをログに記録する」](#) を参照してください。

例 2: すべての管理イベントとデータイベントを記録する証跡のイベントセレクタを設定する

証跡あたり最大 5 つのイベントセレクタと、証跡あたり最大 250 のデータリソースを設定できます。イベントセレクタは、基本的なイベントセレクタとも呼ばれます。イベントセレクタを使

用して、S3 オブジェクト、Lambda 関数、および DynamoDB テーブルの管理イベントとデータイベントを記録できます。他のリソースタイプのデータイベントをログに記録するには、アドバンスドイベントセレクタを使用する必要があります。

次の例では、`TrailName` という名前の証跡のイベントセレクタを作成し、すべての管理イベント、2 つの Amazon S3 バケット/プレフィックスの組み合わせのデータイベント、および `TrailName` という名前の単一の AWS Lambda 関数のデータイベント `TrailName` を含めます `hello-world-python-function`。

```
aws cloudtrail put-event-selectors \  
  --trail-name TrailName \  
  --event-selectors '[{"ReadWriteType": "All", "IncludeManagementEvents":  
    true, "DataResources": [{"Type": "AWS::S3::Object", "Values":  
      ["arn:aws:s3:::mybucket/prefix", "arn:aws:s3:::mybucket2/prefix2"]},  
    {"Type": "AWS::Lambda::Function", "Values": ["arn:aws:lambda:us-  
west-2:999999999999:function:hello-world-python-function"]}]]'
```

出力:

```
{  
  "EventSelectors": [  
    {  
      "IncludeManagementEvents": true,  
      "DataResources": [  
        {  
          "Values": [  
            "arn:aws:s3:::mybucket/prefix",  
            "arn:aws:s3:::mybucket2/prefix2"  
          ],  
          "Type": "AWS::S3::Object"  
        },  
        {  
          "Values": [  
            "arn:aws:lambda:us-west-2:123456789012:function:hello-world-  
python-function"  
          ],  
          "Type": "AWS::Lambda::Function"  
        },  
      ],  
      "ReadWriteType": "All"  
    }  
  ],  
}
```



```
"TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName"
}
```

詳細については、AWS CloudTrail 「ユーザーガイド」の [「基本的なイベントセレクタを使用してイベントをログに記録する」](#) を参照してください。

例 3: 管理イベント、S3 オブジェクト上のすべての S3 データイベント、およびアカウントの関数上のすべての Lambda データイベントを記録する証跡のイベントセレクタを設定する

次の例では、アカウント内のすべての Amazon S3 バケットと AWS Lambda 関数のすべての管理イベントとすべてのデータイベント TrailName2 を含む という名前の証跡のイベントセレクタを作成します AWS。

```
aws cloudtrail put-event-selectors \
  --trail-name TrailName2 \
  --event-selectors '[{"ReadWriteType": "All", "IncludeManagementEvents":
true, "DataResources": [{"Type": "AWS::S3::Object", "Values": ["arn:aws:s3"]},
{"Type": "AWS::Lambda::Function", "Values": ["arn:aws:lambda"]}]]'
```

出力:

```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [
        {
          "Values": [
            "arn:aws:s3"
          ],
          "Type": "AWS::S3::Object"
        },
        {
          "Values": [
            "arn:aws:lambda"
          ],
          "Type": "AWS::Lambda::Function"
        }
      ],
      "ReadWriteType": "All"
    }
  ],
}
```

```
"TrailARN": "arn:aws:cloudtrail:us-east-2:123456789012:trail/TrailName2"
}
```

詳細については、AWS CloudTrail 「ユーザーガイド」の [「基本的なイベントセレクタを使用してイベントをログに記録する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutEventSelectors](#)」の「」を参照してください。

remove-tags

次のコード例は、remove-tags を使用する方法を示しています。

AWS CLI

証跡のタグを削除するには

次のremove-tagsコマンドは、 に指定されたタグを削除しますTrail1。

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveTags](#)」の「」を参照してください。

start-logging

次の例は、start-logging を使用する方法を説明しています。

AWS CLI

証跡のログ記録を開始するには

次のstart-loggingコマンドは、 のログ記録を有効にしますTrail1。

```
aws cloudtrail start-logging --name Trail1
```

- API 詳細については、AWS CLI 「コマンドリファレンス [StartLogging](#)」の「」を参照してください。

stop-logging

次のコード例は、stop-logging を使用する方法を示しています。

AWS CLI

証跡のログ記録を停止するには

次のstop-loggingコマンドは、 のログ記録をオフにしますTrail1。

```
aws cloudtrail stop-logging --name Trail1
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StopLogging](#)」の「」を参照してください。

update-subscription

次の例は、update-subscription を使用する方法を説明しています。

AWS CLI

証跡の設定を更新するには

次のupdate-subscriptionコマンドは、証跡を更新して新しい S3 バケットとSNSトピックを指定します。

```
aws cloudtrail update-subscription --name Trail1 --s3-new-bucket my-bucket-new --  
sns-new-topic my-topic-new
```

出力:

```
Setting up new S3 bucket my-bucket-new...  
Setting up new SNS topic my-topic-new...  
Creating/updating CloudTrail configuration...  
CloudTrail configuration:  
{  
  "trailList": [  
    {  
      "IncludeGlobalServiceEvents": true,  
      "Name": "Trail1",  
      "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1",  
      "LogFileValidationEnabled": false,
```

```
    "IsMultiRegionTrail": false,
    "S3BucketName": "my-bucket-new",
    "SnsTopicName": "my-topic-new",
    "HomeRegion": "us-east-1"
  }
],
"ResponseMetadata": {
  "HTTPStatusCode": 200,
  "RequestId": "31126f8a-c616-11e5-9cc6-2fd637936879"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSubscription](#)」の「」を参照してください。

update-trail

次のコード例は、update-trail を使用する方法を示しています。

AWS CLI

証跡を更新するには

次のupdate-trailコマンドは、ログ配信に既存のバケットを使用するように証跡を更新します。

```
aws cloudtrail update-trail --name Trail1 --s3-bucket-name my-bucket
```

出力:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTrail](#)」の「」を参照してください。

validate-logs

次のコード例は、`validate-logs` を使用する方法を示しています。

AWS CLI

ログファイルを検証するには

次の`validate-logs`コマンドは、 のログを検証しますTrail1。

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --start-time 20160129T19:00:00Z
```

出力:

```
Validating log files for trail arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 between 2016-01-29T19:00:00Z and 2016-01-29T22:15:43Z
Results requested for 2016-01-29T19:00:00Z to 2016-01-29T22:15:43Z
Results found for 2016-01-29T19:24:57Z to 2016-01-29T21:24:57Z:
3/3 digest files valid
15/15 log files valid
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ValidateLogs](#)」の「」を参照してください。

CloudWatch を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CloudWatch。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-alarms

次のコード例は、delete-alarms を使用する方法を示しています。

AWS CLI

アラームを削除するには

次の例では、delete-alarms コマンドを使用して「myalarm」という名前の Amazon CloudWatch アラームを削除します。

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

出力:

```
This command returns to the prompt if successful.
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAlarms](#)」の「」を参照してください。

describe-alarm-history

次のコード例は、describe-alarm-history を使用する方法を示しています。

AWS CLI

アラームの履歴を取得するには

次の例では、describe-alarm-history コマンドを使用して「myalarm」という名前の Amazon CloudWatch アラームの履歴を取得します。

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type StateUpdate
```

出力:

```
{
  "AlarmHistoryItems": [
    {
```

```

    "Timestamp": "2014-04-09T18:59:06.442Z",
    "HistoryItemType": "StateUpdate",
    "AlarmName": "myalarm",
    "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue\":
\\\"ALARM\\\",\\\"stateReason\\\":\\\"testing purposes\\\"},\\\"newState\\\":{\\\"stateValue\\\":\\\"OK
\\\",\\\"stateReason\\\":\\\"Threshold Crossed: 2 datapoints were not greater than the
threshold (70.0). The most recent datapoints: [38.958, 40.292].\\\",\\\"stateReasonData
\\\":{\\\"version\\\":\\\"1.0\\\",\\\"queryDate\\\":\\\"2014-04-09T18:59:06.419+0000\\\",\\\"startDate
\\\":\\\"2014-04-09T18:44:00.000+0000\\\",\\\"statistic\\\":\\\"Average\\\",\\\"period\\\":300,
\\\"recentDatapoints\\\":[38.958,40.292],\\\"threshold\\\":70.0}}}\",
    "HistorySummary": "Alarm updated from ALARM to OK"
  },
  {
    "Timestamp": "2014-04-09T18:59:05.805Z",
    "HistoryItemType": "StateUpdate",
    "AlarmName": "myalarm",
    "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\\\":\\\"OK\\\",\\\"stateReason\\\":\\\"Threshold Crossed: 2 datapoints were
not greater than the threshold (70.0). The most recent datapoints:
[38.839999999999996, 39.714].\\\",\\\"stateReasonData\\\":{\\\"version\\\":
\\\"1.0\\\",\\\"queryDate\\\":\\\"2014-03-11T22:45:41.569+0000\\\",\\\"startDate\\\":
\\\"2014-03-11T22:30:00.000+0000\\\",\\\"statistic\\\":\\\"Average\\\",\\\"period\\\":300,
\\\"recentDatapoints\\\":[38.839999999999996,39.714],\\\"threshold\\\":70.0}},\\\"newState\\\":
{\\\"stateValue\\\":\\\"ALARM\\\",\\\"stateReason\\\":\\\"testing purposes\\\"}}\",
    "HistorySummary": "Alarm updated from OK to ALARM"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAlarmHistory](#)」の「」を参照してください。

describe-alarms-for-metric

次の例は、describe-alarms-for-metric を使用する方法を説明しています。

AWS CLI

メトリクスに関連するアラームについての情報を表示するには

次の例では、describe-alarms-for-metric コマンドを使用して、Amazon EC2CPUUtilizationメトリクスと ID i-0c986c72 のインスタンスに関連付けられたアラームに関する情報を表示します。

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

出力:

```
{
  "MetricAlarms": [
    {
      "EvaluationPeriods": 10,
      "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
      "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
      "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
      "ComparisonOperator": "GreaterThanOrEqualToThreshold",
      "AlarmActions": [
        "arn:aws:sns:us-east-1:111122223333:NotifyMe"
      ],
      "Namespace": "AWS/EC2",
      "AlarmDescription": "CPU usage exceeds 70 percent",
      "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
      "Period": 300,
      "StateValue": "OK",
      "Threshold": 70.0,
      "AlarmName": "myHighCpuAlarm2",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-0c986c72"
        }
      ],
      "Statistic": "Average",
      "StateReason": "Threshold Crossed: 10 datapoints were not greater than
or equal to the threshold (70.0). The most recent datapoints: [40.760000000000005,
41.316].",
      "InsufficientDataActions": [],
      "OKActions": [],
      "ActionsEnabled": true,
      "MetricName": "CPUUtilization"
    },
  ],
}
```



```

    {
      "EvaluationPeriods": 2,
      "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
      "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
      "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
      "ComparisonOperator": "GreaterThanThreshold",
      "AlarmActions": [
        "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
      ],
      "Namespace": "AWS/EC2",
      "AlarmDescription": "CPU usage exceeds 70 percent",
      "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\\\"2014-04-09T18:59:06.419+0000\\\",\\\"startDate\\\":\\\"2014-04-09T18:44:00.000+0000\\\",
\\\"statistic\\\":\\\"Average\\\",\\\"period\\\":300,\\\"recentDatapoints\\\":[38.958,40.292],
\\\"threshold\\\":70.0}\",
      "Period": 300,
      "StateValue": "OK",
      "Threshold": 70.0,
      "AlarmName": "myHighCpuAlarm",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-0c986c72"
        }
      ],
      "Statistic": "Average",
      "StateReason": "Threshold Crossed: 2 datapoints were not greater than
the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
      "InsufficientDataActions": [],
      "OKActions": [],
      "ActionsEnabled": false,
      "MetricName": "CPUUtilization"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAlarmsForMetric](#)」の「」を参照してください。

describe-alarms

次のコード例は、describe-alarms を使用する方法を示しています。

AWS CLI

アラームに関する情報を一覧表示するには

次の例は、`describe-alarms` コマンドを使用して、「myalarm」という名前のアラームに関する情報を表示します。

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

出力:

```
{
  "MetricAlarms": [
    {
      "EvaluationPeriods": 2,
      "AlarmArn": "arn:aws:cloudwatch:us-east-1:123456789012:alarm:myalarm",
      "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
      "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
      "ComparisonOperator": "GreaterThanThreshold",
      "AlarmActions": [
        "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
      ],
      "Namespace": "AWS/EC2",
      "AlarmDescription": "CPU usage exceeds 70 percent",
      "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":\
\\\"2014-04-09T18:59:06.419+0000\\\", \"startDate\": \"2014-04-09T18:44:00.000+0000\\\",
\\\"statistic\": \"Average\\\", \"period\": 300, \"recentDatapoints\": [38.958, 40.292],
\\\"threshold\": 70.0}\",
      "Period": 300,
      "StateValue": "OK",
      "Threshold": 70.0,
      "AlarmName": "myalarm",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-0c986c72"
        }
      ],
      "Statistic": "Average",
      "StateReason": "Threshold Crossed: 2 datapoints were not greater than
the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
      "InsufficientDataActions": [],
      "OKActions": []
    }
  ]
}
```

```
        "ActionsEnabled": true,  
        "MetricName": "CPUUtilization"  
    }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAlarms](#)」の「」を参照してください。

disable-alarm-actions

次の例は、`disable-alarm-actions` を使用する方法を説明しています。

AWS CLI

アラームのアクションを無効化するには

次の例は、`disable-alarm-actions` コマンドを使用して、「myalarm」という名前のアラームのアクションをすべて無効化します。

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

正常に完了すると、このコマンドはプロンプトに戻ります。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableAlarmActions](#)」の「」を参照してください。

enable-alarm-actions

次のコード例は、`enable-alarm-actions` を使用する方法を示しています。

AWS CLI

アラームのすべてのアクションを有効化するには

次の例は、`enable-alarm-actions` コマンドを使用して、「myalarm」という名前のアラームのアクションをすべて有効化します。

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

正常に完了すると、このコマンドはプロンプトに戻ります。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableAlarmActions](#)」の「」を参照してください。

get-metric-statistics

次の例は、get-metric-statistics を使用する方法を説明しています。

AWS CLI

EC2インスタンスあたりのCPU使用率を取得するには

次の例では、get-metric-statistics コマンドを使用して、ID i-abcdef を持つEC2インスタンスのCPU使用率を取得します。

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --namespace AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

出力:

```
{
  "Datapoints": [
    {
      "Timestamp": "2014-04-09T11:18:00Z",
      "Maximum": 44.79,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T20:18:00Z",
      "Maximum": 47.92,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T19:18:00Z",
      "Maximum": 50.85,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-04-09T09:18:00Z",
      "Maximum": 47.92,
      "Unit": "Percent"
    }
  ],
}
```

```
{
  "Timestamp": "2014-04-09T03:18:00Z",
  "Maximum": 76.84,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T21:18:00Z",
  "Maximum": 48.96,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T14:18:00Z",
  "Maximum": 47.92,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T08:18:00Z",
  "Maximum": 47.92,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T16:18:00Z",
  "Maximum": 45.55,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T06:18:00Z",
  "Maximum": 47.92,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T13:18:00Z",
  "Maximum": 45.08,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T05:18:00Z",
  "Maximum": 47.92,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T18:18:00Z",
  "Maximum": 46.88,
  "Unit": "Percent"
}
```

```
},
{
  "Timestamp": "2014-04-09T17:18:00Z",
  "Maximum": 52.08,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T07:18:00Z",
  "Maximum": 47.92,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T02:18:00Z",
  "Maximum": 51.23,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T12:18:00Z",
  "Maximum": 47.67,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-08T23:18:00Z",
  "Maximum": 46.88,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T10:18:00Z",
  "Maximum": 51.91,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T04:18:00Z",
  "Maximum": 47.13,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T15:18:00Z",
  "Maximum": 48.96,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-04-09T00:18:00Z",
  "Maximum": 48.16,
```

```

        "Unit": "Percent"
    },
    {
        "Timestamp": "2014-04-09T01:18:00Z",
        "Maximum": 49.18,
        "Unit": "Percent"
    }
],
"Label": "CPUUtilization"
}

```

複数のディメンションを指定する

次の例は、複数のディメンションを指定する方法を示しています。各ディメンションは名前/値のペアとして指定され、名前と値の間にはカンマが入ります。複数のディメンションはスペースで区切ります。また、1つのメトリクスに複数のディメンションを含む場合は、定義されているディメンションごとに値を指定する必要があります。

get-metric-statistics コマンドを使用するその他の例については、「Amazon CloudWatch デベロッパーガイド」の「メトリクスの統計を取得する」を参照してください。

```

aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace
--dimensions Name=InstanceID,Value=i-abcdef Name=InstanceType,Value=m1.small --
start-time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average
--period 60

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetMetricStatistics](#)」の「」を参照してください。

list-metrics

次の例は、list-metrics を使用する方法を説明しています。

AWS CLI

Amazon のメトリクスを一覧表示するには SNS

次のlist-metrics例は、Amazon のメトリクスを示していますSNS。

```

aws cloudwatch list-metrics \
--namespace "AWS/SNS"

```

出力:

```
{
  "Metrics": [
    {
      "Namespace": "AWS/SNS",
      "Dimensions": [
        {
          "Name": "TopicName",
          "Value": "NotifyMe"
        }
      ],
      "MetricName": "PublishSize"
    },
    {
      "Namespace": "AWS/SNS",
      "Dimensions": [
        {
          "Name": "TopicName",
          "Value": "CF0"
        }
      ],
      "MetricName": "PublishSize"
    },
    {
      "Namespace": "AWS/SNS",
      "Dimensions": [
        {
          "Name": "TopicName",
          "Value": "NotifyMe"
        }
      ],
      "MetricName": "NumberOfNotificationsFailed"
    },
    {
      "Namespace": "AWS/SNS",
      "Dimensions": [
        {
          "Name": "TopicName",
          "Value": "NotifyMe"
        }
      ],
      "MetricName": "NumberOfNotificationsDelivered"
    }
  ],
}
```



```
{
  "Namespace": "AWS/SNS",
  "Dimensions": [
    {
      "Name": "TopicName",
      "Value": "NotifyMe"
    }
  ],
  "MetricName": "NumberOfMessagesPublished"
},
{
  "Namespace": "AWS/SNS",
  "Dimensions": [
    {
      "Name": "TopicName",
      "Value": "CF0"
    }
  ],
  "MetricName": "NumberOfMessagesPublished"
},
{
  "Namespace": "AWS/SNS",
  "Dimensions": [
    {
      "Name": "TopicName",
      "Value": "CF0"
    }
  ],
  "MetricName": "NumberOfNotificationsDelivered"
},
{
  "Namespace": "AWS/SNS",
  "Dimensions": [
    {
      "Name": "TopicName",
      "Value": "CF0"
    }
  ],
  "MetricName": "NumberOfNotificationsFailed"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMetrics](#)」の「」を参照してください。

put-metric-alarm

次のコード例は、put-metric-alarm を使用する方法を示しています。

AWS CLI

CPU使用率が 70% を超えたときに Amazon Simple Notification Service の E メールメッセージを送信するには

次の例では、put-metric-alarm コマンドを使用して、CPU使用率が 70% を超えたときに Amazon Simple Notification Service の E メールメッセージを送信します。

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --dimensions "Name=InstanceId,Value=i-12345678" --evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent
```

正常に完了すると、このコマンドはプロンプトに戻ります。同じ名前のアラームが既に存在する場合は、新しいアラームで上書きされます。

複数のディメンションを指定するには

次の例は、複数のディメンションを指定する方法を示しています。各ディメンションは名前/値のペアとして指定され、名前と値の間にはカンマが入ります。複数のディメンションはスペースで区切ります。

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-description "The default example alarm" --namespace "CW EXAMPLE METRICS" --metric-name Default_Test --statistic Average --period 60 --evaluation-periods 3 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --dimensions Name=key1,Value=value1 Name=key2,Value=value2
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutMetricAlarm](#)」の「」を参照してください。

put-metric-data

次の例は、put-metric-data を使用する方法を説明しています。

AWS CLI

カスタムメトリクスを Amazon に発行するには CloudWatch

次の例では、put-metric-data コマンドを使用してカスタムメトリクスを Amazon に発行します CloudWatch。

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://metric.json
```

メトリクス自体の値は、JSON ファイル に保存されます metric.json。

ファイルの内容は次のとおりです。

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

詳細については、「Amazon CloudWatch デベロッパーガイド」の「カスタムメトリクスの発行」を参照してください。

複数のディメンションを指定するには

次の例は、複数のディメンションを指定する方法を示しています。各ディメンションは Name=Value ペアとして指定されます。複数のディメンションはコンマで区切ります。

```
aws cloudwatch put-metric-data --metric-name Buffers --
namespace MyNameSpace --unit Bytes --value 231434333 --
dimensions InstanceID=1-23456789,InstanceType=m1.small
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutMetricData](#)」の「」を参照してください。

set-alarm-state

次の例は、set-alarm-state を使用する方法を説明しています。

AWS CLI

アラームの状態を一時的に変更するには

次の例では、set-alarm-state コマンドを使用して「myalarm」という名前の Amazon CloudWatch アラームの状態を一時的に変更し、テスト目的で ALARM 状態に設定します。

```
aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value ALARM --state-reason "testing purposes"
```

正常に完了すると、このコマンドはプロンプトに戻ります。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetAlarmState](#)」の「」を参照してください。

CloudWatch を使用して例をログに記録します AWS CLI

次のコード例は、を CloudWatch Logs AWS Command Line Interface で使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-log-group

次のコード例は、create-log-group を使用する方法を示しています。

AWS CLI

次のコマンドは、my-logs という名前のロググループを作成します。

```
aws logs create-log-group --log-group-name my-logs
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLogGroup](#)」の「」を参照してください。

create-log-stream

次の例は、create-log-stream を使用する方法を説明しています。

AWS CLI

次のコマンドは、ロググループ my-logs に 20150601 という名前のログストリームを作成します。

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLogStream](#)」の「」を参照してください。

delete-log-group

次の例は、delete-log-group を使用する方法を説明しています。

AWS CLI

以下のコマンドは、my-logs という名前のロググループを削除します。

```
aws logs delete-log-group --log-group-name my-logs
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLogGroup](#)」の「」を参照してください。

delete-log-stream

次の例は、delete-log-stream を使用する方法を説明しています。

AWS CLI

次のコマンドは、 という名前のロググループ20150531から という名前のログストリームを削除しますmy-logs。

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLogStream](#)」の「」を参照してください。

delete-retention-policy

次のコード例は、delete-retention-policy を使用する方法を示しています。

AWS CLI

次のコマンドは、 という名前のロググループに以前に適用された保持ポリシーを削除しますmy-logs。

```
aws logs delete-retention-policy --log-group-name my-logs
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRetentionPolicy](#)」の「」を参照してください。

describe-log-groups

次の例は、describe-log-groups を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-logs という名前のロググループを記述します。

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

出力:

```
{
  "logGroups": [
    {
      "storedBytes": 0,
```

```
        "metricFilterCount": 0,  
        "creationTime": 1433189500783,  
        "logGroupName": "my-logs",  
        "retentionInDays": 5,  
        "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"  
    }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLogGroups](#)」の「」を参照してください。

describe-log-streams

次の例は、describe-log-streams を使用する方法を説明しています。

AWS CLI

次のコマンドは、ロググループのプレフィックスで始まるすべての2015ログストリームを表示しますmy-logs。

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-prefix 2015
```

出力:

```
{  
  "logStreams": [  
    {  
      "creationTime": 1433189871774,  
      "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-stream:20150531",  
      "logStreamName": "20150531",  
      "storedBytes": 0  
    },  
    {  
      "creationTime": 1433189873898,  
      "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-stream:20150601",  
      "logStreamName": "20150601",  
      "storedBytes": 0  
    }  
  ]  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLogStreams](#)」の「」を参照してください。

get-log-events

次のコード例は、get-log-events を使用する方法を示しています。

AWS CLI

次のコマンドは、ロググループ 20150601 の という名前のログストリームからログイベントを取得しますmy-logs。

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

出力:

```
{
  "nextForwardToken":
  "f/31961209122447488583055879464742346735121166569214640130",
  "events": [
    {
      "ingestionTime": 1433190494190,
      "timestamp": 1433190184356,
      "message": "Example Event 1"
    },
    {
      "ingestionTime": 1433190516679,
      "timestamp": 1433190184356,
      "message": "Example Event 1"
    },
    {
      "ingestionTime": 1433190494190,
      "timestamp": 1433190184358,
      "message": "Example Event 2"
    }
  ],
  "nextBackwardToken":
  "b/31961209122358285602261756944988674324553373268216709120"
}
```


- API 詳細については、AWS CLI 「コマンドリファレンス[GetLogEvents](#)」の「」を参照してください。

put-log-events

次の例は、put-log-events を使用する方法を説明しています。

AWS CLI

次のコマンドは、ロググループ 20150601内の という名前のログストリームにログイベントを配置しますmy-logs。

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-events file://events
```

出力:

```
{
  "nextSequenceToken": "49542672486831074009579604567656788214806863282469607346"
}
```

上記の例では、現在のディレクトリeventsにある という名前のファイルからイベントのJSON配列を読み取ります。

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

後続の各呼び出しでは、前の呼び出しで提供された次のシーケンストークンをシーケンストークンオプションで指定する必要があります。

```
aws logs put-log-events --log-group-name my-logs --log-  
stream-name 20150601 --log-events file://events2 --sequence-  
token "49542672486831074009579604567656788214806863282469607346"
```

出力:

```
{  
  "nextSequenceToken": "49542672486831074009579604567900991230369019956308219826"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLogEvents](#)」の「」を参照してください。

put-retention-policy

次のコード例は、put-retention-policy を使用する方法を示しています。

AWS CLI

次のコマンドは、 という名前のロググループに 5 日間の保持ポリシーを追加しますmy-logs。

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRetentionPolicy](#)」の「」を参照してください。

CloudWatch を使用したネットワークモニタリングの例 AWS CLI

次のコード例は、CloudWatch Network Monitoring AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

• [アクション](#)

アクション

create-monitor

次のコード例は、`create-monitor` を使用する方法を示しています。

AWS CLI

例 1: 集約期間を使用してネットワークモニターを作成するには

次の`create-monitor`例では、という名前`Example_NetworkMonitor`のモニターを作成し、を 30 秒`aggregationPeriod`に設定します。モニターにはプローブが関連付けられていない`INACTIVE`ため、モニター`state`のイニシャルは になります。状態は、プローブが追加された`ACTIVE`場合にのみ に変わります。[update-monitor](#) コマンドまたは [create-probe](#) コマンドを使用して、このモニターにプローブを追加できます。

```
aws networkmonitor create-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30
```

出力:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "INACTIVE",  
  "aggregationPeriod": 30,  
  "tags": {}  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

例 2: を使用してプローブでネットワークモニターを作成しTCP、タグを含めるには

次の`create-monitor`例では、という名前のモニターを作成します`Example_NetworkMonitor`。コマンドは、ICMPプロトコルを使用し、タグを含む 1 つのプ

プローブも作成します。リクエストで `aggregationPeriod` が渡されないため、60秒がデフォルトとして設定されます。プローブを備えたモニターstateのは、モニターが PENDINGになるまでになりますACTIVE。これには数分かかる場合があります。その時点で、stateは に変わりACTIVE、CloudWatch メトリクスの表示を開始できます。

```
aws networkmonitor create-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --probes sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-  
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,probeTags={Name=Prob  
 \  
  --tags Monitor=Monitor1
```

出力:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region111122223333:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 60,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

例 3: を使用してプローブでネットワークモニターを作成しICMP、タグを含めるには

次のcreate-monitor例では、という名前aggregationPeriodのモニターを30秒Example_NetworkMonitorで作成します。コマンドは、ICMPプロトコルを使用し、タグを含む1つのプローブも作成します。リクエストで `aggregationPeriod` が渡されないため、60秒がデフォルトとして設定されます。プローブを備えたモニターstateのは、モニターが PENDINGになるまでになりますACTIVE。これには数分かかる場合があります。その時点で、stateは に変わりACTIVE、CloudWatch メトリクスの表示を開始できます。

```
aws networkmonitor create-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30 \  
  --probes protocol=ICMP,packetSize=64,probeTags={Name=Probe1}
```

```
--probes sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-  
id,destination=10.0.0.100,protocol=ICMP,packetSize=56,probeTags={Name=Probe1} \  
--tags Monitor=Monitor1
```

出力:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 30,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateMonitor](#)」の「」を参照してください。

create-probe

次のコード例は、create-probe を使用する方法を示しています。

AWS CLI

例 1: を使用してネットワークモニターTCPに追加するプローブを作成するには

次のcreate-probe例では、 を使用するプローブを作成しTCPprotocol、プローブを という名前のモニターに追加しますExample_NetworkMonitor。作成されると、プローブを含むモニターstateのは、モニターが PENDINGになるまでになりますACTIVE。これには数分かかる場合があります、その時点で状態は に変わりACTIVE、 CloudWatch メトリクスの表示を開始できます。

```
aws networkmonitor create-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe sourceArn=arn:aws:ec2:region:111122223333:subnet/subnet-  
id,destination=10.0.0.100,destinationPort=80,protocol=TCP,packetSize=56,tags={Name=Probe1}
```

出力:

```
{
  "probeId": "probe-12345",
  "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",
  "destination": "10.0.0.100",
  "destinationPort": 80,
  "packetSize": 56,
  "addressFamily": "IPV4",
  "vpcId": "vpc-12345",
  "state": "PENDING",
  "createdAt": "2024-03-29T12:41:57.314000-04:00",
  "modifiedAt": "2024-03-29T12:41:57.314000-04:00",
  "tags": {
    "Name": "Probe1"
  }
}
```

例 2: を使用してプローブを使用するプローブを作成しICMP、ネットワークモニターに追加するには

次のcreate-probe例では、 を使用するプローブを作成しICMPprotocol、プローブを という名前のモニターに追加しますExample_NetworkMonitor。作成されると、プローブを含むモニターstateのは、モニターが PENDINGになるまでになりますACTIVE。これには数分かかる場合があります、その時点で状態は に変わりACTIVE、CloudWatch メトリクスの表示を開始できます。

```
aws networkmonitor create-probe \
  --monitor-name Example_NetworkMonitor \
  --probe sourceArn=arn:aws:ec2:region:012345678910:subnet/subnet-
id,destination=10.0.0.100,protocol=ICMP,packetSize=56,tags={Name=Probe1}
```

出力:

```
{
  "probeId": "probe-12345",
  "probeArn": "arn:aws:networkmonitor:region:111122223333:probe/probe-12345",
  "destination": "10.0.0.100",
  "packetSize": 56,
  "addressFamily": "IPV4",
  "vpcId": "vpc-12345",
  "state": "PENDING",
  "createdAt": "2024-03-29T12:44:02.452000-04:00",
```

```
"modifiedAt": "2024-03-29T12:44:02.452000-04:00",
"tags": {
  "Name": "Probe1"
}
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProbe](#)」の「」を参照してください。

delete-monitor

次の例は、delete-monitor を使用する方法を説明しています。

AWS CLI

モニターを削除するには

次のdelete-monitor例では、`Example_NetworkMonitor` という名前のモニターを削除します。

```
aws networkmonitor delete-monitor \
  --monitor-name Example_NetworkMonitor
```

このコマンドでは何も出力されません。

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMonitor](#)」の「」を参照してください。

delete-probe

次の例は、delete-probe を使用する方法を説明しています。

AWS CLI

プローブを削除するには

次のdelete-probe例では、 という名前のネットワークモニターprobe-12345から ID を持つプローブを削除しますExample_NetworkMonitor。

```
aws networkmonitor delete-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe-id probe-12345
```

このコマンドでは何も出力されません。

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProbe](#)」の「」を参照してください。

get-monitor

次のコード例は、get-monitor を使用する方法を示しています。

AWS CLI

モニター情報を取得するには

次のget-monitor例では、 という名前のモニターに関する情報を取得しますExample_NetworkMonitor。

```
aws networkmonitor get-monitor \  
  --monitor-name Example_NetworkMonitor
```

出力:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "ACTIVE",  
  "aggregationPeriod": 60,  
  "tags": {},  
  "probes": [],  
  "createdAt": "2024-04-01T17:58:07.211000-04:00",  
  "modifiedAt": "2024-04-01T17:58:07.211000-04:00"
```



```
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMonitor](#)」の「」を参照してください。

get-probe

次の例は、get-probe を使用する方法を説明しています。

AWS CLI

プローブの詳細を表示するには

次のget-probe例では、 という名前のモニターprobeIDprobe-12345に関連付けられているを持つプローブの詳細を返しますExample_NetworkMonitor。

```
aws networkmonitor get-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe-id probe-12345
```

出力:

```
{  
  "probeId": "probe-12345",  
  "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",  
  "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",  
  "destination": "10.0.0.100",  
  "destinationPort": 80,  
  "protocol": "TCP",  
  "packetSize": 56,  
  "addressFamily": "IPV4",  
  "vpcId": "vpc-12345",  
  "state": "ACTIVE",  
  "createdAt": "2024-03-29T12:41:57.314000-04:00",  
  "modifiedAt": "2024-03-29T12:42:28.610000-04:00",  
  "tags": {  
    "Name": "Probe1"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[GetProbe](#)」の「」を参照してください。

list-monitors

次のコード例は、list-monitors を使用する方法を示しています。

AWS CLI

例 1: すべてのモニターを一覧表示するには (単一モニター)

次のlist-monitors例では、1つのモニターのみを返します。モニターstateはACTIVEで、は60秒aggregationPeriodです。

```
aws networkmonitor list-monitors
```

出力:

```
{
  "monitors": [{
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "ACTIVE",
    "aggregationPeriod": 60,
    "tags": {
      "Monitor": "Monitor1"
    }
  ]
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

例 2: すべてのモニターを一覧表示するには (複数のモニター)

次のlist-monitors例では、3つのモニターのリストを返します。1つのモニターstateのはACTIVEであり、CloudWatch メトリクスを生成します。他の2つのモニターの状態は

INACTIVEであり、CloudWatch メトリクスを生成しません。3 つのモニターはすべて 60 秒aggregationPeriodの を使用します。

```
aws networkmonitor list-monitors
```

出力:

```
{
  "monitors": [
    {
      "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor",
      "monitorName": "Example_NetworkMonitor",
      "state": "INACTIVE",
      "aggregationPeriod": 60,
      "tags": {}
    },
    {
      "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
Example_NetworkMonitor2",
      "monitorName": "Example_NetworkMonitor2",
      "state": "ACTIVE",
      "aggregationPeriod": 60,
      "tags": {
        "Monitor": "Monitor1"
      }
    },
    {
      "monitorArn": "arn:aws:networkmonitor:us-east-1:111122223333:monitor/
TestNetworkMonitor_CLI",
      "monitorName": "TestNetworkMonitor_CLI",
      "state": "INACTIVE",
      "aggregationPeriod": 60,
      "tags": {}
    }
  ]
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMonitors](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、 という名前のモニターのタグのリストを返しますExample_NetworkMonitor。

```
aws networkmonitor list-tags-for-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor
```

出力:

```
{  
  "tags": {  
    "Environment": "Dev",  
    "Application": "PetStore"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、 と タグExample_NetworkMonitorで という名前のモニターにApplication=PetStoreタグを付けEnvironment=Devます。

```
aws networkmonitor tag-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor
```

```
--resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
--tags Environment=Dev,Application=PetStore
```

このコマンドでは何も出力されません。

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、`untag-resource` を使用する方法を説明しています。

AWS CLI

リソースのタグを解除するには

次の`untag-resource`例では、 という名前のモニターとの関連付けEnvironment Applicationから、 のキーと値のペアを持つ `tag-keys`パラメータを削除しますExample_NetworkMonitor。

```
aws networkmonitor untag-resource \  
--resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
--tag-keys Environment Application
```

このコマンドでは何も出力されません。

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-monitor

次のコード例は、`update-monitor` を使用する方法を示しています。

AWS CLI

モニターを更新するには

次のupdate-monitor例では、モニターの aggregationPeriod を60秒から30秒に変更します。

```
aws networkmonitor update-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30
```

出力:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 30,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMonitor](#)」の「」を参照してください。

update-probe

次の例は、update-probe を使用する方法を説明しています。

AWS CLI

プローブを更新するには

次のupdate-probe例では、プローブの元の destination IP アドレスを更新し、 packetSize に更新します60。

```
aws networkmonitor update-probe \  
  --destination-ip 10.0.0.1 \  
  --packet-size 60
```

```
--monitor-name Example_NetworkMonitor \  
--probe-id probe-12345 \  
--destination 10.0.0.150 \  
--packet-size 60
```

出力:

```
{  
  "probeId": "probe-12345",  
  "probeArn": "arn:aws:networkmonitor:region:012345678910:probe/probe-12345",  
  "sourceArn": "arn:aws:ec2:region:012345678910:subnet/subnet-12345",  
  "destination": "10.0.0.150",  
  "destinationPort": 80,  
  "protocol": "TCP",  
  "packetSize": 60,  
  "addressFamily": "IPV4",  
  "vpcId": "vpc-12345",  
  "state": "PENDING",  
  "createdAt": "2024-03-29T12:41:57.314000-04:00",  
  "modifiedAt": "2024-03-29T13:52:23.115000-04:00",  
  "tags": {  
    "Name": "Probe1"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon CloudWatch Network Monitor の仕組み」](#)を参照してください。 CloudWatch

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateProbe](#)」の「」を参照してください。

CodeArtifact を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CodeArtifact。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-external-connection

次の例は、`associate-external-connection` を使用する方法を説明しています。

AWS CLI

リポジトリに外部接続を追加するには

次の`associate-external-connection`例では、`test-repo` という名前のリポジトリに `npmjs.com` への外部接続を追加します。

```
aws codeartifact associate-external-connection \  
  --repository test-repo \  
  --domain test-domain \  
  --external-connection public:npmjs
```

出力:

```
{  
  "repository": {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/  
test-repo",  
    "upstreams": [],  
    "externalConnections": [  
      {  
        "externalConnectionName": "public:npmjs",  
        "packageFormat": "npm",  
        "status": "AVAILABLE"  
      }  
    ]  
  }  
}
```


詳細については、「ユーザーガイド」の「[外部接続の追加](#)」を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateExternalConnection](#)」の「」を参照してください。

copy-package-versions

次の例は、copy-package-versions を使用する方法を説明しています。

AWS CLI

あるリポジトリから別のリポジトリにパッケージバージョンをコピーするには

以下は、test-package という名前のパッケージのバージョン 4.0.0 および 5.0.0 を my-repo から test-repo copy-package-versions に移動します。

```
aws codeartifact copy-package-versions \  
  --domain test-domain \  
  --source-repository my-repo \  
  --destination-repository test-repo \  
  --format npm \  
  --package test-package \  
  --versions '["4.0.0", "5.0.0"]'
```

出力:

```
{  
  "format": "npm",  
  "package": "test-package",  
  "versions": [  
    {  
      "version": "5.0.0",  
      "revision": "REVISION-1-SAMPLE-6C81EFF7DA55CC",  
      "status": "Published"  
    },  
    {  
      "version": "4.0.0",  
      "revision": "REVISION-2-SAMPLE-55C752BEE772FC",  
      "status": "Published"  
    }  
  ]  
}
```

```
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の「[リポジトリ間でパッケージをコピーする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyPackageVersions](#)」の「」を参照してください。

create-domain

次の例は、create-domain を使用する方法を説明しています。

AWS CLI

ドメインを作成するには

次のcreate-domain例では、test-domain という名前のドメインを作成します。

```
aws codeartifact create-domain \  
  --domain test-domain
```

出力:

```
{  
  "domain": {  
    "name": "test-domain",  
    "owner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",  
    "status": "Active",  
    "createdTime": "2020-10-20T13:16:48.559000-04:00",  
    "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111",  
    "repositoryCount": 0,  
    "assetSizeBytes": 0  
  }  
}
```

詳細については、「ユーザーガイド」の「[ドメインの作成](#)」を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDomain](#)」の「」を参照してください。

create-repository

次の例は、create-repository を使用する方法を説明しています。

AWS CLI

リポジトリを作成するには

次のcreate-repository例では、test-domain という名前のドメイン内に test-repo という名前のリポジトリを作成します。

```
aws codeartifact create-repository \  
  --domain test-domain \  
  --domain-owner 111122223333 \  
  --repository test-repo \  
  --description "This is a test repository."
```

出力:

```
{  
  "repository": {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/  
test-repo",  
    "description": "This is a test repository.",  
    "upstreams": [],  
    "externalConnections": []  
  }  
}
```

詳細については、「ユーザーガイド」の「[ドメインの作成](#)」を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRepository](#)」の「」を参照してください。

delete-domain-permissions-policy

次の例は、delete-domain-permissions-policy を使用する方法を説明しています。

AWS CLI

アクセス許可ポリシードキュメントをドメインから削除するには

次のdelete-domain-permissions-policy例では、test-domain という名前のドメインからアクセス許可ポリシーを削除します。

```
aws codeartifact delete-domain-permissions-policy \  
  --domain test-domain
```

出力:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "BasicDomainPolicy",  
      "Action": [  
        "codeartifact:GetDomainPermissionsPolicy",  
        "codeartifact:ListRepositoriesInDomain",  
        "codeartifact:GetAuthorizationToken",  
        "codeartifact:CreateRepository"  
      ],  
      "Effect": "Allow",  
      "Resource": "*",  
      "Principal": {  
        "AWS": "arn:aws:iam::111122223333:root"  
      }  
    }  
  ]  
}
```

詳細については、「ユーザーガイド」の[「ドメインポリシーの削除」](#)を参照してください。

AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDomainPermissionsPolicy](#)」の「」を参照してください。

delete-domain

次の例は、delete-domain を使用する方法を説明しています。

AWS CLI

ドメインを削除するには

次のdelete-domain例では、`test-domain`という名前のドメインを削除します。

```
aws codeartifact delete-domain \  
  --domain test-domain
```

出力:

```
{  
  "domain": {  
    "name": "test-domain",  
    "owner": "417498243647",  
    "arn": "arn:aws:codeartifact:us-west-2:417498243647:domain/test-domain",  
    "status": "Deleted",  
    "createdTime": "2020-10-20T13:16:48.559000-04:00",  
    "encryptionKey": "arn:aws:kms:us-west-2:417498243647:key/c9fe2447-0795-4fda-  
afbe-8464574ae162",  
    "repositoryCount": 0,  
    "assetSizeBytes": 0  
  }  
}
```

詳細については、「ユーザーガイド」の「[ドメインの削除](#)」を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDomain](#)」の「」を参照してください。

delete-package-versions

次のコード例は、delete-package-versions を使用する方法を示しています。

AWS CLI

パッケージバージョンを削除するには

次のdelete-package-versions例では、`test-package` という名前のパッケージのバージョン `4.0.0` を削除します。

```
aws codeartifact delete-package-versions \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --versions 4.0.0
```

出力:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Deleted"  
    }  
  },  
  "failedVersions": {}  
}
```

詳細については、「ユーザーガイド」の「[パッケージバージョンの削除](#)」を参照してください。
AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePackageVersions](#)」の「」を参照してください。

delete-repository-permissions-policy

次のコード例は、delete-repository-permissions-policy を使用する方法を示しています。

AWS CLI

リポジトリからアクセス許可ポリシーを削除するには

次のdelete-repository-permissions-policy例では、test-repo という名前のリポジトリからアクセス許可ポリシーを削除します。

```
aws codeartifact delete-repository-permissions-policy \  
  --domain test-domain \  
  --repository test-repo
```

出力:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "codeartifact:DescribePackageVersion",
        "codeartifact:DescribeRepository",
        "codeartifact:GetPackageVersionReadme",
        "codeartifact:GetRepositoryEndpoint",
        "codeartifact:ListPackages",
        "codeartifact:ListPackageVersions",
        "codeartifact:ListPackageVersionAssets",
        "codeartifact:ListPackageVersionDependencies",
        "codeartifact:ReadFromRepository"
      ],
      "Resource": "*"
    }
  ]
}
```

詳細については、「ユーザーガイド」の「[ポリシーの削除](#)」を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRepositoryPermissionsPolicy](#)」の「」を参照してください。

delete-repository

次の例は、delete-repository を使用する方法を説明しています。

AWS CLI

リポジトリを削除するには

次のdelete-repository例では、 という名前のドメインtest-repo内の という名前のリポジトリを削除しますtest-domain。

```
aws codeartifact delete-repository \
```

```
--domain test-domain \  
--repository test-repo
```

出力:

```
{  
  "repository": {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/  
test-repo",  
    "description": "This is a test repository",  
    "upstreams": [],  
    "externalConnections": []  
  }  
}
```

詳細については、「ユーザーガイド」の [「リポジトリの削除」](#) を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteRepository](#)」の「」を参照してください。

describe-domain

次の例は、describe-domain を使用する方法を説明しています。

AWS CLI

ドメインに関する情報を取得するには

次のdescribe-domain例では、test-domain という名前のドメインの DomainDescription オブジェクトを返します。

```
aws codeartifact describe-domain \  
--domain test-domain
```

出力:

```
{
```



```
"domain": {
  "name": "test-domain",
  "owner": "111122223333",
  "arn": "arn:aws:codeartifact:us-west-2:111122223333:domain/test-domain",
  "status": "Active",
  "createdTime": "2020-10-20T13:16:48.559000-04:00",
  "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
  "repositoryCount": 2,
  "assetSizeBytes": 0,
  "s3BucketArn": "arn:aws:s3:::assets-111122223333-us-west-2"
}
```

詳細については、「ユーザーガイド」の[「ドメインの概要」](#)を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDomain](#)」の「」を参照してください。

describe-repository

次のコード例は、describe-repository を使用する方法を示しています。

AWS CLI

リポジトリに関する情報を取得するには

次のdescribe-repository例では、test-repo という名前のリポジトリの RepositoryDescription オブジェクトを返します。

```
aws codeartifact describe-repository \
  --domain test-domain \
  --repository test-repo
```

出力:

```
{
  "repository": {
    "name": "test-repo",
    "administratorAccount": "111122223333",
    "domainName": "test-domain",
```

```
    "domainOwner": "111122223333",
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
    "description": "This is a test repository.",
    "upstreams": [],
    "externalConnections": []
  }
}
```

詳細については、「ユーザーガイド」の[「ドメインの作成」](#)を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRepository](#)」の「」を参照してください。

disassociate-external-connection

次の例は、disassociate-external-connection を使用する方法を説明しています。

AWS CLI

リポジトリから外部接続を削除するには

次のdisassociate-external-connection例では、test-repo という名前のリポジトリから npmjs.com への外部接続を削除します。

```
aws codeartifact disassociate-external-connection \
  --repository test-repo \
  --domain test-domain \
  --external-connection public:npmjs
```

出力:

```
{
  "repository": {
    "name": "test-repo",
    "administratorAccount": "111122223333",
    "domainName": "test-domain",
    "domainOwner": "111122223333",
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/
test-repo",
    "upstreams": [],
```

```
    "externalConnections": []
  }
}
```

詳細については、「ユーザーガイド」の[「外部接続の削除」](#)を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateExternalConnection](#)」の「」を参照してください。

dispose-package-versions

次の例は、dispose-package-versions を使用する方法を説明しています。

AWS CLI

パッケージバージョンのアセットを削除し、そのステータスを Disposed に設定するには

次のdispose-package-versions例では、テストパッケージバージョン 4.0.0 のアセットを削除し、そのステータスを Disposed に設定します。

```
aws codeartifact dispose-package-versions \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
  --versions 4.0.0
```

出力:

```
{
  "successfulVersions": {
    "4.0.0": {
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
      "status": "Disposed"
    }
  },
  "failedVersions": {}
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の[「でのパッケージの使用 CodeArtifact」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisposePackageVersions](#)」の「」を参照してください。

get-authorization-token

次の例は、get-authorization-token を使用する方法を説明しています。

AWS CLI

認証トークンを取得するには

次のget-authorization-token例では、CodeArtifact 認証トークンを取得します。

```
aws codeartifact get-authorization-token \  
  --domain test-domain \  
  --query authorizationToken \  
  --output text
```

出力:

```
This command will return the authorization token. You can store the output in an  
environment variable when calling the command.
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の「[ログインコマンドなしで pip を設定する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAuthorizationToken](#)」の「」を参照してください。

get-domain-permissions-policy

次のコード例は、get-domain-permissions-policy を使用する方法を示しています。

AWS CLI

ドメインのアクセス許可ポリシードキュメントを取得するには

次のget-domain-permissions-policy例では、test-domain という名前のドメインにアタッチされたアクセス許可ポリシーを取得します。

```
aws codeartifact get-domain-permissions-policy \  
  --domain test-domain \  
  --policy test-policy \  
  --output text
```

```
--domain test-domain
```

出力:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BasicDomainPolicy",
      "Action": [
        "codeartifact:GetDomainPermissionsPolicy",
        "codeartifact:ListRepositoriesInDomain",
        "codeartifact:GetAuthorizationToken",
        "codeartifact:CreateRepository"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      }
    }
  ]
}
```

詳細については、「ユーザーガイド」の [「ドメインポリシーの読み取り」](#) を参照してください。
AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDomainPermissionsPolicy](#)」の「」を参照してください。

get-package-version-asset

次の例は、get-package-version-asset を使用する方法を説明しています。

AWS CLI

パッケージバージョンからアセットを取得するには

次のget-package-version-asset例では、test-package という名前の npm パッケージのバージョン 4.0.0 のpackage.tgzアセットを取得します。

```
aws codeartifact get-package-version-asset \
```

```
--domain test-domain \  
--repository test-repo \  
--format npm \  
--package test-package \  
--package-version 4.0.0 \  
--asset 'package.tgz' \  
outfileName
```

出力:

The output for this command will also store the raw asset in the file provided in place of outfileName.

```
{  
  "assetName": "package.tgz",  
  "packageVersion": "4.0.0",  
  "packageVersionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=" }  
}
```

詳細については、「ユーザーガイド」の[「パッケージバージョンアセットを一覧表示する」](#)を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPackageVersionAsset](#)」の「」を参照してください。

get-package-version-readme

次の例は、get-package-version-readme を使用する方法を説明しています。

AWS CLI

パッケージバージョンの readme ファイルを取得するには

次のget-package-version-readme例では、test-package という名前の npm パッケージのバージョン 4.0.0 の readme ファイルを取得します。

```
aws codeartifact get-package-version-readme \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

出力:

```
{
  "format": "npm",
  "package": "test-package",
  "version": "4.0.0",
  "readme": "<div align=\"center\">\n  <a href=\"https://github.com/test-package/testpack\"> ... more content ... \n",
  "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs="
}
```

詳細については、「ユーザーガイド」の[「パッケージバージョンの readme ファイルを表示する」](#)を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPackageVersionReadme](#)」の「」を参照してください。

get-repository-endpoint

次の例は、get-repository-endpoint を使用する方法を説明しています。

AWS CLI

リポジトリのURLエンドポイントを取得するには

次のget-repository-endpoint例では、test-repo リポジトリの npm エンドポイントを返します。

```
aws codeartifact get-repository-endpoint \
  --domain test-domain \
  --repository test-repo \
  --format npm
```

出力:

```
{
  "repositoryEndpoint": "https://test-domain-111122223333.d.codeartifact.us-west-2.amazonaws.com/npm/test-repo/"
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の[「リポジトリに接続する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRepositoryEndpoint](#)」の「」を参照してください。

get-repository-permissions-policy

次の例は、get-repository-permissions-policy を使用する方法を説明しています。

AWS CLI

リポジトリのアクセス許可ポリシードキュメントを取得するには

次のget-repository-permissions-policy例では、test-repo という名前のリポジトリにアタッチされたアクセス許可ポリシーを取得します。

```
aws codeartifact get-repository-permissions-policy \  
  --domain test-domain \  
  --repository test-repo
```

出力:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::111122223333:root"  
      },  
      "Action": [  
        "codeartifact:DescribePackageVersion",  
        "codeartifact:DescribeRepository",  
        "codeartifact:GetPackageVersionReadme",  
        "codeartifact:GetRepositoryEndpoint",  
        "codeartifact:ListPackages",  
        "codeartifact:ListPackageVersions",  
        "codeartifact:ListPackageVersionAssets",  
        "codeartifact:ListPackageVersionDependencies",  
        "codeartifact:ReadFromRepository"  
      ],  
      "Resource": "*"   
    }  
  ]  
}
```



```
}
```

詳細については、「ユーザーガイド」の「[ポリシーの読み取り](#)」を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRepositoryPermissionsPolicy](#)」の「」を参照してください。

list-domains

次の例は、list-domains を使用する方法を説明しています。

AWS CLI

ドメインを一覧表示するには

次のlist-domains例では、呼び出しを行う AWS アカウントが所有するすべてのドメインの概要を返します。

```
aws codeartifact list-domains
```

出力:

```
{
  "domains": [
    {
      "name": "my-domain",
      "owner": "111122223333",
      "status": "Active",
      "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    {
      "name": "test-domain",
      "owner": "111122223333",
      "status": "Active",
      "encryptionKey": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    }
  ]
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の「[でのドメインの使用 CodeArtifact](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDomains](#)」の「」を参照してください。

list-package-version-assets

次のコード例は、list-package-version-assets を使用する方法を示しています。

AWS CLI

パッケージバージョンのアセットを表示するには

次のlist-package-version-assets例では、test-package という名前の npm パッケージのバージョン 4.0.0 のアセットを取得します。

```
aws codeartifact list-package-version-assets \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

出力:

```
{  
  "format": "npm",  
  "package": "test-package",  
  "version": "4.0.0",  
  "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
  "assets": [  
    {  
      "name": "package.tgz",  
      "size": 316680,  
      "hashes": {  
        "MD5": "60078ec6d9e76b89fb55c860832742b2",  
        "SHA-1": "b44a9b6297bcb698f1c51a3545a2b3b368d59c52",  
        "SHA-256":  
        "d2aa8c6afc3c8591765785a37d1c5acae482a8eb3ab9729ed28922692454f2e2",  
        "SHA-512":  
        "3e585d15c8a594e20d7de57b362ea81754c011acb2641a19f1b72c8531ea39825896bab344ae616a0a5a824cb9"
```

```
    }  
  }  
]  
}
```

詳細については、「ユーザーガイド」の「[パッケージバージョンアセットを一覧表示する](#)」を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPackageVersionAssets](#)」の「」を参照してください。

list-package-version-dependencies

次のコード例は、list-package-version-dependencies を使用する方法を示しています。

AWS CLI

パッケージバージョンの依存関係を表示するには

次のlist-package-version-dependencies例では、test-package という名前の npm パッケージのバージョン 4.0.0 の依存関係を取得します。

```
aws codeartifact list-package-version-dependencies \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

出力:

```
{  
  "format": "npm",  
  "package": "test-package",  
  "version": "4.0.0",  
  "versionRevision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
  "dependencies": [  
    {  
      "namespace": "testns",  
      "package": "testdep1",  
      "dependencyType": "regular",  
      "versionRequirement": "1.8.5"    }  
  ]  
}
```

```
    },  
    {  
      "namespace": "testns",  
      "package": "testdep2",  
      "dependencyType": "regular",  
      "versionRequirement": "1.8.5"  
    }  
  ]  
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の [「パッケージバージョンの詳細と依存関係の表示と更新」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListPackageVersionDependencies](#)」の「」を参照してください。

list-package-versions

次のコード例は、list-package-versions を使用する方法を示しています。

AWS CLI

パッケージのパッケージバージョンを一覧表示するには

次のlist-package-versions例では、 という名前のパッケージのパッケージバージョンのリストを返しますkind-of。

```
aws codeartifact list-package-versions \  
  --package kind-of \  
  --domain test-domain \  
  --repository test-repo \  
  --format npm
```

出力:

```
{  
  "defaultDisplayVersion": "1.0.1",  
  "format": "npm",  
  "package": "kind-of",  
  "versions": [  
    {
```

```
    "version": "1.0.1",
    "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC",
    "status": "Published"
  },
  {
    "version": "1.0.0",
    "revision": "REVISION-SAMPLE-2-C752BEEF6D2CFC",
    "status": "Published"
  },
  {
    "version": "0.1.2",
    "revision": "REVISION-SAMPLE-3-654S65A5C5E1FC",
    "status": "Published"
  },
  {
    "version": "0.1.1",
    "revision": "REVISION-SAMPLE-1-C7F4S5E9B772FC",
    "status": "Published"
  },
  {
    "version": "0.1.0",
    "revision": "REVISION-SAMPLE-4-AF669139B772FC",
    "status": "Published"
  }
]
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の [「パッケージバージョンを一覧表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListPackageVersions](#)」の「」を参照してください。

list-packages

次のコード例は、list-packages を使用する方法を示しています。

AWS CLI

リポジトリ内のパッケージを一覧表示するには

次のlist-packages例では、`test-domain` という名前のドメインにある `test-repo` という名前のリポジトリ `test-repo` のパッケージを一覧表示します。

```
aws codeartifact list-packages \  
  --domain test-domain \  
  --repository test-repo
```

出力:

```
{  
  "packages": [  
    {  
      "format": "npm",  
      "package": "lodash"  
    },  
    {  
      "format": "python",  
      "package": "test-package"  
    }  
  ]  
}
```

詳細については、AWS CodeArtifact「ユーザーガイド」の「[パッケージ名を一覧表示する](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListPackages](#)」の「」を参照してください。

list-repositories-in-domain

次のコード例は、list-repositories-in-domain を使用する方法を示しています。

AWS CLI

ドメイン内のリポジトリを一覧表示するには

次のlist-repositories-in-domain例では、テストドメイン内のすべてのリポジトリの概要を返します。

```
aws codeartifact list-repositories-in-domain \  
  --domain test-domain
```

出力:

```
{
  "repositories": [
    {
      "name": "test-repo",
      "administratorAccount": "111122223333",
      "domainName": "test-domain",
      "domainOwner": "111122223333",
      "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo",
      "description": "This is a test repository."
    },
    {
      "name": "test-repo2",
      "administratorAccount": "111122223333",
      "domainName": "test-domain",
      "domainOwner": "111122223333",
      "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-
domain/test-repo2",
      "description": "This is a test repository."
    }
  ]
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の [「リポジトリのリスト」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListRepositoriesInDomain](#)」の「」を参照してください。

list-repositories

次のコード例は、list-repositories を使用する方法を示しています。

AWS CLI

リポジトリを一覧表示するには

次のlist-repositories例では、呼び出しを行う AWS アカウントが所有するドメイン内のすべてのリポジトリの概要を返します。

```
aws codeartifact list-repositories
```

出力:

```
{
  "repositories": [
    {
      "name": "npm-store",
      "administratorAccount": "111122223333",
      "domainName": "my-domain",
      "domainOwner": "111122223333",
      "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-domain/npm-store",
      "description": "Provides npm artifacts from npm, Inc."
    },
    {
      "name": "target-repo",
      "administratorAccount": "111122223333",
      "domainName": "my-domain",
      "domainOwner": "111122223333",
      "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/my-domain/target-repo",
      "description": "test target repo"
    },
    {
      "name": "test-repo2",
      "administratorAccount": "111122223333",
      "domainName": "test-domain",
      "domainOwner": "111122223333",
      "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/test-repo2",
      "description": "This is a test repository."
    }
  ]
}
```

詳細については、AWS CodeArtifact「ユーザーガイド」の[「リポジトリを一覧表示する」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListRepositories](#)」の「」を参照してください。

login

次の例は、login を使用する方法を説明しています。

AWS CLI

ログインコマンドを使用してリポジトリへの認証を設定するには

次のlogin例では、test-domain という名前のドメインに test-repo という名前のリポジトリを使用して npm パッケージマネージャーを設定します。

```
aws codeartifact login \  
  --domain test-domain \  
  --repository test-repo \  
  --tool npm
```

出力:

```
Successfully configured npm to use AWS CodeArtifact repository https://test-domain-111122223333.d.codeartifact.us-west-2.amazonaws.com/npm/test-repo/  
Login expires in 12 hours at 2020-11-12 01:53:16-05:00
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の「[の開始方法 AWS CLI](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[ログイン](#)」を参照してください。AWS CLI

put-domain-permissions-policy

次のコード例は、put-domain-permissions-policy を使用する方法を示しています。

AWS CLI

アクセス許可ポリシーをドメインにアタッチするには

次のput-domain-permissions-policy例では、policy.json ファイルで定義されているアクセス許可ポリシーを test-domain という名前のドメインにアタッチします。

```
aws codeartifact put-domain-permissions-policy \  
  --domain test-domain \  
  --policy-document file://PATH/TO/policy.json
```

出力:

```
{
  "policy": {
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:domain/test-
domain",
    "document": "{ ...policy document content...}",
    "revision": "MQ1yyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxxx="
  }
}
```

詳細については、「ユーザーガイド」の「[ドメインポリシーの設定](#)」を参照してください。
AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス[PutDomainPermissionsPolicy](#)」の「」を参照してください。

put-repository-permissions-policy

次のコード例は、put-repository-permissions-policy を使用する方法を示しています。

AWS CLI

アクセス許可ポリシーをリポジトリにアタッチするには

次のput-repository-permissions-policy例では、policy.json ファイルで定義されているアクセス許可ポリシーを test-repo という名前のリポジトリにアタッチします。

```
aws codeartifact put-repository-permissions-policy \
  --domain test-domain \
  --repository test-repo \
  --policy-document file://PATH/T0/policy.json
```

出力:

```
{
  "policy": {
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:repository/test-
domain/test-repo",
    "document": "{ ...policy document content...}",
    "revision": "MQ1yyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxxx="
  }
}
```

詳細については、AWS CodeArtifact 「ユーザーガイド」の [「ポリシーの設定」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutRepositoryPermissionsPolicy](#)」の「」を参照してください。

update-package-versions-status

次のコード例は、update-package-versions-status を使用する方法を示しています。

AWS CLI

パッケージバージョンのステータスを更新するには

次のupdate-package-versions-status例では、テストパッケージパッケージのバージョン 4.0.0 のステータスを Archived に更新します。

```
aws codeartifact update-package-versions-status \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --versions 4.0.0 \  
  --target-status Archived
```

出力:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Archived"  
    }  
  },  
  "failedVersions": {}  
}
```

詳細については、「ユーザーガイド」の [「パッケージのバージョンステータスの更新」](#) を参照してください。AWS CodeArtifact

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdatePackageVersionsStatus](#)」の「」を参照してください。

update-repository

次のコード例は、update-repository を使用する方法を示しています。

AWS CLI

リポジトリを更新するには

次のupdate-repository例では、test-domain という名前のドメインの test-repo という名前のリポジトリの説明を「これは更新された説明」に更新します。

```
aws codeartifact update-repository \  
  --domain test-domain \  
  --repository test-repo \  
  --description "this is an updated description"
```

出力:

```
{  
  "repository": {  
    "name": "test-repo",  
    "administratorAccount": "111122223333",  
    "domainName": "test-domain",  
    "domainOwner": "111122223333",  
    "arn": "arn:aws:codeartifact:us-west-2:111122223333:repository/test-domain/  
test-repo",  
    "description": "this is an updated description",  
    "upstreams": [],  
    "externalConnections": []  
  }  
}
```

詳細については、AWS CodeArtifact [「ユーザーガイド」の「リポジトリ設定の表示または変更」](#)を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスUpdateRepository」](#)の「」を参照してください。

CodeBuild を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CodeBuild。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-delete-builds

次のコード例は、batch-delete-builds を使用する方法を示しています。

AWS CLI

でビルドを削除するには AWS CodeBuild。

次のbatch-delete-builds例では、指定された CodeBuild を使用して のビルドを削除します IDs。

```
aws codebuild batch-delete-builds --ids my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE
```

出力:

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE"
  ]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[ビルドの削除 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDeleteBuilds](#)」の「」を参照してください。

batch-get-build-batches

次の例は、batch-get-build-batches を使用する方法を説明しています。

AWS CLI

でビルドの詳細を表示するには AWS CodeBuild。

次のbatch-get-build-batches例では、指定された CodeBuild を使用して でビルドバッチに関する情報を取得しますIDs。

```
aws codebuild batch-get-build-batches \  
  --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE
```

出力:

```
{  
  "buildBatches": [  
    {  
      "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",  
      "arn": "arn:aws:codebuild:us-west-2:123456789012:build-batch/codebuild-  
demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",  
      "startTime": "2020-11-03T21:52:20.775000+00:00",  
      "endTime": "2020-11-03T21:56:59.784000+00:00",  
      "currentPhase": "SUCCEEDED",  
      "buildBatchStatus": "SUCCEEDED",  
      "resolvedSourceVersion": "0a6546f68309560d08a310daac92314c4d378f6b",  
      "projectName": "codebuild-demo-project",  
      "phases": [  
        {  
          "phaseType": "SUBMITTED",  
          "phaseStatus": "SUCCEEDED",  
          "startTime": "2020-11-03T21:52:20.775000+00:00",  
          "endTime": "2020-11-03T21:52:20.976000+00:00",  
          "durationInSeconds": 0  
        },  
        {
```

```
    "phaseType": "DOWNLOAD_BATCHSPEC",
    "phaseStatus": "SUCCEEDED",
    "startTime": "2020-11-03T21:52:20.976000+00:00",
    "endTime": "2020-11-03T21:52:57.401000+00:00",
    "durationInSeconds": 36
  },
  {
    "phaseType": "IN_PROGRESS",
    "phaseStatus": "SUCCEEDED",
    "startTime": "2020-11-03T21:52:57.401000+00:00",
    "endTime": "2020-11-03T21:56:59.751000+00:00",
    "durationInSeconds": 242
  },
  {
    "phaseType": "COMBINE_ARTIFACTS",
    "phaseStatus": "SUCCEEDED",
    "startTime": "2020-11-03T21:56:59.751000+00:00",
    "endTime": "2020-11-03T21:56:59.784000+00:00",
    "durationInSeconds": 0
  },
  {
    "phaseType": "SUCCEEDED",
    "startTime": "2020-11-03T21:56:59.784000+00:00"
  }
],
"source": {
  "type": "GITHUB",
  "location": "https://github.com/my-repo/codebuild-demo-project.git",
  "gitCloneDepth": 1,
  "gitSubmodulesConfig": {
    "fetchSubmodules": false
  },
  "reportBuildStatus": false,
  "insecureSsl": false
},
"secondarySources": [],
"secondarySourceVersions": [],
"artifacts": {
  "location": ""
},
"secondaryArtifacts": [],
"cache": {
  "type": "NO_CACHE"
},
}
```

```
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
  "computeType": "BUILD_GENERAL1_SMALL",
  "environmentVariables": [],
  "privilegedMode": false,
  "imagePullCredentialsType": "CODEBUILD"
},
"logConfig": {
  "cloudWatchLogs": {
    "status": "ENABLED"
  },
  "s3Logs": {
    "status": "DISABLED",
    "encryptionDisabled": false
  }
},
"buildTimeoutInMinutes": 60,
"queuedTimeoutInMinutes": 480,
"complete": true,
"initiator": "Strohm",
"encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
"buildBatchNumber": 6,
"buildBatchConfig": {
  "serviceRole": "arn:aws:iam::123456789012:role/service-role/
codebuild-demo-project",
  "restrictions": {
    "maximumBuildsAllowed": 100
  },
  "timeoutInMins": 480
},
"buildGroups": [
  {
    "identifier": "DOWNLOAD_SOURCE",
    "ignoreFailure": false,
    "currentBuildSummary": {
      "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:379737d8-bc35-48ec-97fd-776d27545315",
      "requestedOn": "2020-11-03T21:52:21.394000+00:00",
      "buildStatus": "SUCCEEDED",
      "primaryArtifact": {
        "type": "no_artifacts",
        "identifier": "DOWNLOAD_SOURCE"
      }
    }
  },

```



```
        "secondaryArtifacts": []
      }
    },
    {
      "identifier": "linux_small",
      "dependsOn": [],
      "ignoreFailure": false,
      "currentBuildSummary": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:dd785171-ed84-4bb6-8ede-ceeb86e54bdb",
        "requestedOn": "2020-11-03T21:52:57.604000+00:00",
        "buildStatus": "SUCCEEDED",
        "primaryArtifact": {
          "type": "no_artifacts",
          "identifier": "linux_small"
        },
        "secondaryArtifacts": []
      }
    },
    {
      "identifier": "linux_medium",
      "dependsOn": [
        "linux_small"
      ],
      "ignoreFailure": false,
      "currentBuildSummary": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:97cf7bd4-5313-4786-8243-4aef350a1267",
        "requestedOn": "2020-11-03T21:54:18.474000+00:00",
        "buildStatus": "SUCCEEDED",
        "primaryArtifact": {
          "type": "no_artifacts",
          "identifier": "linux_medium"
        },
        "secondaryArtifacts": []
      }
    },
    {
      "identifier": "linux_large",
      "dependsOn": [
        "linux_medium"
      ],
      "ignoreFailure": false,
      "currentBuildSummary": {
```

```

        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/
codebuild-demo-project:60a194cd-0d03-4337-9db1-d41476a17d27",
        "requestedOn": "2020-11-03T21:55:39.203000+00:00",
        "buildStatus": "SUCCEEDED",
        "primaryArtifact": {
            "type": "no_artifacts",
            "identifier": "linux_large"
        },
        "secondaryArtifacts": []
    },
    ]
}
],
"buildBatchesNotFound": []
}

```

詳細については、AWS CodeBuild 「ユーザーガイド」の AWS CodeBuild 「<<https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>>」の「バッチビルド」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchGetBuildBatches](#)」の「」を参照してください。

batch-get-builds

次のコード例は、batch-get-builds を使用する方法を示しています。

AWS CLI

でビルドの詳細を表示するには AWS CodeBuild。

次のbatch-get-builds例では、指定された CodeBuild を使用した のビルドに関する情報を取得しますIDs。

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE
```

出力:

```
{
  "buildsNotFound": [],

```

```
"builds": [
  {
    "artifacts": {
      "md5sum": "0e95edf915048a0c22efe6d139fff837",
      "location": "arn:aws:s3:::codepipeline-us-west-2-820783811474/CodeBuild-Python-Pip/BuildArtif/6DJsqQa",
      "encryptionDisabled": false,
      "sha256sum":
"cfaf0df33a090966a737f64ae4fe498969fdc842a0c9aec540bf93c37ac0d05a2"
    },
    "logs": {
      "cloudWatchLogs": {
        "status": "ENABLED"
      },
      "s3Logs": {
        "status": "DISABLED"
      },
      "streamName": "46472baf-8f6b-43c2-9255-b3b963af2732",
      "groupName": "/aws/codebuild/codebuild-demo-project",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=46472baf-8f6b-43c2-9255-b3b963af2732"
    },
    "timeoutInMinutes": 60,
    "environment": {
      "privilegedMode": false,
      "computeType": "BUILD_GENERAL1_MEDIUM",
      "image": "aws/codebuild/windows-base:1.0",
      "environmentVariables": [],
      "type": "WINDOWS_CONTAINER"
    },
    "projectName": "codebuild-demo-project",
    "buildComplete": true,
    "source": {
      "gitCloneDepth": 1,
      "insecureSsl": false,
      "type": "CODEPIPELINE"
    },
    "buildStatus": "SUCCEEDED",
    "secondaryArtifacts": [],
    "phases": [
      {
        "durationInSeconds": 0,
        "startTime": 1548717462.122,
```

```
    "phaseType": "SUBMITTED",
    "endTime": 1548717462.484,
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 0,
    "startTime": 1548717462.484,
    "phaseType": "QUEUED",
    "endTime": 1548717462.775,
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 34,
    "endTime": 1548717496.909,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717462.775,
    "phaseType": "PROVISIONING",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 15,
    "endTime": 1548717512.555,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ],
    "startTime": 1548717496.909,
    "phaseType": "DOWNLOAD_SOURCE",
    "phaseStatus": "SUCCEEDED"
  },
  {
    "durationInSeconds": 0,
    "endTime": 1548717512.734,
    "contexts": [
      {
        "statusCode": "",
        "message": ""
      }
    ]
  }
]
```

```
    }
  ],
  "startTime": 1548717512.555,
  "phaseType": "INSTALL",
  "phaseStatus": "SUCCEEDED"
},
{
  "durationInSeconds": 0,
  "endTime": 1548717512.924,
  "contexts": [
    {
      "statusCode": "",
      "message": ""
    }
  ],
  "startTime": 1548717512.734,
  "phaseType": "PRE_BUILD",
  "phaseStatus": "SUCCEEDED"
},
{
  "durationInSeconds": 9,
  "endTime": 1548717522.254,
  "contexts": [
    {
      "statusCode": "",
      "message": ""
    }
  ],
  "startTime": 1548717512.924,
  "phaseType": "BUILD",
  "phaseStatus": "SUCCEEDED"
},
{
  "durationInSeconds": 3,
  "endTime": 1548717525.498,
  "contexts": [
    {
      "statusCode": "",
      "message": ""
    }
  ],
  "startTime": 1548717522.254,
  "phaseType": "POST_BUILD",
  "phaseStatus": "SUCCEEDED"
}
```

```
    },
    {
      "durationInSeconds": 9,
      "endTime": 1548717534.646,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548717525.498,
      "phaseType": "UPLOAD_ARTIFACTS",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 2,
      "endTime": 1548717536.846,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548717534.646,
      "phaseType": "FINALIZING",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "startTime": 1548717536.846,
      "phaseType": "COMPLETED"
    }
  ],
  "startTime": 1548717462.122,
  "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
  "initiator": "codepipeline/CodeBuild-Pipeline",
  "secondarySources": [],
  "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-service-role",
  "currentPhase": "COMPLETED",
  "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
  "cache": {
    "type": "NO_CACHE"
  },
},
```

```

    "sourceVersion": "arn:aws:s3:::codepipeline-us-west-2-820783811474/
CodeBuild-Python-Pip/SourceArti/1TspnN3.zip",
    "endTime": 1548717536.846,
    "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
    "queuedTimeoutInMinutes": 480,
    "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
  },
  {
    "artifacts": {
      "md5sum": "",
      "overrideArtifactName": false,
      "location": "arn:aws:s3:::my-artifacts/codebuild-demo-project",
      "encryptionDisabled": false,
      "sha256sum": ""
    },
    "logs": {
      "cloudWatchLogs": {
        "status": "ENABLED"
      },
      "s3Logs": {
        "status": "DISABLED"
      },
      "streamName": "4dea3ca4-20ec-4898-b22a-a9eb9292775d",
      "groupName": "/aws/codebuild/codebuild-demo-project",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=4dea3ca4-20ec-4898-b22a-a9eb9292775d"
    },
    "timeoutInMinutes": 60,
    "environment": {
      "privilegedMode": false,
      "computeType": "BUILD_GENERAL1_MEDIUM",
      "image": "aws/codebuild/windows-base:1.0",
      "environmentVariables": [],
      "type": "WINDOWS_CONTAINER"
    },
    "projectName": "codebuild-demo-project",
    "buildComplete": true,
    "source": {
      "gitCloneDepth": 1,
      "location": "https://github.com/my-repo/codebuild-demo-project.git",
      "insecureSsl": false,
      "reportBuildStatus": false,

```

```
    "type": "GITHUB"
  },
  "buildStatus": "SUCCEEDED",
  "secondaryArtifacts": [],
  "phases": [
    {
      "durationInSeconds": 0,
      "startTime": 1548716241.89,
      "phaseType": "SUBMITTED",
      "endTime": 1548716242.241,
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 0,
      "startTime": 1548716242.241,
      "phaseType": "QUEUED",
      "endTime": 1548716242.536,
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 33,
      "endTime": 1548716276.171,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548716242.536,
      "phaseType": "PROVISIONING",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 15,
      "endTime": 1548716291.809,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548716276.171,
      "phaseType": "DOWNLOAD_SOURCE",
      "phaseStatus": "SUCCEEDED"
    }
  ]
}
```



```
    },
    {
      "durationInSeconds": 0,
      "endTime": 1548716291.993,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548716291.809,
      "phaseType": "INSTALL",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 0,
      "endTime": 1548716292.191,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548716291.993,
      "phaseType": "PRE_BUILD",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 9,
      "endTime": 1548716301.622,
      "contexts": [
        {
          "statusCode": "",
          "message": ""
        }
      ],
      "startTime": 1548716292.191,
      "phaseType": "BUILD",
      "phaseStatus": "SUCCEEDED"
    },
    {
      "durationInSeconds": 3,
      "endTime": 1548716304.783,
      "contexts": [
```

```
        {
            "statusCode": "",
            "message": ""
        }
    ],
    "startTime": 1548716301.622,
    "phaseType": "POST_BUILD",
    "phaseStatus": "SUCCEEDED"
},
{
    "durationInSeconds": 8,
    "endTime": 1548716313.775,
    "contexts": [
        {
            "statusCode": "",
            "message": ""
        }
    ],
    "startTime": 1548716304.783,
    "phaseType": "UPLOAD_ARTIFACTS",
    "phaseStatus": "SUCCEEDED"
},
{
    "durationInSeconds": 2,
    "endTime": 1548716315.935,
    "contexts": [
        {
            "statusCode": "",
            "message": ""
        }
    ],
    "startTime": 1548716313.775,
    "phaseType": "FINALIZING",
    "phaseStatus": "SUCCEEDED"
},
{
    "startTime": 1548716315.935,
    "phaseType": "COMPLETED"
}
],
"startTime": 1548716241.89,
"secondarySourceVersions": [],
"initiator": "my-codebuild-project",
```

```

    "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
    "currentPhase": "COMPLETED",
    "id": "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
    "cache": {
      "type": "NO_CACHE"
    },
    "endTime": 1548716315.935,
    "secondarySources": [],
    "queuedTimeoutInMinutes": 480,
    "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
  }
]
}

```

詳細については、「ユーザーガイド」の「[ビルドの詳細の表示 \(AWS CLI\)](#)」を参照してください。AWS CodeBuild

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetBuilds](#)」の「」を参照してください。

batch-get-projects

次のコード例は、batch-get-projects を使用する方法を示しています。

AWS CLI

AWS CodeBuild ビルドプロジェクト名のリストを取得するには。

次のbatch-get-projects例では、名前指定された CodeBuild ビルドプロジェクトのリストを取得します。

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-
project2 my-other-demo-project
```

次の出力では、projectsNotFound配列は、指定されたが見つからないビルドプロジェクト名を一覧表示します。projects 配列は、情報が見つかった各ビルドプロジェクトの詳細を示しています。

```

{
  "projectsNotFound": [],
  "projects": [
    {
      "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
      "name": "codebuild-demo-project2",
      "queuedTimeoutInMinutes": 480,
      "timeoutInMinutes": 60,
      "source": {
        "buildspec": "version: 0.2\n\n#env:\n #variables:\n      # key:\n\n\"value\"\n      # key: \"value\"\n #parameter-store:\n      # key: \"value\"\n\n# key: \"value\"\n\n#phases:\n #install:\n      #commands:\n      # - command\n      # - command\n #pre_build:\n      #commands:\n      # - command\n      # - command\n\n build:\n      commands:\n      # - command\n      # - command\n #post_build:\n      #commands:\n      # - command\n      # - command\n#artifacts:\n #files:\n      # - location\n      # - location\n #name: $(date +%Y-%m-%d)\n #discard-paths: yes\n #base-directory: location\n#cache:\n #paths:\n      # - paths",
        "type": "NO_SOURCE",
        "insecureSsl": false,
        "gitCloneDepth": 1
      },
      "artifacts": {
        "type": "NO_ARTIFACTS"
      },
      "badge": {
        "badgeEnabled": false
      },
      "lastModified": 1540588091.108,
      "created": 1540588091.108,
      "arn": "arn:aws:codebuild:us-west-2:123456789012:project/test-for-sample",
      "secondarySources": [],
      "secondaryArtifacts": [],
      "cache": {
        "type": "NO_CACHE"
      },
      "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-test-role",
      "environment": {
        "image": "aws/codebuild/java:openjdk-8",
        "privilegedMode": true,
        "type": "LINUX_CONTAINER",
        "computeType": "BUILD_GENERAL1_SMALL",

```

```
        "environmentVariables": [],
    },
    "tags": [],
},
{
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "name": "my-other-demo-project",
    "queuedTimeoutInMinutes": 480,
    "timeoutInMinutes": 60,
    "source": {
        "location": "https://github.com/iversonic/codedeploy-sample.git",
        "reportBuildStatus": false,
        "buildspec": "buildspec.yml",
        "insecureSsl": false,
        "gitCloneDepth": 1,
        "type": "GITHUB",
        "auth": {
            "type": "OAUTH"
        }
    },
    "artifacts": {
        "type": "NO_ARTIFACTS"
    },
    "badge": {
        "badgeEnabled": false
    },
    "lastModified": 1523401711.73,
    "created": 1523401711.73,
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/Project2",
    "cache": {
        "type": "NO_CACHE"
    },
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/codebuild-Project2-service-role",
    "environment": {
        "image": "aws/codebuild/nodejs:4.4.7",
        "privilegedMode": false,
        "type": "LINUX_CONTAINER",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": []
    },
    "tags": []
}
]
```

```
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[ビルドプロジェクトの詳細を表示する \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetProjects](#)」の「」を参照してください。

batch-get-report-groups

次のコード例は、batch-get-report-groups を使用する方法を示しています。

AWS CLI

で 1 つ以上のレポートグループに関する情報を取得するには AWS CodeBuild。

次のbatch-get-report-groups例では、指定された を持つレポートグループに関する情報を取得しますARN。

```
aws codebuild batch-get-report-groups \  
  --report-group-arns arn:aws:codebuild:<region-ID>:<user-ID>:report-group/  
  <report-group-name>
```

出力:

```
{  
  "reportGroups": [  
    {  
      "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-  
group-name>",  
      "name": "report-group-name",  
      "type": "TEST",  
      "exportConfig": {  
        "exportConfigType": "NO_EXPORT"  
      },  
      "created": "2020-10-01T18:04:08.466000+00:00",  
      "lastModified": "2020-10-01T18:04:08.466000+00:00",  
      "tags": []  
    }  
  ],  
  "reportGroupsNotFound": []  
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートグループの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchGetReportGroups](#)」の「」を参照してください。

batch-get-reports

次の例は、batch-get-reports を使用する方法を説明しています。

AWS CLI

で 1 つ以上のレポートに関する情報を取得するには AWS CodeBuild。

次のbatch-get-reports例では、指定された を持つレポートに関する情報を取得します ARNs。

```
aws codebuild batch-get-reports \  
  --report-arns arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report 1 ID> arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report 2 ID>
```

出力:

```
{  
  "reports": [  
    {  
      "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report 1 ID>",  
      "type": "TEST",  
      "name": "<report-group-name>",  
      "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>",  
      "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-reports:<ID>",  
      "status": "FAILED",  
      "created": "2020-10-01T11:25:22.531000-07:00",  
      "expired": "2020-10-31T11:25:22-07:00",  
      "exportConfig": {  
        "exportConfigType": "NO_EXPORT"  
      },  
      "truncated": false,  
      "testSummary": {
```

```
        "total": 28,
        "statusCounts": {
            "ERROR": 5,
            "FAILED": 1,
            "SKIPPED": 4,
            "SUCCEEDED": 18,
            "UNKNOWN": 0
        },
        "durationInNanoSeconds": 94000000
    }
},
{
    "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-
name>:<report 2 ID>",
    "type": "TEST",
    "name": "<report-group-name>",
    "reportGroupArn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>",
    "executionId": "arn:aws:codebuild:<region-ID>:<user-ID>:build/test-
reports:<ID>",
    "status": "FAILED",
    "created": "2020-10-01T11:13:05.816000-07:00",
    "expired": "2020-10-31T11:13:05-07:00",
    "exportConfig": {
        "exportConfigType": "NO_EXPORT"
    },
    "truncated": false,
    "testSummary": {
        "total": 28,
        "statusCounts": {
            "ERROR": 5,
            "FAILED": 1,
            "SKIPPED": 4,
            "SUCCEEDED": 18,
            "UNKNOWN": 0
        },
        "durationInNanoSeconds": 94000000
    }
}
],
"reportsNotFound": []
}
```


詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchGetReports](#)」の「」を参照してください。

create-project

次の例は、create-project を使用する方法を説明しています。

AWS CLI

例 1: AWS CodeBuild ビルドプロジェクトを作成するには

次のcreate-project例では、S3 バケットのソースファイルを使用して CodeBuild ビルドプロジェクトを作成します。

```
aws codebuild create-project \
  --name "my-demo-project" \
  --source "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-  
input-bucket/my-source.zip\"}" \
  --artifacts "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-  
output-bucket\"}" \
  --environment "{\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/  
standard:1.0\", \"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
  --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role"
```

出力:

```
{
  "project": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
    "name": "my-cli-demo-project",
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role",
    "lastModified": 1556839783.274,
    "badge": {
      "badgeEnabled": false
    },
    "queuedTimeoutInMinutes": 480,
    "environment": {
```

```
    "image": "aws/codebuild/standard:1.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "type": "LINUX_CONTAINER",
    "imagePullCredentialsType": "CODEBUILD",
    "privilegedMode": false,
    "environmentVariables": []
  },
  "artifacts": {
    "location": "codebuild-us-west-2-123456789012-output-bucket",
    "name": "my-cli-demo-project",
    "namespaceType": "NONE",
    "type": "S3",
    "packaging": "NONE",
    "encryptionDisabled": false
  },
  "source": {
    "type": "S3",
    "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
    "insecureSsl": false
  },
  "timeoutInMinutes": 60,
  "cache": {
    "type": "NO_CACHE"
  },
  "created": 1556839783.274
}
}
```

例 2: パラメータJSONの入カファイルを使用して AWS CodeBuild ビルドプロジェクトを作成するには

次のcreate-project例では、JSON入カファイルに必要なパラメータをすべて渡すことで CodeBuild ビルドプロジェクトを作成します。--generate-cli-skeleton parameterのみを含むコマンドを実行して、入カファイルテンプレートを作成します。

```
aws codebuild create-project --cli-input-json file://create-project.json
```

入カJSONファイルcreate-project.jsonには、次の内容が含まれます。

```
{
  "name": "codebuild-demo-project",
```

```

"source": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
},
"artifacts": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-output-bucket"
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:1.0",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "serviceIAMRole"
}

```

出力:

```

{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",
      "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:1.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/
MessageUtil.zip"
    },
  },
}

```

```
"encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
  "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project"
}
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[ビルドプロジェクトの作成 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProject](#)」の「」を参照してください。

create-report-group

次のコード例は、create-report-group を使用する方法を示しています。

AWS CLI

でレポートグループを作成するには AWS CodeBuild。

次のcreate-report-group例では、新しいレポートグループを作成します。

```
aws codebuild create-report-group \
  --cli-input-json file://create-report-group-source.json
```

create-report-group-source.json の内容 :

```
{
  "name": "cli-created-report-group",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "my-s3-bucket",
      "path": "",
      "packaging": "ZIP",
      "encryptionDisabled": true
    }
  }
}
```

出力:

```
{
  "reportGroup": {
    "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-group",
    "name": "cli-created-report-group",
    "type": "TEST",
    "exportConfig": {
      "exportConfigType": "S3",
      "s3Destination": {
        "bucket": "my-s3-bucket",
        "path": "",
        "packaging": "ZIP",
        "encryptionDisabled": true
      }
    },
    "created": 1602020026.775,
    "lastModified": 1602020026.775
  }
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートグループの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateReportGroup](#)」の「」を参照してください。

create-webhook

次の例は、create-webhook を使用する方法を説明しています。

AWS CLI

AWS CodeBuild プロジェクトのウェブフックフィルターを作成するには

次のcreate-webhook例では、2つのフィルターグループを持つという名前my-projectのCodeBuildプロジェクトのウェブフックを作成します。最初のフィルターグループでは、正規表現`^refs/heads/master$`と一致するGit参照名および`^refs/heads/myBranch$`と一致するヘッド参照を持つブランチに対してプルリクエストを作成、更新、または再開することを指定します。2番目のフィルターグループは、正規表現と一致しないGit参照名を持つブランチへのプッシュリクエストを指定します`^refs/heads/myBranch$`。

```
aws codebuild create-webhook \
```

```
--project-name my-project \
--filter-groups "[[{"type":"EVENT","\pattern":"PULL_REQUEST_CREATED,
PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED"}, {"type":"HEAD_REF","\pattern
":"^refs/heads/myBranch$","\excludeMatchedPattern":true}, {"type":"BASE_REF
","\pattern":"^refs/heads/master$","\excludeMatchedPattern":true}], [{"type":"
EVENT","\pattern":"PUSH"}, {"type":"HEAD_REF","\pattern":"^refs/heads/
myBranch$","\excludeMatchedPattern":true}]"]
```

出力:

```
{
  "webhook": {
    "payloadUrl": "https://codebuild.us-west-2.amazonaws.com/webhooks?
t=eyJlbnNyeXB0ZWREYXRhIjoiVV15MGtoeGRwSzZFRX12Wnh4bld1Z0tKZ291TVpQNEtFamQ3RD1DYWpRaGIreVFrdm
    "url": "https://api.github.com/repos/iversonic/codedeploy-sample/
hooks/105190656",
    "lastModifiedSecret": 1556311319.069,
    "filterGroups": [
      [
        {
          "type": "EVENT",
          "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED",
          "excludeMatchedPattern": false
        },
        {
          "type": "HEAD_REF",
          "pattern": "refs/heads/myBranch$",
          "excludeMatchedPattern": true
        },
        {
          "type": "BASE_REF",
          "pattern": "refs/heads/master$",
          "excludeMatchedPattern": true
        }
      ],
      [
        {
          "type": "EVENT",
          "pattern": "PUSH",
          "excludeMatchedPattern": false
        },
        {
```

```

        "type": "HEAD_REF",
        "pattern": "refs/heads/myBranch$",
        "excludeMatchedPattern": true
      }
    ]
  }
}

```

詳細については、「ユーザーガイド」の [GitHub「ウェブフックイベントをフィルタリングする \(SDK\)」](#) を参照してください。AWS CodeBuild

- API 詳細については、AWS CLI「コマンドリファレンス [CreateWebhook](#)」の「」を参照してください。

delete-build-batch

次の例は、delete-build-batch を使用する方法を説明しています。

AWS CLI

でバッチビルドを削除するには AWS CodeBuild。

次のdelete-build-batch例では、指定されたバッチビルドを削除します。

```

aws codebuild delete-build-batch \
  --id <project-name>:<batch-ID>

```

出力:

```

{
  "statusCode": "BATCH_DELETED",
  "buildsDeleted": [
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>",
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>",
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>",
    "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>"
  ],
  "buildsNotDeleted": []
}

```

```
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[のバッチビルド AWS CodeBuild](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBuildBatch](#)」の「」を参照してください。

delete-project

次のコード例は、delete-project を使用する方法を示しています。

AWS CLI

AWS CodeBuild ビルドプロジェクトを削除するには

次のdelete-project例では、指定された CodeBuild ビルドプロジェクトを削除します。

```
aws codebuild delete-project --name my-project
```

このコマンドでは何も出力されません。

詳細については、AWS CodeBuild 「ユーザーガイド」の「[ビルドプロジェクトの削除 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProject](#)」の「」を参照してください。

delete-report-group

次の例は、delete-report-group を使用する方法を説明しています。

AWS CLI

でレポートグループを削除するには AWS CodeBuild。

次のdelete-report-group例では、指定された を持つレポートグループを削除しますARN。

```
aws codebuild delete-report-group \  
  --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

このコマンドでは何も出力されません。

詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートグループの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteReportGroup](#)」の「」を参照してください。

delete-report

次のコード例は、delete-report を使用する方法を示しています。

AWS CLI

でレポートを削除するには AWS CodeBuild。

次のdelete-report例では、指定されたレポートを削除します。

```
aws codebuild delete-report \  
  --arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

このコマンドでは何も出力されません。

詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteReport](#)」の「」を参照してください。

delete-source-credentials

次の例は、delete-source-credentials を使用する方法を説明しています。

AWS CLI

ソースプロバイダーから切断し、そのアクセストークンを削除するには。

次のdelete-source-credentials例では、ソースプロバイダーから切断し、トークンを削除します。ソースプロバイダーへの接続に使用されるソース認証情報ARNのは、どのソース認証情報かを決定します。

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

出力:

```
{
  "arn": "arn:aws:codebuild:your-region:your-account-id:token/your-server-type"
}
```

詳細については、[「ユーザーガイド」の「アクセストークンを使用したソースプロバイダーの接続 \(CLI\)」](#)を参照してください。AWS CodeBuild

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSourceCredentials](#)」の「」を参照してください。

delete-webhook

次のコード例は、delete-webhook を使用する方法を示しています。

AWS CLI

AWS CodeBuild プロジェクトからウェブフックフィルターを削除するには

次のdelete-webhook例では、指定した CodeBuild プロジェクトからウェブフックを削除します。

```
aws codebuild delete-webhook --project-name my-project
```

このコマンドでは何も出力されません。

詳細については、AWS CodeBuild 「ユーザーガイド」の[「ビルドの自動実行の停止 \(AWS CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteWebhook](#)」の「」を参照してください。

describe-code-coverages

次のコード例は、describe-code-coverages を使用する方法を示しています。

AWS CLI

でコードカバレッジテスト結果に関する詳細情報を取得するには AWS CodeBuild。

次のdescribe-code-coverages例では、指定されたレポートのコードカバレッジテスト結果に関する情報を取得します。

```
aws codebuild describe-code-coverages \  
  --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

出力:

```
{  
  "codeCoverages": [  
    {  
      "id": "20a0adcc-db13-4b66-804b-ecaf9f852855",  
      "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-group-name>:<report-ID>",  
      "filePath": "<source-file-1-path>",  
      "lineCoveragePercentage": 83.33,  
      "linesCovered": 5,  
      "linesMissed": 1,  
      "branchCoveragePercentage": 50.0,  
      "branchesCovered": 1,  
      "branchesMissed": 1,  
      "expired": "2020-11-20T21:22:45+00:00"  
    },  
    {  
      "id": "0887162d-bf57-4cf1-a164-e432373d1a83",  
      "reportARN": "arn:aws:codebuild:<region-ID>:972506530580:report/<report-group-name>:<report-ID>",  
      "filePath": "<source-file-2-path>",  
      "lineCoveragePercentage": 90.9,  
      "linesCovered": 10,  
      "linesMissed": 1,  
      "branchCoveragePercentage": 50.0,  
      "branchesCovered": 1,  
      "branchesMissed": 1,  
      "expired": "2020-11-20T21:22:45+00:00"  
    }  
  ]  
}
```

詳細については、「ユーザーガイド」の[「コードカバレッジレポート」](#)を参照してください。
AWS CodeBuild

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCodeCoverages](#)」の「」を参照してください。

describe-test-cases

次の例は、describe-test-cases を使用する方法を説明しています。

AWS CLI

のテストケースに関する詳細情報を取得するには AWS CodeBuild。

次のdescribe-test-cases例では、指定されたレポートのテストケースに関する情報を取得します。

```
aws codebuild describe-test-cases \  
  --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

出力:

```
{  
  "testCases": [  
    {  
      "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>",  
      "testRawDataPath": "<test-report-path>",  
      "prefix": "NUnit.Tests.Assemblies.MockTestFixture",  
      "name": "NUnit.Tests.Assemblies.MockTestFixture.NotRunnableTest",  
      "status": "ERROR",  
      "durationInNanoSeconds": 0,  
      "message": "No arguments were provided\n",  
      "expired": "2020-11-20T17:52:10+00:00"  
    },  
    {  
      "reportArn": "arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>",  
      "testRawDataPath": "<test-report-path>",  
      "prefix": "NUnit.Tests.Assemblies.MockTestFixture",  
      "name": "NUnit.Tests.Assemblies.MockTestFixture.TestWithException",  
      "status": "ERROR",  
      "durationInNanoSeconds": 0,  
    }  
  ]  
}
```

```
    "message": "System.ApplicationException : Intentional Exception
\nat NUnit.Tests.Assemblies.MockTestFixture.MethodThrowsException()\nat
NUnit.Tests.Assemblies.MockTestFixture.TestWithException()\n\n",
    "expired": "2020-11-20T17:52:10+00:00"
  }
]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「でのテストレポートの使用 AWS CodeBuild」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTestCases](#)」の「」を参照してください。

import-source-credentials

次のコード例は、import-source-credentials を使用する方法を示しています。

AWS CLI

ソースプロバイダーの認証情報をインポートして、AWS CodeBuild ユーザーをソースプロバイダーに接続します。

次のimport-source-credentials例では、認証タイプに BASIC_AUTH を使用する Bitbucket リポジトリのトークンをインポートします。

```
aws codebuild import-source-credentials --server-type BITBUCKET --auth-
type BASIC_AUTH --token my-Bitbucket-password --username my-Bitbucket-username
```

出力:

```
{
  "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket"
}
```

詳細については、[「ユーザーガイド」の「アクセストークンを使用してソースプロバイダーを接続する \(CLI\)」](#) を参照してください。AWS CodeBuild

- API 詳細については、AWS CLI 「コマンドリファレンス [ImportSourceCredentials](#)」の「」を参照してください。

invalidate-project-cache

次のコード例は、`invalidate-project-cache` を使用する方法を示しています。

AWS CLI

AWS CodeBuild ビルドプロジェクトのキャッシュをリセットするには。

次の`invalidate-project-cache`例では、指定された CodeBuild プロジェクトのキャッシュをリセットします。

```
aws codebuild invalidate-project-cache --project-name my-project
```

このコマンドでは何も出力されません。

詳細については、AWS CodeBuild 「ユーザーガイド」の [「でのキャッシュの構築 CodeBuild」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [InvalidateProjectCache](#)」の「」を参照してください。

list-build-batches-for-project

次のコード例は、`list-build-batches-for-project` を使用する方法を示しています。

AWS CLI

で特定のビルドプロジェクトのバッチビルドを一覧表示するには AWS CodeBuild。

次の`list-build-batches-for-project`例では、指定されたプロジェクトの CodeBuild バッチビルドを一覧表示します。

```
aws codebuild list-build-batches-for-project \  
  --project-name "<project-name>"
```

出力:

```
{  
  "ids": [  
    "<project-name>:<batch-ID>",  
    "<project-name>:<batch-ID>"  
  ]  
}
```

```
]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[のバッチビルド AWS CodeBuild](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBuildBatchesForProject](#)」の「」を参照してください。

list-build-batches

次の例は、list-build-batches を使用する方法を説明しています。

AWS CLI

でバッチビルドを一覧表示するには AWS CodeBuild。

次のlist-build-batches例では、現在のアカウントの CodeBuild バッチビルドを一覧表示します。

```
aws codebuild list-build-batches
```

出力:

```
{
  "ids": [
    "<project-name>:<batch-ID>",
    "<project-name>:<batch-ID>"
  ]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の AWS CodeBuild 「< <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>>」のバッチビルド」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBuildBatches](#)」の「」を参照してください。

list-builds-for-project

次の例は、list-builds-for-project を使用する方法を説明しています。

AWS CLI

ビルドプロジェクトの AWS CodeBuild ビルドのリストを表示するには。

次のlist-builds-for-project例では、指定されたビルドプロジェクトの CodeBuild ビルドを降順IDsで一覧表示します。

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order DESCENDING
```

出力:

```
{
  "ids": [
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-11111example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-22222example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-33333example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-44444example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-55555example"
  ]
}
```

詳細については、AWS CodeBuild 「[ユーザーガイド](#)」の「[ビルドプロジェクト \(AWS CLI\) IDs のビルドのリストを表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListBuildsForProject](#)」の「」を参照してください。

list-builds

次のコード例は、list-builds を使用する方法を示しています。

AWS CLI

AWS CodeBuild ビルド のリストを取得するにはIDs。

次のlist-builds例では、昇順でソートされたリスト CodeBuild IDsを取得します。

```
aws codebuild list-builds --sort-order ASCENDING
```

出力には、使用可能な出力が多いことを示すnextToken値が含まれます。


```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for
  brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

このコマンドを再度実行し、前のレスポンスのnextToken値をパラメータとして指定して、出力の次の部分を取得します。レスポンスにnextToken値が表示されなくなるまで繰り返します。

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

出力の次の部分：

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

詳細については、AWS CodeBuild 「[ユーザーガイド](#)」の「[ビルドのリストを表示する IDs \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListBuilds](#)」の「」を参照してください。

list-curated-environment-images

次のコード例は、list-curated-environment-images を使用する方法を示しています。

AWS CLI

ビルドに使用できる によって管理 AWS CodeBuild される Docker イメージのリストを取得するには。

次の `list-curated-environment-images` 例では、 が管理する CodeBuild ビルドに使用できる Docker イメージを一覧表示します。

```
aws codebuild list-curated-environment-images
```

出力:

```
{
  "platforms": [
    {
      "platform": "AMAZON_LINUX",
      "languages": [
        {
          "language": "JAVA",
          "images": [
            {
              "description": "AWS ElasticBeanstalk - Java 7 Running on
Amazon Linux 64bit v2.1.3",
              "name": "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3",
              "versions": [
                "aws/codebuild/eb-java-7-amazonlinux-64:2.1.3-1.0.0"
              ]
            },
            {
              "description": "AWS ElasticBeanstalk - Java 8 Running on
Amazon Linux 64bit v2.1.3",
              "name": "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3",
              "versions": [
                "aws/codebuild/eb-java-8-amazonlinux-64:2.1.3-1.0.0"
              ]
            },
            ... LIST TRUNCATED FOR BREVITY ...
          ]
        }
      ]
    }
  ]
}
```

```
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[が提供する Docker イメージ CodeBuild](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCuratedEnvironmentImages](#)」の「」を参照してください。

list-projects

次のコード例は、list-projects を使用する方法を示しています。

AWS CLI

AWS CodeBuild ビルドプロジェクト名のリストを取得するには。

次のlist-projects例では、CodeBuild ビルドプロジェクトのリストを名前で昇順にソートします。

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

出力には、使用可能な出力が多いことを示すnextToken値が含まれます。

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project99"
  ]
}
```

このコマンドを再度実行し、前のレスポンスのnextToken値をパラメータとして指定して、出力の次の部分を取得します。レスポンスにnextToken値が表示されなくなるまで繰り返します。

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

```
{
  "projects": [
    "codebuild-demo-project100",
    "codebuild-demo-project101",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project122"
  ]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の[「ビルドプロジェクト名 \(AWS CLI\) のリストを表示する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProjects](#)」の「」を参照してください。

list-report-groups

次の例は、list-report-groups を使用する方法を説明しています。

AWS CLI

ARNsでレポートグループのリストを取得するには AWS CodeBuild。

次のlist-report-groups例では、リージョン内のARNsアカウントのレポートグループを取得します。

```
aws codebuild list-report-groups
```

出力:

```
{
  "reportGroups": [
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
  ]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の[「レポートグループの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReportGroups](#)」の「」を参照してください。

list-reports-for-report-group

次の例は、list-reports-for-report-group を使用する方法を説明しています。

AWS CLI

のレポートグループ内のレポートのリストを取得するには AWS CodeBuild。

次のlist-report-for-report-groups例では、リージョン内のアカウントの指定されたレポートグループのレポートを取得します。

```
aws codebuild list-reports-for-report-group \  
  --report-group-arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-  
group-name>
```

出力:

```
{  
  "reports": [  
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-1",  
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-2",  
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/report-3"  
  ]  
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[レポートグループの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReportsForReportGroup](#)」の「」を参照してください。

list-reports

次のコード例は、list-reports を使用する方法を示しています。

AWS CLI

で現在のアカウントのレポートのリストを取得するには AWS CodeBuild。

次のlist-reports例では、現在のアカウントのARNsレポートの を取得します。

```
aws codebuild list-reports
```

出力:

```
{
  "reports": [
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report ID>",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report ID>",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report/<report-group-name>:<report ID>"
  ]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReports](#)」の「」を参照してください。

list-shared-projects

次の例は、list-shared-projects を使用する方法を説明しています。

AWS CLI

で共有プロジェクトを一覧表示するには AWS CodeBuild。

次のlist-shared-projects例では、現在のアカウントで利用可能な CodeBuild 共有プロジェクトを一覧表示します。

```
aws codebuild list-shared-projects
```

出力:

```
{
  "projects": [
    "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-name-1>",
  ]
}
```

```
    "arn:aws:codebuild:<region-ID>:<account-ID>:project/<shared-project-name-2>"
  ]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「共有プロジェクトの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListSharedProjects](#)」の「」を参照してください。

list-shared-report-groups

次のコード例は、list-shared-report-groups を使用する方法を示しています。

AWS CLI

ARNsで共有レポートグループのリストを取得するには AWS CodeBuild。

次のlist-shared-report-groups例では、リージョン内のARNsアカウントのレポートグループを取得します。

```
aws codebuild list-shared-report-groups
```

出力:

```
{
  "reportGroups": [
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-1",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-2",
    "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/report-group-3"
  ]
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートグループの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListSharedReportGroups](#)」の「」を参照してください。

list-source-credentials

次のコード例は、list-source-credentials を使用する方法を示しています。

AWS CLI

のリストを表示するには `sourceCredentialsObjects`

次の `list-source-credentials` 例では、1 つの Bitbucket AWS アカウントと 1 つのアカウントに接続されている GitHub アカウントのトークンを一覧表示します。レスポンス内の各 `sourceCredentialsInfos` オブジェクトには、接続されたソース認証情報情報が含まれています。

```
aws codebuild list-source-credentials
```

出力:

```
{
  "sourceCredentialsInfos": [
    {
      "serverType": "BITBUCKET",
      "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket",
      "authType": "BASIC_AUTH"
    },
    {
      "serverType": "GITHUB",
      "arn": "arn:aws:codebuild:us-west-2:123456789012:token/github",
      "authType": "OAUTH"
    }
  ]
}
```

詳細については、[「ユーザーガイド」の「アクセストークンを使用してソースプロバイダーを接続する \(CLI\)」](#)を参照してください。AWS CodeBuild

- API 詳細については、AWS CLI 「コマンドリファレンス [ListSourceCredentials](#)」の「」を参照してください。

retry-build-batch

次の例は、`retry-build-batch` を使用する方法を説明しています。

AWS CLI

で失敗したバッチビルドを再試行するには AWS CodeBuild。

次のretry-build-batch例では、指定されたバッチビルドを再起動します。

```
aws codebuild retry-build-batch \  
  --id <project-name>:<batch-ID>
```

出力:

```
{  
  "buildBatch": {  
    "id": "<project-name>:<batch-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-  
name>:<batch-ID>",  
    "startTime": "2020-10-21T17:26:23.099000+00:00",  
    "currentPhase": "SUBMITTED",  
    "buildBatchStatus": "IN_PROGRESS",  
    "resolvedSourceVersion": "3a9e11cb419e8fff14b03883dc4e64f6155aaa7e",  
    "projectName": "<project-name>",  
    "phases": [  
      {  
        "phaseType": "SUBMITTED",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T17:26:23.099000+00:00",  
        "endTime": "2020-10-21T17:26:23.457000+00:00",  
        "durationInSeconds": 0  
      },  
      {  
        "phaseType": "DOWNLOAD_BATCHSPEC",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T17:26:23.457000+00:00",  
        "endTime": "2020-10-21T17:26:54.902000+00:00",  
        "durationInSeconds": 31  
      },  
      {  
        "phaseType": "IN_PROGRESS",  
        "phaseStatus": "CLIENT_ERROR",  
        "startTime": "2020-10-21T17:26:54.902000+00:00",  
        "endTime": "2020-10-21T17:28:16.060000+00:00",  
        "durationInSeconds": 81  
      },  
      {  
        "phaseType": "FAILED",  
        "phaseStatus": "RETRY",  
        "startTime": "2020-10-21T17:28:16.060000+00:00",
```

```
        "endTime": "2020-10-21T17:29:39.709000+00:00",
        "durationInSeconds": 83
    },
    {
        "phaseType": "SUBMITTED",
        "startTime": "2020-10-21T17:29:39.709000+00:00"
    }
],
"source": {
    "type": "GITHUB",
    "location": "https://github.com/strohm-a/<project-name>-graph.git",
    "gitCloneDepth": 1,
    "gitSubmodulesConfig": {
        "fetchSubmodules": false
    },
    "reportBuildStatus": false,
    "insecureSsl": false
},
"secondarySources": [],
"secondarySourceVersions": [],
"artifacts": {
    "location": ""
},
"secondaryArtifacts": [],
"cache": {
    "type": "NO_CACHE"
},
"environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
},
"logConfig": {
    "cloudWatchLogs": {
        "status": "ENABLED"
    },
    "s3Logs": {
        "status": "DISABLED",
        "encryptionDisabled": false
    }
},
},
```

```

    "buildTimeoutInMinutes": 60,
    "queuedTimeoutInMinutes": 480,
    "complete": false,
    "initiator": "<username>",
    "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
    "buildBatchNumber": 4,
    "buildBatchConfig": {
      "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
      "restrictions": {
        "maximumBuildsAllowed": 100
      },
      "timeoutInMins": 480
    },
    "buildGroups": [
      {
        "identifier": "DOWNLOAD_SOURCE",
        "ignoreFailure": false,
        "currentBuildSummary": {
          "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
          "requestedOn": "2020-10-21T17:26:23.889000+00:00",
          "buildStatus": "SUCCEEDED",
          "primaryArtifact": {
            "type": "no_artifacts",
            "identifier": "DOWNLOAD_SOURCE"
          },
          "secondaryArtifacts": []
        },
        "dependsOn": [],
        "ignoreFailure": false,
        "currentBuildSummary": {
          "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
          "requestedOn": "2020-10-21T17:26:55.115000+00:00",
          "buildStatus": "FAILED",
          "primaryArtifact": {
            "type": "no_artifacts",
            "identifier": "linux_small"
          },
          "secondaryArtifacts": []
        }
      }
    ]
  }
}

```

```
    }
  },
  {
    "identifier": "linux_medium",
    "dependsOn": [
      "linux_small"
    ],
    "ignoreFailure": false,
    "currentBuildSummary": {
      "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
      "requestedOn": "2020-10-21T17:26:54.594000+00:00",
      "buildStatus": "STOPPED"
    }
  },
  {
    "identifier": "linux_large",
    "dependsOn": [
      "linux_medium"
    ],
    "ignoreFailure": false,
    "currentBuildSummary": {
      "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
      "requestedOn": "2020-10-21T17:26:54.701000+00:00",
      "buildStatus": "STOPPED"
    }
  }
]
}
```

詳細については、AWS CodeBuild [「ユーザーガイド」の「のバッチビルド AWS CodeBuild」](#) を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンス `RetryBuildBatch`」](#) の「」を参照してください。

retry-build

次のコード例は、`retry-build` を使用する方法を示しています。

AWS CLI

で失敗したビルドを再試行するには AWS CodeBuild。

次のretry-build例では、指定されたビルドを再起動します。

```
aws codebuild retry-build \  
  --id <project-name>:<build-ID>
```

出力:

```
{  
  "build": {  
    "id": "<project-name>:<build-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/<project-name>:<build-ID>",  
    "buildNumber": 9,  
    "startTime": "2020-10-21T17:51:38.161000+00:00",  
    "currentPhase": "QUEUED",  
    "buildStatus": "IN_PROGRESS",  
    "projectName": "<project-name>",  
    "phases": [  
      {  
        "phaseType": "SUBMITTED",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T17:51:38.161000+00:00",  
        "endTime": "2020-10-21T17:51:38.210000+00:00",  
        "durationInSeconds": 0  
      },  
      {  
        "phaseType": "QUEUED",  
        "startTime": "2020-10-21T17:51:38.210000+00:00"  
      }  
    ],  
    "source": {  
      "type": "GITHUB",  
      "location": "<GitHub-repo-URL>",  
      "gitCloneDepth": 1,  
      "gitSubmodulesConfig": {  
        "fetchSubmodules": false  
      },  
      "reportBuildStatus": false,  
      "insecureSsl": false  
    }  
  }  
}
```

```

    },
    "secondarySources": [],
    "secondarySourceVersions": [],
    "artifacts": {
      "location": ""
    },
    },
    "secondaryArtifacts": [],
    "cache": {
      "type": "NO_CACHE"
    },
    },
    "environment": {
      "type": "LINUX_CONTAINER",
      "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
      "computeType": "BUILD_GENERAL1_SMALL",
      "environmentVariables": [],
      "privilegedMode": false,
      "imagePullCredentialsType": "CODEBUILD"
    },
    "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-role-
name>",
    "logs": {
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=<region-ID>#logEvent:group=null;stream=null",
      "cloudWatchLogsArn": "arn:aws:logs:<region-ID>:<account-ID>:log-
group:null:log-stream:null",
      "cloudWatchLogs": {
        "status": "ENABLED"
      },
      "s3Logs": {
        "status": "DISABLED",
        "encryptionDisabled": false
      }
    },
    },
    "timeoutInMinutes": 60,
    "queuedTimeoutInMinutes": 480,
    "buildComplete": false,
    "initiator": "<username>",
    "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3"
  }
}

```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[でのバッチビルド AWS CodeBuild](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RetryBuild](#)」の「」を参照してください。

start-build-batch

次のコード例は、start-build-batch を使用する方法を示しています。

AWS CLI

でバッチビルドを開始するには AWS CodeBuild。

次のstart-build-batch例では、指定されたプロジェクトのバッチビルドを開始します。

```
aws codebuild start-build-batch \  
  --project-name <project-name>
```

出力:

```
{  
  "buildBatch": {  
    "id": "<project-name>:<batch-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-name>:<batch-ID>",  
    "startTime": "2020-10-21T16:54:24.740000+00:00",  
    "currentPhase": "SUBMITTED",  
    "buildBatchStatus": "IN_PROGRESS",  
    "projectName": "<project-name>",  
    "source": {  
      "type": "GITHUB",  
      "location": "<GitHub-repo-URL>",  
      "gitCloneDepth": 1,  
      "gitSubmodulesConfig": {  
        "fetchSubmodules": false  
      },  
      "reportBuildStatus": false,  
      "insecureSsl": false  
    },  
    "secondarySources": [],  
    "secondarySourceVersions": [],  
    "artifacts": {  
      "location": ""  
    },  
  },  
}
```

```
"secondaryArtifacts": [],
"cache": {
  "type": "NO_CACHE"
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
  "computeType": "BUILD_GENERAL1_SMALL",
  "environmentVariables": [],
  "privilegedMode": false,
  "imagePullCredentialsType": "CODEBUILD"
},
"logConfig": {
  "cloudWatchLogs": {
    "status": "ENABLED"
  },
  "s3Logs": {
    "status": "DISABLED",
    "encryptionDisabled": false
  }
},
"buildTimeoutInMinutes": 60,
"queuedTimeoutInMinutes": 480,
"complete": false,
"initiator": "<username>",
"encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
"buildBatchNumber": 3,
"buildBatchConfig": {
  "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<service-
role-name>",
  "restrictions": {
    "maximumBuildsAllowed": 100
  },
  "timeoutInMins": 480
}
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[のバッチビルド AWS CodeBuild](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartBuildBatch](#)」の「」を参照してください。

start-build

次の例は、start-build を使用する方法を説明しています。

AWS CLI

ビルドプロジェクトの AWS CodeBuild ビルドの実行を開始するには。

次のstart-build例では、指定された CodeBuild プロジェクトのビルドを開始します。ビルドは、タイムアウトする前にビルドをキューに入れることができる分数に対するプロジェクトの設定と、プロジェクトのアーティファクト設定の両方を上書きします。

```
aws codebuild start-build \  
  --project-name "my-demo-project" \  
  --queued-timeout-in-minutes-override 5 \  
  --artifacts-override {"\type\": \"S3\", \"location\": \"arn:aws:s3::artifacts-  
override\", \"overrideArtifactName\": true"}
```

出力:

```
{  
  "build": {  
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-  
service-role",  
    "buildStatus": "IN_PROGRESS",  
    "buildComplete": false,  
    "projectName": "my-demo-project",  
    "timeoutInMinutes": 60,  
    "source": {  
      "insecureSsl": false,  
      "type": "S3",  
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-  
source.zip"  
    },  
    "queuedTimeoutInMinutes": 5,  
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",  
    "currentPhase": "QUEUED",  
    "startTime": 1556905683.568,  
    "environment": {  
      "computeType": "BUILD_GENERAL1_MEDIUM",  
      "environmentVariables": [],  
      "type": "LINUX_CONTAINER",  
    }  
  }  
}
```

```
    "privilegedMode": false,
    "image": "aws/codebuild/standard:1.0",
    "imagePullCredentialsType": "CODEBUILD"
  },
  "phases": [
    {
      "phaseStatus": "SUCCEEDED",
      "startTime": 1556905683.568,
      "phaseType": "SUBMITTED",
      "durationInSeconds": 0,
      "endTime": 1556905684.524
    },
    {
      "startTime": 1556905684.524,
      "phaseType": "QUEUED"
    }
  ],
  "logs": {
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=null;stream=null"
  },
  "artifacts": {
    "encryptionDisabled": false,
    "location": "arn:aws:s3:::artifacts-override/my-demo-project",
    "overrideArtifactName": true
  },
  "cache": {
    "type": "NO_CACHE"
  },
  "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
  "initiator": "my-aws-account-name",
  "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
}
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「ビルドの実行 \(AWS CLI\)」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartBuild](#)」の「」を参照してください。

stop-build-batch

次の例は、stop-build-batch を使用する方法を説明しています。

AWS CLI

で進行中のバッチビルドを停止するには AWS CodeBuild。

次のstop-build-batch例では、指定されたバッチビルドを停止します。

```
aws codebuild stop-build-batch \  
  --id <project-name>:<batch-ID>
```

出力:

```
{  
  "buildBatch": {  
    "id": "<project-name>:<batch-ID>",  
    "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build-batch/<project-name>:<batch-ID>",  
    "startTime": "2020-10-21T16:54:24.740000+00:00",  
    "endTime": "2020-10-21T16:56:05.152000+00:00",  
    "currentPhase": "STOPPED",  
    "buildBatchStatus": "STOPPED",  
    "resolvedSourceVersion": "aef7744ed069c51098e15c360f4102cd2cd1ad64",  
    "projectName": "<project-name>",  
    "phases": [  
      {  
        "phaseType": "SUBMITTED",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T16:54:24.740000+00:00",  
        "endTime": "2020-10-21T16:54:25.039000+00:00",  
        "durationInSeconds": 0  
      },  
      {  
        "phaseType": "DOWNLOAD_BATCHSPEC",  
        "phaseStatus": "SUCCEEDED",  
        "startTime": "2020-10-21T16:54:25.039000+00:00",  
        "endTime": "2020-10-21T16:54:56.583000+00:00",  
        "durationInSeconds": 31  
      },  
      {  
        "phaseType": "IN_PROGRESS",  
        "phaseStatus": "STOPPED",
```

```
        "startTime": "2020-10-21T16:54:56.583000+00:00",
        "endTime": "2020-10-21T16:56:05.152000+00:00",
        "durationInSeconds": 68
    },
    {
        "phaseType": "STOPPED",
        "startTime": "2020-10-21T16:56:05.152000+00:00"
    }
],
"source": {
    "type": "GITHUB",
    "location": "<GitHub-repo-URL>",
    "gitCloneDepth": 1,
    "gitSubmodulesConfig": {
        "fetchSubmodules": false
    },
    "reportBuildStatus": false,
    "insecureSsl": false
},
"secondarySources": [],
"secondarySourceVersions": [],
"artifacts": {
    "location": ""
},
"secondaryArtifacts": [],
"cache": {
    "type": "NO_CACHE"
},
"environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:3.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [],
    "privilegedMode": false,
    "imagePullCredentialsType": "CODEBUILD"
},
"logConfig": {
    "cloudWatchLogs": {
        "status": "ENABLED"
    },
    "s3Logs": {
        "status": "DISABLED",
        "encryptionDisabled": false
    }
}
```

```

    },
    "buildTimeoutInMinutes": 60,
    "queuedTimeoutInMinutes": 480,
    "complete": true,
    "initiator": "Strohm",
    "encryptionKey": "arn:aws:kms:<region-ID>:<account-ID>:alias/aws/s3",
    "buildBatchNumber": 3,
    "buildBatchConfig": {
      "serviceRole": "arn:aws:iam::<account-ID>:role/service-role/<project-
name>",
      "restrictions": {
        "maximumBuildsAllowed": 100
      },
      "timeoutInMins": 480
    },
  },
  "buildGroups": [
    {
      "identifier": "DOWNLOAD_SOURCE",
      "ignoreFailure": false,
      "currentBuildSummary": {
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
        "requestedOn": "2020-10-21T16:54:25.468000+00:00",
        "buildStatus": "SUCCEEDED",
        "primaryArtifact": {
          "type": "no_artifacts",
          "identifier": "DOWNLOAD_SOURCE"
        },
        "secondaryArtifacts": []
      }
    },
    {
      "identifier": "linux_small",
      "dependsOn": [],
      "ignoreFailure": false,
      "currentBuildSummary": {
        "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
        "requestedOn": "2020-10-21T16:54:56.833000+00:00",
        "buildStatus": "IN_PROGRESS"
      }
    },
    {
      "identifier": "linux_medium",

```

```

    "dependsOn": [
      "linux_small"
    ],
    "ignoreFailure": false,
    "currentBuildSummary": {
      "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
      "requestedOn": "2020-10-21T16:54:56.211000+00:00",
      "buildStatus": "PENDING"
    }
  },
  {
    "identifier": "linux_large",
    "dependsOn": [
      "linux_medium"
    ],
    "ignoreFailure": false,
    "currentBuildSummary": {
      "arn": "arn:aws:codebuild:<region-ID>:<account-ID>:build/
<project-name>:<build-ID>",
      "requestedOn": "2020-10-21T16:54:56.330000+00:00",
      "buildStatus": "PENDING"
    }
  }
]
}
}

```

詳細については、AWS CodeBuild [「ユーザーガイド」の「でのバッチビルド AWS CodeBuild」](#)を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンス StopBuildBatch」](#)の「」を参照してください。

stop-build

次のコード例は、stop-build を使用する方法を示しています。

AWS CLI

ビルドプロジェクトの AWS CodeBuild ビルドを停止するには。

次のstop-build例では、指定された CodeBuild ビルドを停止します。

```
aws codebuild stop-build --id my-demo-project:12345678-a1b2-c3d4-e5f6-11111EXAMPLE
```

出力:

```
{
  "build": {
    "startTime": 1556906956.318,
    "initiator": "my-aws-account-name",
    "projectName": "my-demo-project",
    "currentPhase": "COMPLETED",
    "cache": {
      "type": "NO_CACHE"
    },
    "source": {
      "insecureSsl": false,
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-source.zip",
      "type": "S3"
    },
    "id": "my-demo-project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
    "endTime": 1556906974.781,
    "phases": [
      {
        "durationInSeconds": 0,
        "phaseType": "SUBMITTED",
        "endTime": 1556906956.935,
        "phaseStatus": "SUCCEEDED",
        "startTime": 1556906956.318
      },
      {
        "durationInSeconds": 1,
        "phaseType": "QUEUED",
        "endTime": 1556906958.272,
        "phaseStatus": "SUCCEEDED",
        "startTime": 1556906956.935
      },
      {
        "phaseType": "PROVISIONING",
        "phaseStatus": "SUCCEEDED",
        "durationInSeconds": 14,
        "contexts": [
          {
            "message": ""
          }
        ]
      }
    ]
  }
}
```

```
        "statusCode": ""
      }
    ],
    "endTime": 1556906972.847,
    "startTime": 1556906958.272
  },
  {
    "phaseType": "DOWNLOAD_SOURCE",
    "phaseStatus": "SUCCEEDED",
    "durationInSeconds": 0,
    "contexts": [
      {
        "message": "",
        "statusCode": ""
      }
    ],
    "endTime": 1556906973.552,
    "startTime": 1556906972.847
  },
  {
    "phaseType": "INSTALL",
    "phaseStatus": "SUCCEEDED",
    "durationInSeconds": 0,
    "contexts": [
      {
        "message": "",
        "statusCode": ""
      }
    ],
    "endTime": 1556906973.75,
    "startTime": 1556906973.552
  },
  {
    "phaseType": "PRE_BUILD",
    "phaseStatus": "SUCCEEDED",
    "durationInSeconds": 0,
    "contexts": [
      {
        "message": "",
        "statusCode": ""
      }
    ],
    "endTime": 1556906973.937,
    "startTime": 1556906973.75
  }
}
```



```
    },
    {
      "durationInSeconds": 0,
      "phaseType": "BUILD",
      "endTime": 1556906974.781,
      "phaseStatus": "STOPPED",
      "startTime": 1556906973.937
    },
    {
      "phaseType": "COMPLETED",
      "startTime": 1556906974.781
    }
  ],
  "artifacts": {
    "location": "arn:aws:s3:::artifacts-override/my-demo-project",
    "encryptionDisabled": false,
    "overrideArtifactName": true
  },
  "buildComplete": true,
  "buildStatus": "STOPPED",
  "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
  "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
  "queuedTimeoutInMinutes": 5,
  "timeoutInMinutes": 60,
  "environment": {
    "type": "LINUX_CONTAINER",
    "environmentVariables": [],
    "computeType": "BUILD_GENERAL1_MEDIUM",
    "privilegedMode": false,
    "image": "aws/codebuild/standard:1.0",
    "imagePullCredentialsType": "CODEBUILD"
  },
  "logs": {
    "streamName": "1a2b3c4d-5678-90ab-cdef-11111EXAMPLE",
    "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-demo-project;stream=1a2b3c4d-5678-90ab-
cdef-11111EXAMPLE",
    "groupName": "/aws/codebuild/my-demo-project"
  },
  "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project:1a2b3c4d-5678-90ab-cdef-11111EXAMPLE"
}
```

```
}

```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[ビルドを停止する \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopBuild](#)」の「」を参照してください。

update-project

次のコード例は、update-project を使用する方法を示しています。

AWS CLI

AWS CodeBuild ビルドプロジェクトの設定を変更するには。

次のupdate-project例では、 という名前の指定された CodeBuild ビルドプロジェクトの設定を変更します my-demo-project。

```
aws codebuild update-project --name "my-demo-project" \
  --description "This project is updated" \
  --source "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-
input-bucket/my-source-2.zip\"}" \
  --artifacts "{\"type\": \"S3\", \"location\": \"codebuild-us-west-2-123456789012-
output-bucket-2\"}" \
  --environment "{\"type\": \"LINUX_CONTAINER\", \"image\": \"aws/codebuild/
standard:1.0\", \"computeType\": \"BUILD_GENERAL1_MEDIUM\"}" \
  --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role"
```

出力には、更新された設定が表示されます。

```
{
  "project": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-project",
    "environment": {
      "privilegedMode": false,
      "environmentVariables": [],
      "type": "LINUX_CONTAINER",
      "image": "aws/codebuild/standard:1.0",
      "computeType": "BUILD_GENERAL1_MEDIUM",
      "imagePullCredentialsType": "CODEBUILD"
```

```
    },
    "queuedTimeoutInMinutes": 480,
    "description": "This project is updated",
    "artifacts": {
      "packaging": "NONE",
      "name": "my-demo-project",
      "type": "S3",
      "namespaceType": "NONE",
      "encryptionDisabled": false,
      "location": "codebuild-us-west-2-123456789012-output-bucket-2"
    },
    "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
    "badge": {
      "badgeEnabled": false
    },
    "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
    "lastModified": 1556840545.967,
    "tags": [],
    "timeoutInMinutes": 60,
    "created": 1556839783.274,
    "name": "my-demo-project",
    "cache": {
      "type": "NO_CACHE"
    },
    "source": {
      "type": "S3",
      "insecureSsl": false,
      "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source-2.zip"
    }
  }
}
```

詳細については、AWS CodeBuild 「[ユーザーガイド](#)」の「[ビルドプロジェクトの設定の変更 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateProject](#)」の「」を参照してください。

update-report-group

次の例は、update-report-group を使用する方法を説明しています。

AWS CLI

でレポートグループを更新するには AWS CodeBuild。

次のupdate-report-group例では、レポートグループのエクスポートタイプを「NO_EXPORT」に変更します。

```
aws codebuild update-report-group \  
  --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-  
group \  
  --export-config="exportConfigType=NO_EXPORT"
```

出力:

```
{  
  "reportGroup": {  
    "arn": "arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-  
report-group",  
    "name": "cli-created-report-group",  
    "type": "TEST",  
    "exportConfig": {  
      "exportConfigType": "NO_EXPORT"  
    },  
    "created": 1602020686.009,  
    "lastModified": 1602021033.454,  
    "tags": []  
  }  
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の [「レポートグループの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateReportGroup](#)」の「」を参照してください。

update-webhook

次の例は、update-webhook を使用する方法を説明しています。

AWS CLI

AWS CodeBuild プロジェクトのウェブフックを更新するには

次のupdate-webhook例では、指定された CodeBuild プロジェクトのウェブフックを2つのフィルターグループで更新します。--rotate-secret パラメータは、コード変更がビルドをトリガーするたびに、プロジェクトのシークレットキーを GitHub ローテーションするように指定します。最初のフィルタグループでは、正規表現 ^refs/heads/master\$ と一致する Git 参照名および ^refs/heads/myBranch\$ と一致するヘッド参照を持つブランチに対してプルリクエストを作成、更新、または再開することを指定します。2番目のフィルターグループは、正規表現 と一致しない Git 参照名を持つブランチへのプッシュリクエストを指定します^refs/heads/myBranch\$。

```
aws codebuild update-webhook \
  --project-name Project2 \
  --rotate-secret \
  --filter-groups "[[{"type":"EVENT","pattern":"PULL_REQUEST_CREATED,
PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED"}, {"type":"HEAD_REF","pattern
":"^refs/heads/myBranch$"},"excludeMatchedPattern":true}, {"type":"BASE_REF
","pattern":"^refs/heads/master$"},"excludeMatchedPattern":true}], [{"type":
"EVENT","pattern":"PUSH"}, {"type":"HEAD_REF","pattern":"^refs/heads/
myBranch$"},"excludeMatchedPattern":true}]]"
```

出力:

```
{
  "webhook": {
    "filterGroups": [
      [
        {
          "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED",
          "type": "EVENT"
        },
        {
          "excludeMatchedPattern": true,
          "pattern": "refs/heads/myBranch$",
          "type": "HEAD_REF"
        },
        {
          "excludeMatchedPattern": true,
          "pattern": "refs/heads/master$",
          "type": "BASE_REF"
        }
      ],
      [
```

```
    {
      "pattern": "PUSH",
      "type": "EVENT"
    },
    {
      "excludeMatchedPattern": true,
      "pattern": "refs/heads/myBranch$",
      "type": "HEAD_REF"
    }
  ],
  "lastModifiedSecret": 1556312220.133
}
```

詳細については、AWS CodeBuild 「ユーザーガイド」の「[ビルドプロジェクトの設定の変更 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateWebhook](#)」の「」を参照してください。

CodeCommit を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CodeCommit。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-approval-rule-template-with-repository

次のコード例は、`associate-approval-rule-template-with-repository` を使用する方法を示しています。

AWS CLI

承認ルールテンプレートをリポジトリに関連付けるには

次の`associate-approval-rule-template-with-repository`例では、指定された承認ルールテンプレートを という名前のリポジトリに関連付けますMyDemoRepo。

```
aws codecommit associate-approval-rule-template-with-repository \  
  --repository-name MyDemoRepo \  
  --approval-rule-template-name 2-approver-rule-for-main
```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit 「ユーザーガイド」の「[承認ルールテンプレートをリポジトリに関連付ける](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateApprovalRuleTemplateWithRepository](#)」の「」を参照してください。

batch-associate-approval-rule-template-with-repositories

次のコード例は、`batch-associate-approval-rule-template-with-repositories` を使用する方法を示しています。

AWS CLI

承認ルールテンプレートを 1 つのオペレーションで複数のリポジトリに関連付けるには

次の`batch-associate-approval-rule-template-with-repositories`例では、指定された承認ルールテンプレートを MyDemoRepo および という名前のリポジトリに関連付けますMyOtherDemoRepo。

注: 承認ルールテンプレートは、作成された AWS リージョンに固有のものです。これらは、その AWS リージョンのリポジトリにのみ関連付けることができます。

```
aws codecommit batch-associate-approval-rule-template-with-repositories \  
  --repository-names MyDemoRepo, MyOtherDemoRepo \  
  --approval-rule-template-name 2-approver-rule-for-main
```

出力:

```
{  
  "associatedRepositoryNames": [  
    "MyDemoRepo",  
    "MyOtherDemoRepo"  
  ],  
  "errors": []  
}
```

詳細については、AWS CodeCommit「ユーザーガイド」の「[承認ルールテンプレートをリポジトリに関連付ける](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[BatchAssociateApprovalRuleTemplateWithRepositories](#)」の「」を参照してください。

batch-describe-merge-conflicts

次の例は、batch-describe-merge-conflicts を使用する方法を説明しています。

AWS CLI

2つのコミット指定子間のマージ内のすべてのファイルまたはファイルのサブセットのマージ競合に関する情報を取得するには

次のbatch-describe-merge-conflicts例では、という名前のリポジトリのTHREE_WAY_MERGE戦略mainを使用して、という名前のソースブランチfeature-randomizationfeatureをという名前の宛先ブランチとマージする際のマージ競合を決定しますMyDemoRepo。

```
aws codecommit batch-describe-merge-conflicts \  
  --source-commit-specifier feature-randomizationfeature \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE \  
  --repository-name MyDemoRepo
```

出力:


```
{
  "conflicts": [
    {
      "conflictMetadata": {
        "filePath": "readme.md",
        "fileSizes": {
          "source": 139,
          "destination": 230,
          "base": 85
        },
        "fileModes": {
          "source": "NORMAL",
          "destination": "NORMAL",
          "base": "NORMAL"
        },
        "objectTypes": {
          "source": "FILE",
          "destination": "FILE",
          "base": "FILE"
        },
        "numberOfConflicts": 1,
        "isBinaryFile": {
          "source": false,
          "destination": false,
          "base": false
        },
        "contentConflict": true,
        "fileModeConflict": false,
        "objectTypeConflict": false,
        "mergeOperations": {
          "source": "M",
          "destination": "M"
        }
      },
      "mergeHunks": [
        {
          "isConflict": true,
          "source": {
            "startLine": 0,
            "endLine": 3,
            "hunkContent": "VGhpcyBpEXAMPLE=="
          },
          "destination": {
```

```

        "startLine": 0,
        "endLine": 1,
        "hunkContent": "VXNlIHRoEXAMPLE="
      }
    }
  ]
}
],
"errors": [],
"destinationCommitId": "86958e0aEXAMPLE",
"sourceCommitId": "6ccd57fdEXAMPLE",
"baseCommitId": "767b6958EXAMPLE"
}

```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[プルリクエストの競合の解決](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDescribeMergeConflicts](#)」の「」を参照してください。

batch-disassociate-approval-rule-template-from-repositories

次のコード例は、batch-disassociate-approval-rule-template-from-repositories を使用する方法を示しています。

AWS CLI

1 回のオペレーションで複数のリポジトリから承認ルールテンプレートの関連付けを解除するには

次のbatch-disassociate-approval-rule-template-from-repositories例では、指定された承認ルールテンプレートを MyDemoRepo および という名前のリポジトリから関連付け解除しますMyOtherDemoRepo。

```

aws codecommit batch-disassociate-approval-rule-template-from-repositories \
  --repository-names MyDemoRepo, MyOtherDemoRepo \
  --approval-rule-template-name 1-approval-rule-for-all pull requests

```

出力:

```

{
  "disassociatedRepositoryNames": [

```

```
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[承認ルールテンプレートの関連付けを解除する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchDisassociateApprovalRuleTemplateFromRepositories](#)」の「」を参照してください。

batch-get-commits

次のコード例は、batch-get-commits を使用する方法を示しています。

AWS CLI

複数のコミットに関する情報を表示するには

次のbatch-get-commits例では、指定されたコミットの詳細を表示します。

```
aws codecommit batch-get-commits \
  --repository-name MyDemoRepo \
  --commit-ids 317f8570EXAMPLE 4c925148EXAMPLE
```

出力:

```
{
  "commits": [
    {
      "additionalData": "",
      "committer": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
      "author": {
        "date": "1508280564 -0800",
        "name": "Mary Major",
        "email": "mary_major@example.com"
      },
    },
  ],
}
```

```
    "commitId": "317f8570EXAMPLE",
    "treeId": "1f330709EXAMPLE",
    "parents": [
      "6e147360EXAMPLE"
    ],
    "message": "Change variable name and add new response element"
  },
  {
    "additionalData": "",
    "committer": {
      "date": "1508280542 -0800",
      "name": "Li Juan",
      "email": "li_juan@example.com"
    },
    "author": {
      "date": "1508280542 -0800",
      "name": "Li Juan",
      "email": "li_juan@example.com"
    },
    "commitId": "4c925148EXAMPLE",
    "treeId": "1f330709EXAMPLE",
    "parents": [
      "317f8570EXAMPLE"
    ],
    "message": "Added new class"
  }
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「コミットの詳細を表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchGetCommits](#)」の「」を参照してください。

batch-get-repositories

次の例は、batch-get-repositories を使用する方法を説明しています。

AWS CLI

複数のリポジトリの詳細を表示するには

この例では、複数の AWS CodeCommit リポジトリの詳細を示しています。

```
aws codecommit batch-get-repositories \  
--repository-names MyDemoRepo MyOtherDemoRepo
```

出力:

```
{  
  "repositoriesNotFound": [],  
  "repositories": [  
    {  
      "creationDate": 1429203623.625,  
      "defaultBranch": "main",  
      "repositoryName": "MyDemoRepo",  
      "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/  
MyDemoRepo",  
      "lastModifiedDate": 1430783812.0869999,  
      "repositoryDescription": "My demonstration repository",  
      "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/  
MyDemoRepo",  
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",  
      "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo"  
      "accountId": "111111111111"  
    },  
    {  
      "creationDate": 1429203623.627,  
      "defaultBranch": "main",  
      "repositoryName": "MyOtherDemoRepo",  
      "cloneUrlSsh": "ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/  
MyOtherDemoRepo",  
      "lastModifiedDate": 1430783812.0889999,  
      "repositoryDescription": "My other demonstration repository",  
      "cloneUrlHttp": "https://codecommit.us-east-2.amazonaws.com/v1/repos/  
MyOtherDemoRepo",  
      "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE",  
      "Arn": "arn:aws:codecommit:us-east-2:111111111111:MyOtherDemoRepo"  
      "accountId": "111111111111"  
    }  
  ],  
  "repositoriesNotFound": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchGetRepositories](#)」の「」を参照してください。

create-approval-rule-template

次のコード例は、create-approval-rule-template を使用する方法を示しています。

AWS CLI

承認ルールテンプレートを作成するには

次のcreate-approval-rule-template例では、という名前の承認ルールテンプレートを作成して2-approver-rule-for-main`。 The template requires two users who assume the role of `CodeCommitReview、プルリクエストをmainブランチにマージする前に承認します。

```
aws codecommit create-approval-rule-template \  
  --approval-rule-template-name 2-approver-rule-for-main \  
  --approval-rule-template-description "Requires two developers from the team to  
approve the pull request if the destination branch is main" \  
  --approval-rule-template-content '{"Version": "2018-11-08",  
"DestinationReferences": [{"refs/heads/main"}], "Statements": [{"Type  
": "Approvers", "NumberOfApprovalsNeeded": 2, "ApprovalPoolMembers":  
["arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*"]}]}'
```

出力:

```
{  
  "approvalRuleTemplate": {  
    "approvalRuleTemplateName": "2-approver-rule-for-main",  
    "creationDate": 1571356106.936,  
    "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",  
    "approvalRuleTemplateContent": '{"Version": "2018-11-08",  
"DestinationReferences": [{"refs/heads/main"}], "Statements": [{"Type  
": "Approvers", "NumberOfApprovalsNeeded": 2, "ApprovalPoolMembers":  
["arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*"]}]}'",  
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",  
    "approvalRuleTemplateDescription": "Requires two developers from the team to  
approve the pull request if the destination branch is main",  
    "lastModifiedDate": 1571356106.936,  
    "ruleContentSha256": "4711b576EXAMPLE"  
  }  
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[承認ルールテンプレートの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateApprovalRuleTemplate](#)」の「」を参照してください。

create-branch

次のコード例は、create-branch を使用する方法を示しています。

AWS CLI

ブランチを作成するには

この例では、AWS CodeCommit リポジトリにブランチを作成します。このコマンドは、エラーがある場合にのみ出力を生成します。

コマンド:

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
--commit-id 317f8570EXAMPLE
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBranch](#)」の「」を参照してください。

create-commit

次の例は、create-commit を使用する方法を説明しています。

AWS CLI

コミットを作成するには

次のcreate-commit例は、mainブランチMyDemoRepoで という名前のリポジトリにreadme.mdファイルを追加するリポジトリの初期コミットを作成する方法を示しています。

```
aws codecommit create-commit \
```

```
--repository-name MyDemoRepo \  
--branch-name main \  
--put-files "filePath=readme.md,fileContent='Welcome to our team repository.'"
```

出力:

```
{  
  "filesAdded": [  
    {  
      "blobId": "5e1c309d-EXAMPLE",  
      "absolutePath": "readme.md",  
      "fileMode": "NORMAL"  
    }  
  ],  
  "commitId": "4df8b524-EXAMPLE",  
  "treeId": "55b57003-EXAMPLE",  
  "filesDeleted": [],  
  "filesUpdated": []  
}
```

詳細については、AWS CodeCommit 「[ユーザーガイド](#)」の「[コミットの作成 AWS CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス CreateCommit](#)」の「」を参照してください。

create-pull-request-approval-rule

次の例は、create-pull-request-approval-rule を使用する方法を説明しています。

AWS CLI

プルリクエストの承認ルールを作成するには

次のcreate-pull-request-approval-rule例では、指定されたプルリクエストRequire two approved approversに という名前の承認ルールを作成します。このルールでは、承認プールから 2 つの承認が必要であることを指定します。プールには、123456789012 AWS アカウントCodeCommitReviewでのロールを引き受ける CodeCommit こと で にアクセスするすべてのユーザーが含まれます。また、同じ AWS アカウントNikhil_Jayashankarから という名前のIAMユーザーまたはフェデレーテッドユーザーも含まれます。

```
aws codecommit create-pull-request-approval-rule \  

```



```
--approval-rule-name "Require two approved approvers" \
--approval-rule-content "{\"Version\": \"2018-11-08\",\"Statements\":
[{\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}"
```

出力:

```
{
  "approvalRule": {
    "approvalRuleName": "Require two approved approvers",
    "lastModifiedDate": 1570752871.932,
    "ruleContentSha256": "7c44e6ebEXAMPLE",
    "creationDate": 1570752871.932,
    "approvalRuleId": "aac33506-EXAMPLE",
    "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
[{\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major"
  }
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[承認ルールの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePullRequestApprovalRule](#)」の「」を参照してください。

create-pull-request

次の例は、create-pull-request を使用する方法を説明しています。

AWS CLI

プルリクエストを作成するには

次のcreate-pull-request例では、「Pronunciation difficulty analyzer」という名前のプルリクエストを作成し、「火曜日までにこれらの変更を確認してください」と説明します。この変更は「jane-branch」ソースブランチを対象とし、「」という名前の AWS CodeCommit リポジトリのデフォルトのブランチ「main」にマージされますMyDemoRepo。

```
aws codecommit create-pull-request \
  --title "My Pull Request" \
  --description "Please review these changes by Tuesday" \
  --client-request-token 123Example \
  --targets repositoryName=MyDemoRepo,sourceReference=MyNewBranch
```

出力:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
        \"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type
        \": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
        [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
          "approvalRuleTemplateId": "dd3d22fe-EXAMPLE",
          "approvalRuleTemplateName": "2-approver-rule-for-main"
        },
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "authorArn": "arn:aws:iam::111111111111:user/Jane_Doe",
    "description": "Please review these changes by Tuesday",
    "title": "Pronunciation difficulty analyzer",
    "pullRequestTargets": [
      {
        "destinationCommit": "5d036259EXAMPLE",
        "destinationReference": "refs/heads/main",
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "317f8570EXAMPLE",
        "sourceReference": "refs/heads/jane-branch",
        "mergeMetadata": {
          "isMerged": false
        }
      }
    ]
  }
},
```

```
    "lastActivityDate": 1508962823.285,  
    "pullRequestId": "42",  
    "clientRequestToken": "123Example",  
    "pullRequestStatus": "OPEN",  
    "creationDate": 1508962823.285  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePullRequest](#)」の「」を参照してください。

create-repository

次のコード例は、create-repository を使用する方法を示しています。

AWS CLI

リポジトリを作成するには

この例では、リポジトリを作成し、ユーザーの AWS アカウントと関連付けます。

コマンド:

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-  
description "My demonstration repository"
```

出力:

```
{  
  "repositoryMetadata": {  
    "repositoryName": "MyDemoRepo",  
    "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/  
repos/MyDemoRepo",  
    "lastModifiedDate": 1444766838.027,  
    "repositoryDescription": "My demonstration repository",  
    "cloneUrlHttp": "https://git-codecommit.us-east-1.amazonaws.com/v1/  
repos/MyDemoRepo",  
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",  
    "Arn": "arn:aws:codecommit:us-  
east-1:111111111111EXAMPLE:MyDemoRepo",  
    "accountId": "111111111111"  
  }  
}
```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRepository](#)」の「」を参照してください。

create-unreferenced-merge-commit

次の例は、create-unreferenced-merge-commit を使用する方法を説明しています。

AWS CLI

2つのコミット指定子をマージした結果を表す参照されていないコミットを作成するには

次のcreate-unreferenced-merge-commit例では、という名前のリポジトリでTHREE_WAY_MERGE 戦略mainを使用して、という名前のソースブランチbugfix-1234とという名前の宛先ブランチ間のマージの結果を表すコミットを作成しますMyDemoRepo。

```
aws codecommit create-unreferenced-merge-commit \  
  --source-commit-specifier bugfix-1234 \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE \  
  --repository-name MyDemoRepo \  
  --name "Maria Garcia" \  
  --email "maria_garcia@example.com" \  
  --commit-message "Testing the results of this merge."
```

出力:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[プルリクエストの競合の解決](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUnreferencedMergeCommit](#)」の「」を参照してください。

credential-helper

次の例は、credential-helper を使用する方法を説明しています。

AWS CLI

で含まれる AWS CLI 認証情報ヘルパーを設定するには AWS CodeCommit

credential-helper ユーティリティは、 から直接呼び出されるように設計されていません AWS CLI。代わりに、ローカルコンピュータをセットアップするための git config コマンドでパラメータとして使用することを目的としています。これにより、Git が CodeCommit リポジトリとやり取り AWS するために を認証する必要があるときはいつでも、Git が HTTPS および暗号化で署名されたバージョンの IAM ユーザー認証情報または Amazon EC2 インスタンスロールを使用できるようになります。

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

出力:

```
[credential]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「他の方法 AWS CodeCommit を使用するためのセットアップ」を参照してください。内容を注意深く確認し、AWS CodeCommit ユーザーガイドの Linux、macOS または Unix で HTTPS の接続、または Windows で HTTPS の接続のいずれかのトピックの手順に従います。

- API 詳細については、AWS CLI 「コマンドリファレンス [CredentialHelper](#)」の「」を参照してください。

delete-approval-rule-template

次の例は、delete-approval-rule-template を使用する方法を説明しています。

AWS CLI

承認ルールテンプレートを削除するには

次のdelete-approval-rule-template例では、指定された承認ルールテンプレートを削除します。

```
aws codecommit delete-approval-rule-template \  
  --approval-rule-template-name 1-approver-for-all-pull-requests
```

出力:

```
{  
  "approvalRuleTemplateId": "41de97b7-EXAMPLE"  
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[承認ルールテンプレートの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApprovalRuleTemplate](#)」の「」を参照してください。

delete-branch

次の例は、delete-branch を使用する方法を説明しています。

AWS CLI

ブランチを削除するには

この例では、AWS CodeCommit リポジトリ内のブランチを削除する方法を示しています。

コマンド:

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

出力:

```
{  
  "branch": {  
    "commitId": "317f8570EXAMPLE",  
    "branchName": "MyNewBranch"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBranch](#)」の「」を参照してください。

delete-comment-content

次のコード例は、delete-comment-content を使用する方法を示しています。

AWS CLI

コメントの内容を削除するには

コメントを自分で作成した場合のみコメントのコンテンツを削除できます。この例では、システム生成 ID が のコメントの内容を削除する方法を示します `ff30b348EXAMPLEb9aa670f`。

```
aws codecommit delete-comment-content \  
  --comment-id ff30b348EXAMPLEb9aa670f
```

出力:

```
{  
  "comment": {  
    "creationDate": 1508369768.142,  
    "deleted": true,  
    "lastModifiedDate": 1508369842.278,  
    "clientRequestToken": "123Example",  
    "commentId": "ff30b348EXAMPLEb9aa670f",  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "callerReactions": [],  
    "reactionCounts":  
    {  
      "CLAP" : 1  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCommentContent](#)」の「」を参照してください。

delete-file

次のコード例は、delete-file を使用する方法を示しています。

AWS CLI

ファイルを削除するには

次のdelete-file例は、 という名前のリポジトリc5709475EXAMPLEで、最新のコミットID mainが のブランチREADME.mdから という名前のファイルを削除する方法を示していますMyDemoRepo。

```
aws codecommit delete-file \  
  --repository-name MyDemoRepo \  
  --branch-name main \  
  --file-path README.md \  
  --parent-commit-id c5709475EXAMPLE
```

出力:

```
{  
  "blobId": "559b44fEXAMPLE",  
  "commitId": "353cf655EXAMPLE",  
  "filePath": "README.md",  
  "treeId": "6bc824cEXAMPLE"  
}
```

詳細については、AWS CodeCommit APIリファレンスガイドの「[ファイルの編集または削除 AWS CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFile](#)」の「」を参照してください。

delete-pull-request-approval-rule

次のコード例は、delete-pull-request-approval-rule を使用する方法を示しています。

AWS CLI

プルリクエストの承認ルールを削除するには

次のdelete-pull-request-approval-rule例では、指定されたプルリクエストMy Approval Ruleに という名前の承認ルールを削除します。

```
aws codecommit delete-pull-request-approval-rule \  
  --repository-name MyDemoRepo \  
  --pull-request-id MyApprovalRule
```



```
--approval-rule-name "My Approval Rule" \  
--pull-request-id 15
```

出力:

```
{  
  "approvalRuleId": "077d8e8a8-EXAMPLE"  
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「承認ルールの編集または削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeletePullRequestApprovalRule](#)」の「」を参照してください。

delete-repository

次のコード例は、delete-repository を使用する方法を示しています。

AWS CLI

リポジトリを削除するには

この例では、AWS CodeCommit リポジトリを削除する方法を示しています。

コマンド:

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

出力:

```
{  
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteRepository](#)」の「」を参照してください。

describe-merge-conflicts

次の例は、describe-merge-conflicts を使用する方法を説明しています。

AWS CLI

マージ競合に関する詳細情報を取得するには

次のdescribe-merge-conflicts例では、THREE_WAY_MERGE 戦略を使用して、指定されたソースブランチと宛先ブランチreadme.mdで という名前のファイルのマージ競合を決定します。

```
aws codecommit describe-merge-conflicts \  
  --source-commit-specifier feature-randomizationfeature \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE \  
  --file-path readme.md \  
  --repository-name MyDemoRepo
```

出力:

```
{  
  "conflictMetadata": {  
    "filePath": "readme.md",  
    "fileSizes": {  
      "source": 139,  
      "destination": 230,  
      "base": 85  
    },  
    "fileModes": {  
      "source": "NORMAL",  
      "destination": "NORMAL",  
      "base": "NORMAL"  
    },  
    "objectTypes": {  
      "source": "FILE",  
      "destination": "FILE",  
      "base": "FILE"  
    },  
    "numberOfConflicts": 1,  
    "isBinaryFile": {  
      "source": false,  
      "destination": false,  
      "base": false  
    },  
    "contentConflict": true,  
    "fileModeConflict": false,  
  },  
}
```

```
    "objectTypeConflict": false,
    "mergeOperations": {
      "source": "M",
      "destination": "M"
    }
  },
  "mergeHunks": [
    {
      "isConflict": true,
      "source": {
        "startLine": 0,
        "endLine": 3,
        "hunkContent": "VGhpcyBpEXAMPLE="
      },
      "destination": {
        "startLine": 0,
        "endLine": 1,
        "hunkContent": "VXNlIHRoEXAMPLE="
      }
    }
  ],
  "destinationCommitId": "86958e0aEXAMPLE",
  "sourceCommitId": "6ccd57fdEXAMPLE",
  "baseCommitId": "767b69580EXAMPLE"
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[プルリクエストの競合の解決](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeMergeConflicts](#)」の「」を参照してください。

describe-pull-request-events

次のコード例は、describe-pull-request-events を使用する方法を示しています。

AWS CLI

プルリクエストでイベントを表示するには

次のdescribe-pull-request-events例では、ID が 8 のプルリクエストのイベントを取得します。

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

出力:

```
{
  "pullRequestEvents": [
    {
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_CREATED",
      "eventDate": 1510341779.53,
      "actor": "arn:aws:iam::111111111111:user/Zhang_Wei"
    },
    {
      "pullRequestStatusChangedEventMetadata": {
        "pullRequestStatus": "CLOSED"
      },
      "pullRequestId": "8",
      "pullRequestEventType": "PULL_REQUEST_STATUS_CHANGED",
      "eventDate": 1510341930.72,
      "actor": "arn:aws:iam::111111111111:user/Jane_Doe"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePullRequestEvents](#)」の「」を参照してください。

disassociate-approval-rule-template-from-repository

次の例は、`disassociate-approval-rule-template-from-repository` を使用する方法を説明しています。

AWS CLI

リポジトリから承認ルールテンプレートの関連付けを解除するには

次の`disassociate-approval-rule-template-from-repository`例では、指定された承認ルールテンプレートを という名前のリポジトリから関連付け解除しますMyDemoRepo。

```
aws codecommit disassociate-approval-rule-template-from-repository \
  --repository-name MyDemoRepo \
```

```
--approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit ユーザーガイドの [「承認ルールテンプレートの関連付けを解除する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateApprovalRuleTemplateFromRepository](#)」の「」を参照してください。

evaluate-pull-request-approval-rules

次の例は、evaluate-pull-request-approval-rules を使用する方法を説明しています。

AWS CLI

プルリクエストの承認ルールがすべて満たされているかどうかを評価するには

次のevaluate-pull-request-approval-rules例では、指定されたプルリクエストの承認ルールの状態を評価します。この例では、プルリクエストの承認ルールが満たされていないため、コマンドの出力には approved の値が表示されず false。

```
aws codecommit evaluate-pull-request-approval-rules \  
  --pull-request-id 27 \  
  --revision-id 9f29d167EXAMPLE
```

出力:

```
{  
  "evaluation": {  
    "approved": false,  
    "approvalRulesNotSatisfied": [  
      "Require two approved approvers"  
    ],  
    "overridden": false,  
    "approvalRulesSatisfied": []  
  }  
}
```

詳細については、AWS CodeCommit ユーザーガイドの [「プルリクエストのマージ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EvaluatePullRequestApprovalRules](#)」の「」を参照してください。

get-approval-rule-template

次の例は、get-approval-rule-template を使用する方法を説明しています。

AWS CLI

承認ルールテンプレートの内容を取得するには

次のget-approval-rule-template例では、 という名前の承認ルールテンプレートの内容を取得します1-approver-rule-for-all-pull-requests。

```
aws codecommit get-approval-rule-template \  
--approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

出力:

```
{  
  "approvalRuleTemplate": {  
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\":  
  [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\":  
  [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",  
    "ruleContentSha256": "621181bbEXAMPLE",  
    "lastModifiedDate": 1571356106.936,  
    "creationDate": 1571356106.936,  
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",  
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan",  
    "approvalRuleTemplateId": "a29abb15-EXAMPLE",  
    "approvalRuleTemplateDescription": "All pull requests must be approved by  
one developer on the team."  
  }  
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[承認ルールテンプレートの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApprovalRuleTemplate](#)」の「」を参照してください。

get-blob

次の例は、get-blob を使用する方法を説明しています。

AWS CLI

Git BLOB オブジェクトに関する情報を表示するには

次のget-blob例では、ID が '2eb4af3b EXAMPLE' の Git BLOB に関する情報を、'MyDemoRepo' という名前の AWS CodeCommit リポジトリで取得します。

```
aws codecommit get-blob --repository-name MyDemoRepo --blob-id 2eb4af3bEXAMPLE
```

出力:

```
{
  "content": "QSBcaw5hcnkgTGFyToEXAMPLE="
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBlob](#)」の「」を参照してください。

get-branch

次のコード例は、get-branch を使用する方法を示しています。

AWS CLI

ブランチに関する情報を取得するには

この例では、AWS CodeCommit リポジトリ内のブランチに関する情報を取得します。

コマンド:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

出力:

```
{
```

```
"BranchInfo": {
  "commitID": "317f8570EXAMPLE",
  "branchName": "MyNewBranch"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBranch](#)」の「」を参照してください。

get-comment-reactions

次の例は、get-comment-reactions を使用する方法を説明しています。

AWS CLI

コメントに対する絵文字の反応を表示するには

次のget-comment-reactions例では、 の ID を持つコメントに対するすべての絵文字リアクションを一覧表示しますabcd1234EXAMPLEb5678efgh。シエルのフォントが絵文字バージョン 1.0 の表示をサポートしている場合、絵文字emojiの出力に が表示されます。

```
aws codecommit get-comment-reactions \
  --comment-id abcd1234EXAMPLEb5678efgh
```

出力:

```
{
  "reactionsForComment": {
    [
      {
        "reaction": {
          "emoji:"??",
          "shortCode": "thumbsup",
          "unicode": "U+1F44D"
        },
        "users": [
          "arn:aws:iam::123456789012:user/Li_Juan",
          "arn:aws:iam::123456789012:user/Mary_Major",
          "arn:aws:iam::123456789012:user/Jorge_Souza"
        ]
      },
    ]
  },
}
```



```
{
  "reaction": {
    "emoji": "??",
    "shortCode": "thumbsdown",
    "unicode": "U+1F44E"
  },
  "users": [
    "arn:aws:iam::123456789012:user/Nikhil_Jayashankar"
  ]
},
{
  "reaction": {
    "emoji": "??",
    "shortCode": "confused",
    "unicode": "U+1F615"
  },
  "users": [
    "arn:aws:iam::123456789012:user/Saanvi_Sarkar"
  ]
}
]
```

詳細については、「ユーザーガイド」の「[でのコミットに関するコメント AWS CodeCommit](#)」を参照してください。AWS CodeCommit

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCommentReactions](#)」の「」を参照してください。

get-comment

次のコード例は、get-comment を使用する方法を示しています。

AWS CLI

コメントの詳細を表示するには

この例では、システム生成のコメント ID が のコメントの詳細を表示する方法を示します ff30b348EXAMPLEb9aa670f。

```
aws codecommit get-comment \
```

```
--comment-id ff30b348EXAMPLEb9aa670f
```

出力:

```
{
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "123Example",
    "commentId": "ff30b348EXAMPLEb9aa670f",
    "content": "Whoops - I meant to add this comment to the line, but I don't
see how to delete it.",
    "creationDate": 1508369768.142,
    "deleted": false,
    "commentId": "",
    "lastModifiedDate": 1508369842.278,
    "callerReactions": [],
    "reactionCounts":
    {
      "SMILE" : 6,
      "THUMBSUP" : 1
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetComment](#)」の「」を参照してください。

get-comments-for-compared-commit

次のコード例は、get-comments-for-compared-commit を使用する方法を示しています。

AWS CLI

コミットに関するコメントを表示するには

この例では、という名前のリポジトリ内の 2 つのコミットの比較に対して行われたコメントを表示する方法を示しますMyDemoRepo。

```
aws codecommit get-comments-for-compared-commit \
  --repository-name MyDemoRepo \
  --before-commit-ID 6e147360EXAMPLE \
  --after-commit-id 317f8570EXAMPLE
```

出力:

```
{
  "commentsForComparedCommitData": [
    {
      "afterBlobId": "1f330709EXAMPLE",
      "afterCommitId": "317f8570EXAMPLE",
      "beforeBlobId": "80906a4cEXAMPLE",
      "beforeCommitId": "6e147360EXAMPLE",
      "comments": [
        {
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
          "clientRequestToken": "123Example",
          "commentId": "ff30b348EXAMPLEeb9aa670f",
          "content": "Whoops - I meant to add this comment to the line,
not the file, but I don't see how to delete it.",
          "creationDate": 1508369768.142,
          "deleted": false,
          "CommentId": "123abc-EXAMPLE",
          "lastModifiedDate": 1508369842.278,
          "callerReactions": [],
          "reactionCounts":
            {
              "SMILE" : 6,
              "THUMBSUP" : 1
            }
        },
        {
          "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
          "clientRequestToken": "123Example",
          "commentId": "553b509bEXAMPLE56198325",
          "content": "Can you add a test case for this?",
          "creationDate": 1508369612.240,
          "deleted": false,
          "commentId": "456def-EXAMPLE",
          "lastModifiedDate": 1508369612.240,
          "callerReactions": [],
          "reactionCounts":
            {
              "THUMBSUP" : 2
            }
        }
      ],
      "location": {
```

```

        "filePath": "cl_sample.js",
        "filePosition": 1232,
        "relativeFileVersion": "after"
    },
    "repositoryName": "MyDemoRepo"
}
],
"nextToken": "exampleToken"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetCommentsForComparedCommit](#)」の「」を参照してください。

get-comments-for-pull-request

次の例は、get-comments-for-pull-request を使用する方法を説明しています。

AWS CLI

プルリクエストのコメントを表示するには

この例では、という名前のリポジトリでプルリクエストのコメントを表示する方法を示しますMyDemoRepo。

```

aws codecommit get-comments-for-pull-request \
  --repository-name MyDemoRepo \
  --before-commit-ID 317f8570EXAMPLE \
  --after-commit-id 5d036259EXAMPLE

```

出力:

```

{
  "commentsForPullRequestData": [
    {
      "afterBlobId": "1f330709EXAMPLE",
      "afterCommitId": "5d036259EXAMPLE",
      "beforeBlobId": "80906a4cEXAMPLE",
      "beforeCommitId": "317f8570EXAMPLE",
      "comments": [
        {
          "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",

```

```
    "clientRequestToken": "",
    "commentId": "abcd1234EXAMPLEb5678efgh",
    "content": "These don't appear to be used anywhere. Can we
remove them?",
    "creationDate": 1508369622.123,
    "deleted": false,
    "lastModifiedDate": 1508369622.123,
    "callerReactions": [],
    "reactionCounts":
    {
        "THUMBSUP" : 6,
        "CONFUSED" : 1
    }
},
{
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "",
    "commentId": "442b498bEXAMPLE5756813",
    "content": "Good catch. I'll remove them.",
    "creationDate": 1508369829.104,
    "deleted": false,
    "lastModifiedDate": 150836912.273,
    "callerReactions": ["THUMBSUP"]
    "reactionCounts":
    {
        "THUMBSUP" : 14
    }
}
],
"location": {
    "filePath": "ahs_count.py",
    "filePosition": 367,
    "relativeFileVersion": "AFTER"
},
"repositoryName": "MyDemoRepo",
"pullRequestId": "42"
}
],
"nextToken": "exampleToken"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCommentsForPullRequest](#)」の「」を参照してください。

get-commit

次の例は、get-commit を使用する方法を説明しています。

AWS CLI

リポジトリ内のコミットに関する情報を表示するには

この例では、「」という名前の AWS CodeCommit リポジトリで、システム生成 ID が「7e9fd3091thisisanexamplethisisanexample1」のコミットの詳細を示しています MyDemoRepo。

コマンド:

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-id 7e9fd3091thisisanexamplethisisanexample1
```

出力:

```
{
  "commit": {
    "additionalData": "",
    "committer": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    },
    "author": {
      "date": "1484167798 -0800",
      "name": "Mary Major",
      "email": "mary_major@example.com"
    },
    "treeId": "347a3408thisisanexampletreeidexample",
    "parents": [
      "7aa87a031thisisanexamplethisisanexample1"
    ],
    "message": "Fix incorrect variable name"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetCommit](#)」の「」を参照してください。

get-differences

次のコード例は、get-differences を使用する方法を示しています。

AWS CLI

リポジトリ内のコミット指定子の違いに関する情報を取得するには

この例では、 という名前の AWS CodeCommit リポジトリの名前が変更されたフォルダ内の 2 つのコミット指定子 (ブランチ、タグ、HEAD、またはコミット などの他の完全修飾参照IDs) 間の変更に関するメタデータ情報を表示します MyDemoRepo。この例では、これらのオプションを使用して結果を制限する方法をより詳細に説明するために、--before-commit-specifier、--before-path、--after-path など、不要なオプションがいくつかあります。レスポンスには、ファイルモードのアクセス許可が含まれます。

コマンド:

```
aws codecommit get-differences --repository-name MyDemoRepo --before-commit-specifier 955bba12thisisanexamplethisisanexample --after-commit-specifier 14a95463thisisanexamplethisisanexample --before-path tmp/example-folder --after-path tmp/renamed-folder
```

出力:

```
{
  "differences": [
    {
      "afterBlob": {
        "path": "blob.txt",
        "blobId": "2eb4af3b1thisisanexamplethisisanexample1",
        "mode": "100644"
      },
      "changeType": "M",
      "beforeBlob": {
        "path": "blob.txt",
        "blobId": "bf7fcf281thisisanexamplethisisanexample1",
        "mode": "100644"
      }
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDifferences](#)」の「」を参照してください。

get-file

次の例は、get-file を使用する方法を説明しています。

AWS CLI

AWS CodeCommit リポジトリ内のファイルの base-64 でエンコードされたコンテンツを取得するには

次のget-file例は、 という名前のリポジトリmain内の という名前のブランチREADME.mdから、 という名前のファイルの base-64 でエンコードされたコンテンツを取得する方法を示していますMyDemoRepo。

```
aws codecommit get-file \  
  --repository-name MyDemoRepo \  
  --commit-specifier main \  
  --file-path README.md
```

出力:

```
{  
  "blobId": "559b44fEXAMPLE",  
  "commitId": "c5709475EXAMPLE",  
  "fileContent": "IyBQaHVzEXAMPLE",  
  "filePath": "README.md",  
  "fileMode": "NORMAL",  
  "fileSize": 1563  
}
```

詳細については、AWS CodeCommit APIリファレンスガイドの[GetFile](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFile](#)」の「」を参照してください。

get-folder

次のコード例は、get-folder を使用する方法を示しています。

AWS CLI

AWS CodeCommit リポジトリ内のフォルダの内容を取得するには

次のget-folder例は、 という名前のリポジトリから最上位フォルダの内容を取得する方法を示していますMyDemoRepo。

```
aws codecommit get-folder --repository-name MyDemoRepo --folder-path ""
```

出力:

```
{
  "commitId":"c5709475EXAMPLE",
  "files":[
    {
      "absolutePath": ".gitignore",
      "blobId": "74094e8bEXAMPLE",
      "fileMode": "NORMAL",
      "relativePath": ".gitignore"
    },
    {
      "absolutePath": "Gemfile",
      "blobId": "9ceb72f6EXAMPLE",
      "fileMode": "NORMAL",
      "relativePath": "Gemfile"
    },
    {
      "absolutePath": "Gemfile.lock",
      "blobId": "795c4a2aEXAMPLE",
      "fileMode": "NORMAL",
      "relativePath": "Gemfile.lock"
    },
    {
      "absolutePath": "LICENSE.txt",
      "blobId": "0c7932c8EXAMPLE",
      "fileMode": "NORMAL",
      "relativePath": "LICENSE.txt"
    },
    {
      "absolutePath": "README.md",
      "blobId": "559b44feEXAMPLE",
      "fileMode": "NORMAL",
      "relativePath": "README.md"
    }
  ]
}
```

```
    }
  ],
  "folderPath": "",
  "subFolders": [
    {
      "absolutePath": "public",
      "relativePath": "public",
      "treeId": "d5e92ae3aEXAMPLE"
    },
    {
      "absolutePath": "tmp",
      "relativePath": "tmp",
      "treeId": "d564d0bcEXAMPLE"
    }
  ],
  "subModules": [],
  "symbolicLinks": [],
  "treeId": "7b3c4dadEXAMPLE"
}
```

詳細については、AWS CodeCommit APIリファレンスガイドの `GetFolder` 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFolder](#)」の「」を参照してください。

get-merge-commit

次の例は、`get-merge-commit` を使用する方法を説明しています。

AWS CLI

マージコミットに関する詳細情報を取得するには

次の`get-merge-commit`例では、`bugfix-bug1234`という名前のリポジトリの `THREE_WAY_MERGE` 戦略`main`を使用して、`bugfix-bug1234`の宛先ブランチのマージコミットの詳細を表示しますMyDemoRepo。

```
aws codecommit get-merge-commit \
  --source-commit-specifier bugfix-bug1234 \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE \
```

```
--repository-name MyDemoRepo
```

出力:

```
{
  "sourceCommitId": "c5709475EXAMPLE",
  "destinationCommitId": "317f8570EXAMPLE",
  "baseCommitId": "fb12a539EXAMPLE",
  "mergeCommitId": "ffc4d608eEXAMPLE"
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[コミットの詳細を表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMergeCommit](#)」の「」を参照してください。

get-merge-conflicts

次のコード例は、get-merge-conflicts を使用する方法を示しています。

AWS CLI

プルリクエストにマージ競合があるかどうかを表示するには

次のget-merge-conflicts例では、という名前のリポジトリで、という名前のソースブランチの先端feature-randomizationfeatureと 'main' という名前の宛先ブランチの間にマージ競合があるかどうかを示しますMyDemoRepo。

```
aws codecommit get-merge-conflicts \
  --repository-name MyDemoRepo \
  --source-commit-specifier feature-randomizationfeature \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE
```

出力:

```
{
  "mergeable": false,
  "destinationCommitId": "86958e0aEXAMPLE",
```

```
"sourceCommitId": "6ccd57fdEXAMPLE",
"baseCommitId": "767b6958EXAMPLE",
"conflictMetadataList": [
  {
    "filePath": "readme.md",
    "fileSizes": {
      "source": 139,
      "destination": 230,
      "base": 85
    },
    "fileModes": {
      "source": "NORMAL",
      "destination": "NORMAL",
      "base": "NORMAL"
    },
    "objectTypes": {
      "source": "FILE",
      "destination": "FILE",
      "base": "FILE"
    },
    "numberOfConflicts": 1,
    "isBinaryFile": {
      "source": false,
      "destination": false,
      "base": false
    },
    "contentConflict": true,
    "fileModeConflict": false,
    "objectTypeConflict": false,
    "mergeOperations": {
      "source": "M",
      "destination": "M"
    }
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMergeConflicts](#)」の「」を参照してください。

get-merge-options

次の例は、get-merge-options を使用する方法を説明しています。

AWS CLI

2つの指定されたブランチをマージするために利用可能なマージオプションに関する情報を取得するには

次のget-merge-options例では、`bugfix-bug1234`という名前のソースブランチを `main` という名前のリポジトリ `main` 内の `bugfix-bug1234` という名前の宛先ブランチとマージするために使用できるマージオプションを決定します `MyDemoRepo`。

```
aws codecommit get-merge-options \  
  --source-commit-specifier bugfix-bug1234 \  
  --destination-commit-specifier main \  
  --repository-name MyDemoRepo
```

出力:

```
{  
  "mergeOptions": [  
    "FAST_FORWARD_MERGE",  
    "SQUASH_MERGE",  
    "THREE_WAY_MERGE"  
  ],  
  "sourceCommitId": "18059494EXAMPLE",  
  "destinationCommitId": "ffd3311dEXAMPLE",  
  "baseCommitId": "ffd3311dEXAMPLE"  
}
```

詳細については、AWS CodeCommit [「ユーザーガイド」の「プルリクエストの競合の解決」](#)を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスGetMergeOptions」](#)の「」を参照してください。

get-pull-request-approval-states

次の例は、get-pull-request-approval-states を使用する方法を説明しています。

AWS CLI

プルリクエストの承認を表示するには

次のget-pull-request-approval-states例では、指定されたプルリクエストの承認を返します。

```
aws codecommit get-pull-request-approval-states \  
  --pull-request-id 8 \  
  --revision-id 9f29d167EXAMPLE
```

出力:

```
{  
  "approvals": [  
    {  
      "userArn": "arn:aws:iam::123456789012:user/Mary_Major",  
      "approvalState": "APPROVE"  
    }  
  ]  
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の[「プルリクエストの表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPullRequestApprovalStates](#)」の「」を参照してください。

get-pull-request-override-state

次のコード例は、get-pull-request-override-state を使用する方法を示しています。

AWS CLI

プルリクエストのオーバーライドステータスに関する情報を取得するには

次のget-pull-request-override-state例では、指定されたプルリクエストのオーバーライド状態を返します。この例では、プルリクエストの承認ルールが Mary Major という名前のユーザーによってオーバーライドされたため、出力は true を返します。

```
aws codecommit get-pull-request-override-state \  
  --pull-request-id 34 \  
  --revision-id 9f29d167EXAMPLE
```

出力:

```
{
  "overridden": true,
  "overrider": "arn:aws:iam::123456789012:user/Mary_Major"
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[プルリクエストの承認ルールの上書き](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPullRequestOverrideState](#)」の「」を参照してください。

get-pull-request

次のコード例は、get-pull-request を使用する方法を示しています。

AWS CLI

プルリクエストの詳細を表示するには

この例では、ID が のプルリクエストに関する情報を表示する方法を示します27。

```
aws codecommit get-pull-request \
  --pull-request-id 27
```

出力:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\": [
          {\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\": [
            \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "lastActivityDate": 1562619583.565,
```

```
"pullRequestTargets": [
  {
    "sourceCommit": "ca45e279EXAMPLE",
    "sourceReference": "refs/heads/bugfix-1234",
    "mergeBase": "a99f5ddbEXAMPLE",
    "destinationReference": "refs/heads/main",
    "mergeMetadata": {
      "isMerged": false
    },
    "destinationCommit": "2abfc6beEXAMPLE",
    "repositoryName": "MyDemoRepo"
  }
],
"revisionId": "e47def21EXAMPLE",
"title": "Quick fix for bug 1234",
"authorArn": "arn:aws:iam::123456789012:user/Nikhil_Jayashankar",
"clientRequestToken": "d8d7612e-EXAMPLE",
"creationDate": 1562619583.565,
"pullRequestId": "27",
"pullRequestStatus": "OPEN"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPullRequest](#)」の「」を参照してください。

get-repository-triggers

次の例は、get-repository-triggers を使用する方法を説明しています。

AWS CLI

リポジトリ内のトリガーに関する情報を取得するには

この例では、という名前の AWS CodeCommit リポジトリ用に設定されたトリガーの詳細を示していますMyDemoRepo。

```
aws codecommit get-repository-triggers \
  --repository-name MyDemoRepo
```

出力:


```
{
  "configurationId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
  "triggers": [
    {
      "destinationArn": "arn:aws:sns:us-
east-1:111111111111:MyCodeCommitTopic",
      "branches": [
        "main",
        "preprod"
      ],
      "name": "MyFirstTrigger",
      "customData": "",
      "events": [
        "all"
      ]
    },
    {
      "destinationArn": "arn:aws:lambda:us-
east-1:111111111111:function:MyCodeCommitPythonFunction",
      "branches": [],
      "name": "MySecondTrigger",
      "customData": "EXAMPLE",
      "events": [
        "all"
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRepositoryTriggers](#)」の「」を参照してください。

get-repository

次のコード例は、get-repository を使用する方法を示しています。

AWS CLI

リポジトリに関する情報を取得するには

この例では、AWS CodeCommit リポジトリの詳細を示しています。

```
aws codecommit get-repository \  
  --repository-name MyDemoRepo
```

出力:

```
{  
  "repositoryMetadata": {  
    "creationDate": 1429203623.625,  
    "defaultBranch": "main",  
    "repositoryName": "MyDemoRepo",  
    "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/v1/  
repos/MyDemoRepo",  
    "lastModifiedDate": 1430783812.0869999,  
    "repositoryDescription": "My demonstration repository",  
    "cloneUrlHttp": "https://codecommit.us-east-1.amazonaws.com/v1/repos/  
MyDemoRepo",  
    "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",  
    "Arn": "arn:aws:codecommit:us-east-1:80398EXAMPLE:MyDemoRepo",  
    "accountId": "111111111111"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRepository](#)」の「」を参照してください。

list-approval-rule-templates

次の例は、list-approval-rule-templates を使用する方法を説明しています。

AWS CLI

AWS リージョン内のすべての承認ルールテンプレートを一覧表示するには

次のlist-approval-rule-templates例では、指定されたリージョンのすべての承認ルールテンプレートを一覧表示します。AWS リージョンがパラメータとして指定されていない場合、コマンドはコマンドの実行に使用されるプロファイルで AWS CLI指定されたリージョンの承認ルールテンプレートを返します。

```
aws codecommit list-approval-rule-templates \  
  --region us-east-2
```

出力:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「承認ルールテンプレートの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListApprovalRuleTemplates](#)」の「」を参照してください。

list-associated-approval-rule-templates-for-repository

次のコード例は、`list-associated-approval-rule-templates-for-repository` を使用する方法を示しています。

AWS CLI

リポジトリに関連付けられているすべてのテンプレートを一覧表示するには

次の `list-associated-approval-rule-templates-for-repository` 例では、という名前のリポジトリに関連付けられたすべての承認ルールテンプレートを一覧表示します `MyDemoRepo`。

```
aws codecommit list-associated-approval-rule-templates-for-repository \
  --repository-name MyDemoRepo
```

出力:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「承認ルールテンプレートの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAssociatedApprovalRuleTemplatesForRepository](#)」の「」を参照してください。

list-branches

次の例は、list-branches を使用する方法を説明しています。

AWS CLI

ブランチ名のリストを表示するには

この例では、AWS CodeCommit リポジトリ内のすべてのブランチ名を一覧表示します。

```
aws codecommit list-branches \  
  --repository-name MyDemoRepo
```

出力:

```
{  
  "branches": [  
    "MyNewBranch",  
    "main"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListBranches](#)」の「」を参照してください。

list-pull-requests

次のコード例は、list-pull-requests を使用する方法を示しています。

AWS CLI

リポジトリ内のプルリクエストのリストを表示するには

この例では、「arn:aws:iam::111111111111:user/Li_Juan」という名前の AWS CodeCommit リポジトリ CLOSED で ARN、「arn:aws:iam::user/Li_Juan」のステータスを持つ IAM ユーザーが作成したプルリクエストを一覧表示する方法を示します MyDemoRepo。

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

出力:

```
{
  "nextToken": "",
  "pullRequestIds": ["2", "12", "16", "22", "23", "35", "30", "39", "47"]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPullRequests](#)」の「」を参照してください。

list-repositories-for-approval-rule-template

次のコード例は、`list-repositories-for-approval-rule-template` を使用方法を示しています。

AWS CLI

テンプレートに関連付けられているすべてのリポジトリを一覧表示するには

次の`list-repositories-for-approval-rule-template`例では、指定された承認ルールテンプレートに関連付けられているすべてのリポジトリを一覧表示します。

```
aws codecommit list-repositories-for-approval-rule-template \
  --approval-rule-template-name 2-approver-rule-for-main
```

出力:

```
{
  "repositoryNames": [
    "MyDemoRepo",
    "MyClonedRepo"
  ]
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[承認ルールテンプレートの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListRepositoriesForApprovalRuleTemplate](#)」の「」を参照してください。

list-repositories

次の例は、list-repositories を使用する方法を説明しています。

AWS CLI

リポジトリのリストを表示するには

この例では、ユーザーの AWS アカウントに関連付けられているすべての AWS CodeCommit リポジトリを一覧表示します。

コマンド:

```
aws codecommit list-repositories
```

出力:

```
{
  "repositories": [
    {
      "repositoryName": "MyDemoRepo"
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    },
    {
      "repositoryName": "MyOtherDemoRepo"
      "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListRepositories](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リポジトリの AWS タグを表示するには

次のlist-tags-for-resource例では、指定されたりポジトリのタグキーとタグ値を一覧表示します。

```
aws codecommit list-tags-for-resource \  
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

出力:

```
{  
  "tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

詳細については、AWS CodeCommit 「[ユーザーガイド](#)」の「[リポジトリのタグの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListTagsForResource](#)」の「」を参照してください。

merge-branches-by-fast-forward

次のコード例は、merge-branches-by-fast-forward を使用する方法を示しています。

AWS CLI

早送りマージ戦略を使用して 2 つのブランチをマージするには

次のmerge-branches-by-fast-forward例では、指定されたソースブランチを、という名前のリポジトリ内の指定された宛先ブランチとマージしますMyDemoRepo。

```
aws codecommit merge-branches-by-fast-forward \  
  --source-commit-specifier bugfix-bug1234 \  
  --destination-commit-specifier bugfix-bug1233 \  
  --repository-name MyDemoRepo
```

出力:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「ブランチの比較とマージ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [MergeBranchesByFastForward](#)」の「」を参照してください。

merge-branches-by-squash

次の例は、merge-branches-by-squash を使用する方法を説明しています。

AWS CLI

スカッシュマージ戦略を使用して 2 つのブランチをマージするには

次のmerge-branches-by-squash例では、指定されたソースブランチを、という名前のリポジトリ内の指定された宛先ブランチとマージしますMyDemoRepo。

```
aws codecommit merge-branches-by-squash \
  --source-commit-specifier bugfix-bug1234 \
  --destination-commit-specifier bugfix-bug1233 \
  --author-name "Maria Garcia" \
  --email "maria_garcia@example.com" \
  --commit-message "Merging two fix branches to prepare for a general patch." \
  --repository-name MyDemoRepo
```

出力:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「ブランチの比較とマージ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[MergeBranchesBySquash](#)」の「」を参照してください。

merge-branches-by-three-way

次のコード例は、merge-branches-by-three-way を使用する方法を示しています。

AWS CLI

3 方向マージ戦略を使用して 2 つのブランチをマージするには

次のmerge-branches-by-three-way例では、指定されたソースブランチを、 という名前のリポジトリ内の指定された宛先ブランチとマージしますMyDemoRepo。

```
aws codecommit merge-branches-by-three-way \  
  --source-commit-specifier main \  
  --destination-commit-specifier bugfix-bug1234 \  
  --author-name "Jorge Souza" --email "jorge_souza@example.com" \  
  --commit-message "Merging changes from main to bugfix branch before additional testing." \  
  --repository-name MyDemoRepo
```

出力:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[ブランチの比較とマージ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[MergeBranchesByThreeWay](#)」の「」を参照してください。

merge-pull-request-by-fast-forward

次のコード例は、merge-pull-request-by-fast-forward を使用する方法を示しています。

AWS CLI

プルリクエストをマージして閉じるには

この例では、という名前のリポジトリで、プルリクエストを '47' の ID と '99132ab0EXAMPLE' のソースコミット ID でマージして閉じる方法を示しますMyDemoRepo。

```
aws codecommit merge-pull-request-by-fast-forward \  
  --pull-request-id 47 \  
  --source-commit-id 99132ab0EXAMPLE \  
  --repository-name MyDemoRepo
```

出力:

```
{  
  "pullRequest": {  
    "approvalRules": [  
      {  
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":  
[{\ \"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\":  
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",  
        "approvalRuleId": "dd8b17fe-EXAMPLE",  
        "approvalRuleName": "I want one approver for this pull request",  
        "creationDate": 1571356106.936,  
        "lastModifiedDate": 571356106.936,  
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",  
        "ruleContentSha256": "4711b576EXAMPLE"  
      }  
    ],  
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",  
    "clientRequestToken": "",  
    "creationDate": 1508530823.142,  
    "description": "Review the latest changes and updates to the global  
variables",  
    "lastActivityDate": 1508887223.155,  
    "pullRequestId": "47",  
    "pullRequestStatus": "CLOSED",  
    "pullRequestTargets": [  
      {  
        "destinationCommit": "9f31c968EXAMPLE",  
        "destinationReference": "refs/heads/main",  
        "mergeMetadata": {  
          "isMerged": true,  
          "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"  
        },  
        "repositoryName": "MyDemoRepo",  
        "sourceCommit": "99132ab0EXAMPLE",
```

```

        "sourceReference": "refs/heads/variables-branch"
      }
    ],
    "title": "Consolidation of global variables"
  }
}

```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[プルリクエストのマージ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[MergePullRequestByFastForward](#)」の「」を参照してください。

merge-pull-request-by-squash

次の例は、merge-pull-request-by-squash を使用する方法を説明しています。

AWS CLI

スカッシュマージ戦略を使用してプルリクエストをマージするには

次のmerge-pull-request-by-squash例では、という名前のリポジトリでACCEPT_SOURCE の競合解決戦略を使用して、指定されたプルリクエストをマージして閉じますMyDemoRepo。

```

aws codecommit merge-pull-request-by-squash \
  --pull-request-id 47 \
  --source-commit-id 99132ab0EXAMPLE \
  --repository-name MyDemoRepo \
  --conflict-detail-level LINE_LEVEL \
  --conflict-resolution-strategy ACCEPT_SOURCE \
  --name "Jorge Souza" --email "jorge_souza@example.com" \
  --commit-message "Merging pull request 47 by squash and accepting source in
merge conflicts"

```

出力:

```

{
  "pullRequest": {
    "approvalRules": [
      {

```

```

        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
\\\"DestinationReferences\\\": [\\\"refs/heads/main\\\"],\\\"Statements\\\": [{\\\"Type
\\\": \\\"Approvers\\\",\\\"NumberOfApprovalsNeeded\\\": 2,\\\"ApprovalPoolMembers\\\":
[\\\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\\\"]}]}\",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
            "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",
            "approvalRuleTemplateName": "2-approver-rule-for-main"
        },
        "ruleContentSha256": "4711b576EXAMPLE"
    }
],
"authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
"clientRequestToken": "",
"creationDate": 1508530823.142,
"description": "Review the latest changes and updates to the global
variables",
"lastActivityDate": 1508887223.155,
"pullRequestId": "47",
"pullRequestStatus": "CLOSED",
"pullRequestTargets": [
    {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
            "isMerged": true,
            "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
    }
],
"title": "Consolidation of global variables"
}
}

```

詳細については、AWS CodeCommit ユーザーガイドの [「プルリクエストのマージ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[MergePullRequestBySquash](#)」の「」を参照してください。

merge-pull-request-by-three-way

次のコード例は、merge-pull-request-by-three-way を使用する方法を示しています。

AWS CLI

3 方向マージ戦略を使用してプルリクエストをマージするには

次のmerge-pull-request-by-three-way例では、 という名前のリポジトリの競合の詳細と競合解決戦略のデフォルトオプションを使用して、指定されたプルリクエストをマージして閉じますMyDemoRepo。

```
aws codecommit merge-pull-request-by-three-way \  
  --pull-request-id 47 \  
  --source-commit-id 99132ab0EXAMPLE \  
  --repository-name MyDemoRepo \  
  --name "Maria Garcia" \  
  --email "maria_garcia@example.com" \  
  --commit-message "Merging pull request 47 by three-way with default options"
```

出力:

```
{  
  "pullRequest": {  
    "approvalRules": [  
      {  
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",  
\"DestinationReferences\": [\"refs/heads/main\"],\"Statements\": [{\"Type  
\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":  
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",  
        "approvalRuleId": "dd8b17fe-EXAMPLE",  
        "approvalRuleName": "2-approver-rule-for-main",  
        "creationDate": 1571356106.936,  
        "lastModifiedDate": 571356106.936,  
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",  
        "originApprovalRuleTemplate": {  
          "approvalRuleTemplateId": "dd8b17fe-EXAMPLE",  
          "approvalRuleTemplateName": "2-approver-rule-for-main"
```

```
        },
        "ruleContentSha256": "4711b576EXAMPLE"
    }
],
"authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
"clientRequestToken": "",
"creationDate": 1508530823.142,
"description": "Review the latest changes and updates to the global
variables",
"lastActivityDate": 1508887223.155,
"pullRequestId": "47",
"pullRequestStatus": "CLOSED",
"pullRequestTargets": [
    {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
            "isMerged": true,
            "mergedBy": "arn:aws:iam::123456789012:user/Mary_Major"
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
    }
],
"title": "Consolidation of global variables"
}
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[プルリクエストのマージ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[MergePullRequestByThreeWay](#)」の「」を参照してください。

override-pull-request-approval-rules

次のコード例は、`override-pull-request-approval-rules` を使用する方法を示しています。

AWS CLI

プルリクエストの承認ルール要件を上書きするには

次のoverride-pull-request-approval-rules例では、指定されたプルリクエストの承認ルールを上書きします。代わりにオーバーライドを取り消すには、--override-statusパラメータ値を に設定しますREVOKE。

```
aws codecommit override-pull-request-approval-rules \  
  --pull-request-id 34 \  
  --revision-id 927df8d8EXAMPLE \  
  --override-status OVERRIDE
```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit 「[ユーザーガイド](#)」の「[プルリクエストの承認ルールの上書き](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスOverridePullRequestApprovalRules](#)」の「」を参照してください。

post-comment-for-compared-commit

次の例は、post-comment-for-compared-commit を使用する方法を説明しています。

AWS CLI

コミットに関するコメントを作成するには

この例では、 という名前のリポジトリ内の 2 つのコミットを比較して、変更"Can you add a test case for this?"に関するコメントを cl_sample.js ファイルに追加する方法を示しますMyDemoRepo。

```
aws codecommit post-comment-for-compared-commit \  
  --repository-name MyDemoRepo \  
  --before-commit-id 317f8570EXAMPLE \  
  --after-commit-id 5d036259EXAMPLE \  
  --client-request-token 123Example \  
  --content "Can you add a test case for this?" \  
  --location filePath=cl_sample.js,filePosition=1232,relativeFileVersion=AFTER
```

出力:

```
{
```

```

"afterBlobId": "1f330709EXAMPLE",
"afterCommitId": "317f8570EXAMPLE",
"beforeBlobId": "80906a4cEXAMPLE",
"beforeCommitId": "6e147360EXAMPLE",
"comment": {
  "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
  "clientRequestToken": "",
  "commentId": "553b509bEXAMPLE56198325",
  "content": "Can you add a test case for this?",
  "creationDate": 1508369612.203,
  "deleted": false,
  "commentId": "abc123-EXAMPLE",
  "lastModifiedDate": 1508369612.203,
  "callerReactions": [],
  "reactionCounts": []
},
"location": {
  "filePath": "cl_sample.js",
  "filePosition": 1232,
  "relativeFileVersion": "AFTER"
,
"repositoryName": "MyDemoRepo"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [PostCommentForComparedCommit](#)」の「」を参照してください。

post-comment-for-pull-request

次のコード例は、`post-comment-for-pull-request` を使用する方法を示しています。

AWS CLI

プルリクエストにコメントを追加するには

次の `post-comment-for-pull-request` 例では、「これらはどこにも使用されていないようです。削除できますか。」という名前のリポジトリ内の ID が のプルリクエスト47の `ahs_count.py` ファイルへの変更 `MyDemoRepo`。

```

aws codecommit post-comment-for-pull-request \
  --pull-request-id "47" \

```



```
--repository-name MyDemoRepo \  
--before-commit-id 317f8570EXAMPLE \  
--after-commit-id 5d036259EXAMPLE \  
--client-request-token 123Example \  
--content "These don't appear to be used anywhere. Can we remove them?" \  
--location filePath=ahs_count.py,filePosition=367,relativeFileVersion=AFTER
```

出力:

```
{  
  "afterBlobId": "1f330709EXAMPLE",  
  "afterCommitId": "5d036259EXAMPLE",  
  "beforeBlobId": "80906a4cEXAMPLE",  
  "beforeCommitId": "317f8570EXAMPLE",  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",  
    "clientRequestToken": "123Example",  
    "commentId": "abcd1234EXAMPLEb5678efgh",  
    "content": "These don't appear to be used anywhere. Can we remove  
them?",  
    "creationDate": 1508369622.123,  
    "deleted": false,  
    "CommentId": "",  
    "lastModifiedDate": 1508369622.123,  
    "callerReactions": [],  
    "reactionCounts": []  
  },  
  "location": {  
    "filePath": "ahs_count.py",  
    "filePosition": 367,  
    "relativeFileVersion": "AFTER"  
  },  
  "repositoryName": "MyDemoRepo",  
  "pullRequestId": "47"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PostCommentForPullRequest](#)」の「」を参照してください。

post-comment-reply

次の例は、post-comment-reply を使用する方法を説明しています。

AWS CLI

コミットまたはプルリクエストのコメントに返信するには

この例では、システム生成 ID が のコメント "Good catch. I'll remove them." に返信を追加する方法を示します `abcd1234EXAMPLEb5678efgh`。

```
aws codecommit post-comment-reply \  
  --in-reply-to abcd1234EXAMPLEb5678efgh \  
  --content "Good catch. I'll remove them." \  
  --client-request-token 123Example
```

出力:

```
{  
  "comment": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "123Example",  
    "commentId": "442b498bEXAMPLE5756813",  
    "content": "Good catch. I'll remove them.",  
    "creationDate": 1508369829.136,  
    "deleted": false,  
    "CommentId": "abcd1234EXAMPLEb5678efgh",  
    "lastModifiedDate": 150836912.221,  
    "callerReactions": [],  
    "reactionCounts": []  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PostCommentReply](#)」の「」を参照してください。

put-comment-reaction

次の例は、`put-comment-reaction` を使用する方法を説明しています。

AWS CLI

絵文字でコミットに関するコメントに返信するには

次の `put-comment-reaction` 例は、絵文字リアクション値が の ID `abcd1234EXAMPLEb5678efgh` を持つコメントに返信します: `thumbsup:`。

```
aws codecommit put-comment-reaction \  
  --comment-id abcd1234EXAMPLEb5678efgh \  
  --reaction-value :thumbsup:
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[でのコミットに関するコメント AWS CodeCommit](#)」を参照してください。AWS CodeCommit

- API 詳細については、AWS CLI 「コマンドリファレンス[PutCommentReaction](#)」の「」を参照してください。

put-file

次の例は、put-file を使用する方法を説明しています。

AWS CLI

リポジトリにファイルを追加するには

次のput-file例では、'ExampleSolution.py' という名前のファイルを、'MyDemoRepo' という名前のリポジトリに、最新のコミットの ID が '4c925148' の 'feature-randomizationfeatureEXAMPLE' という名前のブランチに追加します。

```
aws codecommit put-file \  
  --repository-name MyDemoRepo \  
  --branch-name feature-randomizationfeature \  
  --file-content file://MyDirectory/ExampleSolution.py \  
  --file-path /solutions/ExampleSolution.py \  
  --parent-commit-id 4c925148EXAMPLE \  
  --name "Maria Garcia" \  
  --email "maria_garcia@example.com" \  
  --commit-message "I added a third randomization routine."
```

出力:

```
{  
  "blobId": "2eb4af3bEXAMPLE",  
  "commitId": "317f8570EXAMPLE",  
  "treeId": "347a3408EXAMPLE"
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutFile](#)」の「」を参照してください。

put-repository-triggers

次の例は、put-repository-triggers を使用する方法を説明しています。

AWS CLI

リポジトリでトリガーを追加または更新するには

この例では、 という名前のリポジトリのすべてのトリガーの構造を含む、既に作成されたJSONファイル (ここでは MyTriggers.json という名前) を使用して、MyFirstTrigger" と 'MySecondTrigger' という名前のトリガーを更新する方法を示しています MyDemoRepo。既存のトリガーJSONの を取得する方法については、 コマンドを参照してください get-repository-triggers。

```
aws codecommit put-repository-triggers \  
  --repository-name MyDemoRepo file://MyTriggers.json
```

MyTriggers.json の内容:

```
{  
  "repositoryName": "MyDemoRepo",  
  "triggers": [  
    {  
      "destinationArn": "arn:aws:sns:us-  
east-1:80398EXAMPLE:MyCodeCommitTopic",  
      "branches": [  
        "main",  
        "preprod"  
      ],  
      "name": "MyFirstTrigger",  
      "customData": "",  
      "events": [  
        "all"  
      ]  
    },  
    {
```

```

        "destinationArn": "arn:aws:lambda:us-
east-1:111111111111:function:MyCodeCommitPythonFunction",
        "branches": [],
        "name": "MySecondTrigger",
        "customData": "EXAMPLE",
        "events": [
            "all"
        ]
    }
]
}

```

出力:

```

{
    "configurationId": "6fa51cd8-35c1-EXAMPLE"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRepositoryTriggers](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

既存のリポジトリに AWS タグを追加するには

次のtag-resource例では、指定されたりポジトリに 2 つのタグをタグ付けします。

```

aws codecommit tag-resource \
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
  --tags Status=Secret,Team=Saanvi

```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit 「ユーザーガイド」の「[リポジトリにタグを追加する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

test-repository-triggers

次のコード例は、test-repository-triggers を使用する方法を示しています。

AWS CLI

リポジトリでトリガーをテストするには

この例では、 という名前の AWS CodeCommit リポジトリで 'MyFirstTrigger' という名前のトリガーをテストする方法を示します MyDemoRepo。この例では、リポジトリのイベントは Amazon Simple Notification Service (Amazon SNS) トピックから通知をトリガーします。

コマンド:

```
aws codecommit test-repository-triggers --repository-name MyDemoRepo
--triggers name=MyFirstTrigger,destinationArn=arn:aws:sns:us-east-1:111111111111:MyCodeCommitTopic,branches=mainline,preprod,events=all
```

出力:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [TestRepositoryTriggers](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リポジトリから AWS タグを削除するには

次の untag-resource 例では、 という名前のリポジトリから、指定されたキーを持つ タグを削除します MyDemoRepo。

```
aws codecommit untag-resource \  
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \  
  --tag-keys Status
```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit ユーザーガイドの「[リポジトリからタグを削除する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-approval-rule-template-content

次のコード例は、update-approval-rule-template-content を使用する方法を示しています。

AWS CLI

承認ルールテンプレートの内容を更新するには

次のupdate-approval-rule-template-content例では、指定された承認ルールテンプレートの内容を変更して、のロールを引き受けるユーザーに承認プールを再定義しますCodeCommitReview。

```
aws codecommit update-approval-rule-template-content \  
  --approval-rule-template-name 1-approver-rule \  
  --new-rule-content '{"Version": "2018-11-08", "DestinationReferences": [{"refs/heads/main"}], "Statements": [{"Type": "Approver", "NumberOfApprovalsNeeded": 2, "ApprovalPoolMembers": [{"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*"}]}'
```

出力:

```
{  
  "approvalRuleTemplate": {  
    "creationDate": 1571352720.773,  
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull requests from the CodeCommitReview pool",  
    "lastModifiedDate": 1571358728.41,  
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
```

```

    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\": [
    {\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\": [
    {\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"}]}]\",
    \"approvalRuleTemplateName\": \"1-approver-rule-for-all-pull-requests\",
    \"ruleContentSha256\": \"2f6c21a5EXAMPLE\",
    \"lastModifiedUser\": \"arn:aws:iam::123456789012:user/Li_Juan\"
  }
}

```

詳細については、AWS CodeCommit ユーザーガイドの [「承認ルールテンプレートの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateApprovalRuleTemplateContent](#)」の「」を参照してください。

update-approval-rule-template-description

次の例は、update-approval-rule-template-description を使用する方法を説明していません。

AWS CLI

承認ルールテンプレートの説明を更新するには

次のupdate-approval-rule-template-description例では、指定された承認ルールテンプレートの説明を *Requires 1 approval for all pull requests from the CodeCommitReview pool.* に変更します。

```

aws codecommit update-approval-rule-template-description \
  --approval-rule-template-name 1-approver-rule-for-all-pull-requests \
  --approval-rule-template-description "Requires 1 approval for all pull requests from the CodeCommitReview pool"

```

出力:

```

{
  "approvalRuleTemplate": {
    "creationDate": 1571352720.773,
    "approvalRuleTemplateDescription": "Requires 1 approval for all pull requests from the CodeCommitReview pool",
    "lastModifiedDate": 1571358728.41,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
  }
}

```



```

    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\":
    [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\":
    [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "ruleContentSha256": "2f6c21a5EXAMPLE",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Li_Juan"
  }
}

```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「承認ルールテンプレートの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateApprovalRuleTemplateDescription](#)」の「」を参照してください。

update-approval-rule-template-name

次の例は、update-approval-rule-template-name を使用する方法を説明しています。

AWS CLI

承認ルールテンプレートの名前を更新するには

次のupdate-approval-rule-template-name例では、承認ルールテンプレートの名前を 1-approver-rule から 1-approver-rule-for-all-pull-requests` に変更します。

```

aws codecommit update-approval-rule-template-name \
  --old-approval-rule-template-name 1-approver-rule \
  --new-approval-rule-template-name 1-approver-rule-for-all-pull-requests

```

出力:

```

{
  "approvalRuleTemplate": {
    "approvalRuleTemplateName": "1-approver-rule-for-all-pull-requests",
    "lastModifiedDate": 1571358241.619,
    "approvalRuleTemplateId": "41de97b7-EXAMPLE",
    "approvalRuleTemplateContent": "{\"Version\": \"2018-11-08\", \"Statements\":
    [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 1, \"ApprovalPoolMembers\":
    [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "creationDate": 1571352720.773,
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
  }
}

```

```
"approvalRuleTemplateDescription": "All pull requests must be approved by one
developer on the team.",
"ruleContentSha256": "2f6c21a5cEXAMPLE"
}
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[承認ルールテンプレートの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateApprovalRuleTemplateName](#)」の「」を参照してください。

update-comment

次の例は、update-comment を使用する方法を説明しています。

AWS CLI

コミットに関するコメントを更新するには

この例では、ID が のコメント "Fixed as requested. I'll update the pull request." にコンテンツを追加する方法を示します 442b498bEXAMPLE5756813。

```
aws codecommit update-comment \
  --comment-id 442b498bEXAMPLE5756813 \
  --content "Fixed as requested. I'll update the pull request."
```

出力:

```
{
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "clientRequestToken": "",
    "commentId": "442b498bEXAMPLE5756813",
    "content": "Fixed as requested. I'll update the pull request.",
    "creationDate": 1508369929.783,
    "deleted": false,
    "lastModifiedDate": 1508369929.287,
    "callerReactions": [],
    "reactionCounts":
      {
        "THUMBSUP" : 2
      }
  }
}
```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateComment](#)」の「」を参照してください。

update-default-branch

次のコード例は、update-default-branch を使用する方法を示しています。

AWS CLI

リポジトリのデフォルトブランチを変更するには

この例では、AWS CodeCommit リポジトリのデフォルトのブランチを変更します。このコマンドは、エラーがある場合にのみ出力を生成します。

コマンド:

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-name MyNewBranch
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDefaultBranch](#)」の「」を参照してください。

update-pull-request-approval-rule-content

次の例は、update-pull-request-approval-rule-content を使用する方法を説明しています。

AWS CLI

プルリクエストの承認ルールを編集するには

次のupdate-pull-request-approval-rule-content例では、123456789012 AWS アカウント内の任意のユーザーを含む承認プールから 1 人の IAM ユーザー承認を要求するように、指定した承認ルールを更新しています。

```
aws codecommit update-pull-request-approval-rule-content \
  --pull-request-id 27 \
  --approval-rule-name "Require two approved approvers" \
  --approval-rule-content "{Version: 2018-11-08, Statements: [{Type:
  \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:
  [\"CodeCommitApprovers:123456789012:user/*\"]}]}"
```

出力:

```
{
  "approvalRule": {
    "approvalRuleContent": "{Version: 2018-11-08, Statements:
    [{Type: \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers:
    [\"CodeCommitApprovers:123456789012:user/*\"]}]}",
    "approvalRuleId": "aac33506-EXAMPLE",
    "originApprovalRuleTemplate": {},
    "creationDate": 1570752871.932,
    "lastModifiedDate": 1570754058.333,
    "approvalRuleName": "Require two approved approvers",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
    "ruleContentSha256": "cd93921cEXAMPLE",
  }
}
```

詳細については、AWS CodeCommit ユーザーガイドの [「承認ルールの編集または削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdatePullRequestApprovalRuleContent](#)」の「」を参照してください。

update-pull-request-approval-state

次のコード例は、update-pull-request-approval-state を使用する方法を示しています。

AWS CLI

プルリクエストの承認または取り消しを行うには

次のupdate-pull-request-approval-state例では、の ID 27と のリビジョン ID を持つプルリクエストを承認します9f29d167EXAMPLE。代わりに承認を取り消す場合は、--approval-stateパラメータ値を に設定しますREVOKE。

```
aws codecommit update-pull-request-approval-state \  
  --pull-request-id 27 \  
  --revision-id 9f29d167EXAMPLE \  
  --approval-state "APPROVE"
```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit ユーザーガイドの「[プルリクエストの確認](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePullRequestApprovalState](#)」の「」を参照してください。

update-pull-request-description

次のコード例は、update-pull-request-description を使用する方法を示しています。

AWS CLI

プルリクエストの説明を変更するには

この例では、 の ID でプルリクエストの説明を変更する方法を示します47。

```
aws codecommit update-pull-request-description \  
  --pull-request-id 47 \  
  --description "Updated the pull request to remove unused global variable."
```

出力:

```
{  
  "pullRequest": {  
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",  
    "clientRequestToken": "",  
    "creationDate": 1508530823.155,  
    "description": "Updated the pull request to remove unused global variable.",  
    "lastActivityDate": 1508372423.204,  
    "pullRequestId": "47",  
    "pullRequestStatus": "OPEN",  
    "pullRequestTargets": [  
      {  
        "destinationCommit": "9f31c968EXAMPLE",  
        "destinationReference": "refs/heads/main",
```

```

        "mergeMetadata": {
            "isMerged": false,
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
    }
],
"title": "Consolidation of global variables"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdatePullRequestDescription](#)」の「」を参照してください。

update-pull-request-status

次の例は、update-pull-request-status を使用する方法を説明しています。

AWS CLI

プルリクエストのステータスを変更するには

この例では、ID が のプルリクエストのステータス42を、 という名前の AWS CodeCommit リポジトリCLOSED内の のステータスに変更する方法を示しますMyDemoRepo。

```

aws codecommit update-pull-request-status \
  --pull-request-id 42 \
  --pull-request-status CLOSED

```

出力:

```

{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\": [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\": [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approvers-needed-for-this-change",
        "creationDate": 1571356106.936,

```

```
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "ruleContentSha256": "4711b576EXAMPLE"
    }
],
"authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
"clientRequestToken": "",
"creationDate": 1508530823.165,
"description": "Updated the pull request to remove unused global variable.",
"lastActivityDate": 1508372423.12,
"pullRequestId": "47",
"pullRequestStatus": "CLOSED",
"pullRequestTargets": [
    {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
            "isMerged": false,
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
    }
],
"title": "Consolidation of global variables"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePullRequestStatus](#)」の「」を参照してください。

update-pull-request-title

次の例は、update-pull-request-title を使用する方法を説明しています。

AWS CLI

プルリクエストのタイトルを変更するには

この例では、 の ID でプルリクエストのタイトルを変更する方法を示します47。

```
aws codecommit update-pull-request-title \  
--pull-request-id 47 \  

```

```
--title "Consolidation of global variables - updated review"
```

出力:

```
{
  "pullRequest": {
    "approvalRules": [
      {
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",
        \"DestinationReferences\": [\"refs/heads/main\"], \"Statements\": [{\"Type
        \": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":
        [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "approvalRuleId": "dd8b17fe-EXAMPLE",
        "approvalRuleName": "2-approver-rule-for-main",
        "creationDate": 1571356106.936,
        "lastModifiedDate": 571356106.936,
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
        "originApprovalRuleTemplate": {
          "approvalRuleTemplateId": "dd8b26gr-EXAMPLE",
          "approvalRuleTemplateName": "2-approver-rule-for-main"
        },
        "ruleContentSha256": "4711b576EXAMPLE"
      }
    ],
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
    "clientRequestToken": "",
    "creationDate": 1508530823.12,
    "description": "Review the latest changes and updates to the global
    variables. I have updated this request with some changes, including removing some
    unused variables.",
    "lastActivityDate": 1508372657.188,
    "pullRequestId": "47",
    "pullRequestStatus": "OPEN",
    "pullRequestTargets": [
      {
        "destinationCommit": "9f31c968EXAMPLE",
        "destinationReference": "refs/heads/main",
        "mergeMetadata": {
          "isMerged": false,
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
      }
    ]
  }
}
```



```
    }  
  ],  
  "title": "Consolidation of global variables - updated review"  
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePullRequestTitle](#)」の「」を参照してください。

update-repository-description

次のコード例は、update-repository-description を使用する方法を示しています。

AWS CLI

リポジトリの説明を変更するには

この例では、AWS CodeCommit リポジトリの説明を変更します。このコマンドは、エラーがある場合にのみ出力を生成します。

コマンド:

```
aws codecommit update-repository-description --repository-name MyDemoRepo --  
repository-description "This description was changed"
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRepositoryDescription](#)」の「」を参照してください。

update-repository-name

次のコード例は、update-repository-name を使用する方法を示しています。

AWS CLI

リポジトリの名前を変更するには

この例では、AWS CodeCommit リポジトリの名前を変更します。このコマンドは、エラーがある場合にのみ出力を生成します。AWS CodeCommit リポジトリの名前を変更するSSHと、とが変更されHTTPSURLs、ユーザーはリポジトリに接続する必要があります。ユーザーは、接続設定を更新するまでこのリポジトリに接続できなくなります。また、リポジトリのARNが変更されるため、リポジトリ名を変更すると、このリポジトリの に依存するすべてのIAMユーザーポリシーが無効になりますARN。

コマンド:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-name MyRenamedDemoRepo
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRepositoryName](#)」の「」を参照してください。

CodeDeploy を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CodeDeploy。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags-to-on-premises-instances

次の例は、add-tags-to-on-premises-instances を使用する方法を説明しています。

AWS CLI

オンプレミスインスタンスにタグを追加するには

次のadd-tags-to-on-premises-instances例では AWS CodeDeploy、同じオンプレミスインスタンスタグ内の 2 つのオンプレミスインスタンスを関連付けます。オンプレミスインスタンスは に登録されません AWS CodeDeploy。

```
aws deploy add-tags-to-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTagsToOnPremisesInstances](#)」の「」を参照してください。

batch-get-application-revisions

次のコード例は、batch-get-application-revisions を使用する方法を示しています。

AWS CLI

アプリケーションリビジョンに関する情報を取得するには

次のbatch-get-application-revisions例では、GitHub リポジトリに保存されている指定されたリビジョンに関する情報を取得します。

```
aws deploy batch-get-application-revisions \  
  --application-name my-codedeploy-application \  
  --revisions "[{\\"githubLocation\\": {\\"commitId\\":  
  \\"fa85936EXAMPLEa31736c051f10d77297EXAMPLE\\",\\"repository\\": \\"my-github-token/my-  
  repository\\"},\\"revisionType\\": \\"GitHub\\"}]"
```

出力:

```
{  
  "revisions": [  
    {  
      "genericRevisionInfo": {  
        "description": "Application revision registered by Deployment ID: d-  
A1B2C3111",
```

```

        "lastUsedTime": 1556912355.884,
        "registerTime": 1556912355.884,
        "firstUsedTime": 1556912355.884,
        "deploymentGroups": []
    },
    "revisionLocation": {
        "revisionType": "GitHub",
        "gitHubLocation": {
            "commitId": "fa85936EXAMPLEa31736c051f10d77297EXAMPLE",
            "repository": "my-github-token/my-repository"
        }
    }
}
],
"applicationName": "my-codedeploy-application",
"errorMessage": ""
}

```

詳細については、「リファレンス[BatchGetApplicationRevisions](#)」の「」を参照してください。
AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetApplicationRevisions](#)」の「」を参照してください。

batch-get-applications

次の例は、batch-get-applications を使用する方法を説明しています。

AWS CLI

複数のアプリケーションに関する情報を取得するには

次のbatch-get-applications例は、ユーザーの AWS アカウントに関連付けられている複数のアプリケーションに関する情報を示しています。

```
aws deploy batch-get-applications --application-names WordPress_App MyOther_App
```

出力:

```

{
  "applicationsInfo": [
    {
      "applicationName": "WordPress_App",

```

```

        "applicationId": "d9dd6993-f171-44fa-a811-211e4EXAMPLE",
        "createTime": 1407878168.078,
        "linkedToGitHub": false
    },
    {
        "applicationName": "MyOther_App",
        "applicationId": "8ca57519-31da-42b2-9194-8bb16EXAMPLE",
        "createTime": 1407453571.63,
        "linkedToGitHub": false
    }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetApplications](#)」の「」を参照してください。

batch-get-deployment-groups

次の例は、batch-get-deployment-groups を使用する方法を説明しています。

AWS CLI

1 つ以上のデプロイグループに関する情報を取得するには

次のbatch-get-deployment-groups例では、指定された CodeDeploy アプリケーションに関連付けられている 2 つのデプロイグループに関する情報を取得します。

```

aws deploy batch-get-deployment-groups \
  --application-name my-codedeploy-application \
  --deployment-group-names ["my-deployment-group-1","my-deployment-group-2"]

```

出力:

```

{
  "deploymentGroupsInfo": [
    {
      "deploymentStyle": {
        "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",
        "deploymentType": "IN_PLACE"
      },
      "autoRollbackConfiguration": {
        "enabled": false
      }
    }
  ]
}

```

```
    },
    "onPremisesTagSet": {
      "onPremisesTagSetList": []
    },
    "serviceRoleArn": "arn:aws:iam::123456789012:role/
CodeDeployServiceRole",
    "lastAttemptedDeployment": {
      "endTime": 1556912366.415,
      "status": "Failed",
      "createTime": 1556912355.884,
      "deploymentId": "d-A1B2C3111"
    },
    "autoScalingGroups": [],
    "deploymentGroupName": "my-deployment-group-1",
    "ec2TagSet": {
      "ec2TagSetList": [
        [
          {
            "Type": "KEY_AND_VALUE",
            "Value": "my-EC2-instance",
            "Key": "Name"
          }
        ]
      ]
    },
    "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111example",
    "triggerConfigurations": [],
    "applicationName": "my-codedeploy-application",
    "computePlatform": "Server",
    "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
  },
  {
    "deploymentStyle": {
      "deploymentOption": "WITHOUT_TRAFFIC_CONTROL",
      "deploymentType": "IN_PLACE"
    },
    "autoRollbackConfiguration": {
      "enabled": false
    },
    "onPremisesTagSet": {
      "onPremisesTagSetList": []
    },
    "serviceRoleArn": "arn:aws:iam::123456789012:role/
CodeDeployServiceRole",
```

```

    "autoScalingGroups": [],
    "deploymentGroupName": "my-deployment-group-2",
    "ec2TagSet": {
      "ec2TagSetList": [
        [
          {
            "Type": "KEY_AND_VALUE",
            "Value": "my-EC2-instance",
            "Key": "Name"
          }
        ]
      ]
    },
    "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-2222example",
    "triggerConfigurations": [],
    "applicationName": "my-codedeploy-application",
    "computePlatform": "Server",
    "deploymentConfigName": "CodeDeployDefault.AllAtOnce"
  }
],
"errorMessage": ""
}

```

詳細については、「リファレンス[BatchGetDeploymentGroups](#)」の「」を参照してください。
AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetDeploymentGroups](#)」の「」を参照してください。

batch-get-deployment-targets

次の例は、batch-get-deployment-targets を使用する方法を説明しています。

AWS CLI

デプロイに関連付けられているターゲットを取得するには

次のbatch-get-deployment-targets例では、指定されたデプロイに関連付けられたターゲットの1つに関する情報を返します。

```

aws deploy batch-get-deployment-targets \
  --deployment-id "d-1A2B3C4D5" \

```

```
--target-ids "i-01a2b3c4d5e6f1111"
```

出力:

```
{
  "deploymentTargets": [
    {
      "deploymentTargetType": "InstanceTarget",
      "instanceTarget": {
        "lifecycleEvents": [
          {
            "startTime": 1556918592.162,
            "lifecycleEventName": "ApplicationStop",
            "status": "Succeeded",
            "endTime": 1556918592.247,
            "diagnostics": {
              "scriptName": "",
              "errorCode": "Success",
              "logTail": "",
              "message": "Succeeded"
            }
          },
          {
            "startTime": 1556918593.193,
            "lifecycleEventName": "DownloadBundle",
            "status": "Succeeded",
            "endTime": 1556918593.981,
            "diagnostics": {
              "scriptName": "",
              "errorCode": "Success",
              "logTail": "",
              "message": "Succeeded"
            }
          },
          {
            "startTime": 1556918594.805,
            "lifecycleEventName": "BeforeInstall",
            "status": "Succeeded",
            "endTime": 1556918681.807,
            "diagnostics": {
              "scriptName": "",
              "errorCode": "Success",
              "logTail": "",

```



```
        "message": "Succeeded"
      }
    ]
  ],
  "targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-01a2b3c4d5e6f1111",
  "deploymentId": "d-1A2B3C4D5",
  "lastUpdatedAt": 1556918687.504,
  "targetId": "i-01a2b3c4d5e6f1111",
  "status": "Succeeded"
}
]
}
```

詳細については、「リファレンス[BatchGetDeploymentTargets](#)」の「」を参照してください。

AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetDeploymentTargets](#)」の「」を参照してください。

batch-get-deployments

次の例は、batch-get-deployments を使用する方法を説明しています。

AWS CLI

複数のデプロイに関する情報を取得するには

次のbatch-get-deployments例では、ユーザーの AWS アカウントに関連付けられている複数のデプロイに関する情報を表示します。

```
aws deploy batch-get-deployments --deployment-ids d-A1B2C3111 d-A1B2C3222
```

出力:

```
{
  "deploymentsInfo": [
    {
      "applicationName": "WordPress_App",
      "status": "Failed",
      "deploymentOverview": {
        "Failed": 0,
```

```

        "InProgress": 0,
        "Skipped": 0,
        "Succeeded": 1,
        "Pending": 0
    },
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "creator": "user",
    "deploymentGroupName": "WordPress_DG",
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bundleType": "zip",
            "version": "uTecLusEXAMPLEFXtfUcyfV8bEXAMPLE",
            "bucket": "CodeDeployDemoBucket",
            "key": "WordPressApp.zip"
        }
    },
    "deploymentId": "d-A1B2C3111",
    "createTime": 1408480721.9,
    "completeTime": 1408480741.822
},
{
    "applicationName": "MyOther_App",
    "status": "Failed",
    "deploymentOverview": {
        "Failed": 1,
        "InProgress": 0,
        "Skipped": 0,
        "Succeeded": 0,
        "Pending": 0
    },
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "creator": "user",
    "errorInformation": {
        "message": "Deployment failed: Constraint default violated: No hosts
succeeded.",
        "code": "HEALTH_CONSTRAINTS"
    },
    "deploymentGroupName": "MyOther_DG",
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bundleType": "zip",
            "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",

```

```
        "bucket": "CodeDeployDemoBucket",
        "key": "MyOtherApp.zip"
      }
    },
    "deploymentId": "d-A1B2C3222",
    "createTime": 1409764576.589,
    "completeTime": 1409764596.101
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetDeployments](#)」の「」を参照してください。

batch-get-on-premises-instances

次のコード例は、batch-get-on-premises-instances を使用する方法を示しています。

AWS CLI

1 つ以上のオンプレミスインスタンスに関する情報を取得するには

次のbatch-get-on-premises-instances例では、2 つのオンプレミスインスタンスに関する情報を取得します。

```
aws deploy batch-get-on-premises-instances --instance-
names AssetTag12010298EX AssetTag23121309EX
```

出力:

```
{
  "instanceInfos": [
    {
      "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag12010298EX",
      "tags": [
        {
          "Value": "CodeDeployDemo-OnPrem",
          "Key": "Name"
        }
      ],
    }
  ],
}
```

```
    "instanceName": "AssetTag12010298EX",
    "registerTime": 1425579465.228,
    "instanceArn": "arn:aws:codedeploy:us-west-2:123456789012:instance/
AssetTag12010298EX_4IwLNI2Alh"
  },
  {
    "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/
AssetTag23121309EX",
    "tags": [
      {
        "Value": "CodeDeployDemo-OnPrem",
        "Key": "Name"
      }
    ],
    "instanceName": "AssetTag23121309EX",
    "registerTime": 1425595585.988,
    "instanceArn": "arn:aws:codedeploy:us-west-2:80398EXAMPLE:instance/
AssetTag23121309EX_PomUy64Was"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetOnPremisesInstances](#)」の「」を参照してください。

continue-deployment

次の例は、continue-deployment を使用する方法を説明しています。

AWS CLI

指定された待機時間が経過するのを待たずにトラフィックのルート変更を開始するには。

次のcontinue-deployment例では、元の環境のインスタンスから、代替環境のインスタンスへのトラフィックのシフトを開始する準備ができているインスタンスへのトラフィックの再ルーティングを開始します。

```
aws deploy continue-deployment \
  --deployment-id "d-A1B2C3111" \
  --deployment-wait-type "READY_WAIT"
```

このコマンドでは何も出力されません。

詳細については、「リファレンス[ContinueDeployment](#)」の「」を参照してください。AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス[ContinueDeployment](#)」の「」を参照してください。

create-application

次の例は、create-application を使用する方法を説明しています。

AWS CLI

アプリケーションを作成するには

次のcreate-application例では、アプリケーションを作成し、ユーザーの AWS アカウントと関連付けます。

```
aws deploy create-application --application-name MyOther_App
```

出力:

```
{
  "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateApplication](#)」の「」を参照してください。

create-deployment-config

次の例は、create-deployment-config を使用する方法を説明しています。

AWS CLI

カスタムデプロイ設定を作成するには

次のcreate-deployment-config例では、カスタムデプロイ設定を作成し、ユーザーの AWS アカウントに関連付けます。

```
aws deploy create-deployment-config \
```

```
--deployment-config-name ThreeQuartersHealthy \  
--minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

出力:

```
{  
  "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDeploymentConfig](#)」の「」を参照してください。

create-deployment-group

次のコード例は、create-deployment-group を使用する方法を示しています。

AWS CLI

デプロイグループを作成するには

次のcreate-deployment-group例では、デプロイグループを作成し、指定されたアプリケーションとユーザーの AWS アカウントと関連付けます。

```
aws deploy create-deployment-group \  
  --application-name WordPress_App \  
  --auto-scaling-groups CodeDeployDemo-ASG \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DG \  
  --ec2-tag-filters Key=Name,Value=CodeDeployDemo,Type=KEY_AND_VALUE \  
  --service-role-arn arn:aws:iam::123456789012:role/CodeDeployDemoRole
```

出力:

```
{  
  "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDeploymentGroup](#)」の「」を参照してください。

create-deployment

次の例は、create-deployment を使用する方法を説明しています。

AWS CLI

例 1: EC2/オンプレミスコンピューティングプラットフォームを使用して CodeDeploy デプロイを作成するには

次のcreate-deployment例では、デプロイを作成し、ユーザーの AWS アカウントに関連付けます。

```
aws deploy create-deployment \  
  --application-name WordPress_App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DG \  
  --description "My demo deployment" \  
  --s3-  
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke
```

出力:

```
{  
  "deploymentId": "d-A1B2C3111"  
}
```

例 2: Amazon ECSコンピューティングプラットフォームを使用して CodeDeploy デプロイを作成するには

次のcreate-deployment例では、次の 2 つのファイルを使用して Amazon ECSサービスをデプロイします。

create-deployment.json ファイルの内容。

```
{  
  "applicationName": "ecs-deployment",  
  "deploymentGroupName": "ecs-deployment-dg",  
  "revision": {  
    "revisionType": "S3",  
    "s3Location": {  
      "bucket": "ecs-deployment-bucket",  
      "key": "appspec.yaml",
```

```
        "bundleType": "YAML"
      }
    }
  }
```

このファイルは、次に という S3 バケット `appspect.yaml` から次のファイルを取得します `ecs-deployment-bucket`。

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:region:123456789012:task-definition/ecs-task-def:2"
        LoadBalancerInfo:
          ContainerName: "sample-app"
          ContainerPort: 80
          PlatformVersion: "LATEST"
```

コマンド:

```
aws deploy create-deployment \
  --cli-input-json file://create-deployment.json \
  --region us-east-1
```

出力:

```
{
  "deploymentId": "d-1234ABCDE"
}
```

詳細については、「リファレンス [CreateDeployment](#)」の「」を参照してください。AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDeployment](#)」の「」を参照してください。

delete-application

次の例は、`delete-application` を使用する方法を説明しています。

AWS CLI

アプリケーションを削除するには

次のdelete-application例では、ユーザーの AWS アカウントに関連付けられている指定されたアプリケーションを削除します。

```
aws deploy delete-application --application-name WordPress_App
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApplication](#)」の「」を参照してください。

delete-deployment-config

次の例は、delete-deployment-config を使用する方法を説明しています。

AWS CLI

デプロイ設定を削除するには

次のdelete-deployment-config例では、ユーザーの AWS アカウントに関連付けられているカスタムデプロイ設定を削除します。

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDeploymentConfig](#)」の「」を参照してください。

delete-deployment-group

次の例は、delete-deployment-group を使用する方法を説明しています。

AWS CLI

デプロイグループを削除するには

次のdelete-deployment-group例では、指定されたアプリケーションに関連付けられているデプロイグループを削除します。

```
aws deploy delete-deployment-group \  
  --application-name WordPress_App \  
  --deployment-group-name WordPress_DG
```

出力:

```
{  
  "hooksNotCleanedUp": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDeploymentGroup](#)」の「」を参照してください。

delete-git-hub-account-token

次のコード例は、delete-git-hub-account-token を使用する方法を示しています。

AWS CLI

GitHub アカウント接続を削除するには

次のdelete-git-hub-account-token例では、指定された GitHub アカウントの接続を削除します。

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

出力:

```
{  
  "tokenName": "my-github-account"  
}
```

詳細については、「リファレンス[DeleteGitHubAccountToken](#)」の「」を参照してください。

AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGitHubAccountToken](#)」の「」を参照してください。

deregister-on-premises-instance

次のコード例は、`deregister-on-premises-instance` を使用する方法を示しています。

AWS CLI

オンプレミスインスタンスの登録を解除するには

次の`deregister-on-premises-instance`例では、オンプレミスインスタンスを に登録解除しますが AWS CodeDeploy、インスタンスに関連付けられたIAMユーザーを削除したり、オンプレミスインスタンスタグの関連付け AWS CodeDeploy を解除したりすることはありません。また、インスタンスから AWS CodeDeploy エージェントをアンインストールしたり、インスタンスからオンプレミス設定ファイルを削除したりすることはありません。

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterOnPremisesInstance](#)」の「」を参照してください。

deregister

次の例は、`deregister` を使用する方法を説明しています。

AWS CLI

オンプレミスインスタンスの登録を解除するには

次の`deregister`例では、オンプレミスインスタンスを に登録解除します AWS CodeDeploy。インスタンスに関連付けられているIAMユーザーは削除されません。インスタンスからオンプレミスタグ AWS CodeDeploy の関連付けを解除します。エージェントをインスタンス AWS CodeDeploy からアンインストールしたり、インスタンスからオンプレミス設定ファイルを削除したりすることはありません。

```
aws deploy deregister \  
  --instance-name AssetTag12010298EX \  
  --no-delete-iam-user \  
  --region us-west-2
```

出力:

```
Retrieving on-premises instance information... DONE
IamUserArn: arn:aws:iam::80398EXAMPLE:user/AWS/CodeDeploy/AssetTag12010298EX
Tags: Key=Name,Value=CodeDeployDemo-OnPrem
Removing tags from the on-premises instance... DONE
Deregistering the on-premises instance... DONE
Run the following command on the on-premises instance to uninstall the codedeploy-
agent:
aws deploy uninstall
```

- API 詳細については、「コマンドリファレンス」の「[登録解除](#)」を参照してください。AWS CLI

get-application-revision

次のコード例は、get-application-revision を使用する方法を示しています。

AWS CLI

アプリケーションリビジョンに関する情報を取得するには

次のget-application-revision例では、指定されたアプリケーションに関連付けられているアプリケーションリビジョンに関する情報を表示します。

```
aws deploy get-application-revision \
  --application-name WordPress_App \
  --s3-
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfEXAMPLE8e768f9d77fEXAMPLE,ke
```

出力:

```
{
  "applicationName": "WordPress_App",
  "revisionInfo": {
    "description": "Application revision registered by Deployment ID: d-
A1B2C3111",
    "registerTime": 1411076520.009,
    "deploymentGroups": "WordPress_DG",
    "lastUsedTime": 1411076520.009,
    "firstUsedTime": 1411076520.009
  }
}
```

```
    },
    "revision": {
      "revisionType": "S3",
      "s3Location": {
        "bundleType": "zip",
        "eTag": "dd56cfdEXAMPLE8e768f9d77fEXAMPLE",
        "bucket": "CodeDeployDemoBucket",
        "key": "WordPressApp.zip"
      }
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApplicationRevision](#)」の「」を参照してください。

get-application

次の例は、get-application を使用する方法を説明しています。

AWS CLI

アプリケーションに関する情報を取得するには

次のget-application例は、ユーザーの AWS アカウントに関連付けられているアプリケーションに関する情報を表示します。

```
aws deploy get-application --application-name WordPress_App
```

出力:

```
{
  "application": {
    "applicationName": "WordPress_App",
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "createTime": 1407878168.078,
    "linkedToGitHub": false
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApplication](#)」の「」を参照してください。

get-deployment-config

次のコード例は、get-deployment-config を使用する方法を示しています。

AWS CLI

デプロイ設定に関する情報を取得するには

次のget-deployment-config例では、ユーザーの AWS アカウントに関連付けられているデプロイ設定に関する情報を表示します。

```
aws deploy get-deployment-config --deployment-config-name ThreeQuartersHealthy
```

出力:

```
{
  "deploymentConfigInfo": {
    "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "minimumHealthyHosts": {
      "type": "FLEET_PERCENT",
      "value": 75
    },
    "createTime": 1411081164.379,
    "deploymentConfigName": "ThreeQuartersHealthy"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeploymentConfig](#)」の「」を参照してください。

get-deployment-group

次のコード例は、get-deployment-group を使用する方法を示しています。

AWS CLI

デプロイグループに関する情報を表示するには

次のget-deployment-group例は、指定されたアプリケーションに関連付けられているデプロイグループに関する情報を表示します。

```
aws deploy get-deployment-group \
```

```
--application-name WordPress_App \  
--deployment-group-name WordPress_DG
```

出力:

```
{  
  "deploymentGroupInfo": {  
    "applicationName": "WordPress_App",  
    "autoScalingGroups": [  
      "CodeDeployDemo-ASG"  
    ],  
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",  
    "ec2TagFilters": [  
      {  
        "Type": "KEY_AND_VALUE",  
        "Value": "CodeDeployDemo",  
        "Key": "Name"  
      }  
    ],  
    "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "serviceRoleArn": "arn:aws:iam::123456789012:role/CodeDeployDemoRole",  
    "deploymentGroupName": "WordPress_DG"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeploymentGroup](#)」の「」を参照してください。

get-deployment-instance

次の例は、get-deployment-instance を使用する方法を説明しています。

AWS CLI

デプロイインスタンスに関する情報を取得するには

次のget-deployment-instance例では、指定されたデプロイに関連付けられているデプロイインスタンスに関する情報を表示します。

```
aws deploy get-deployment-instance --deployment-id d-QA4G4F9EX --instance-id i-902e9fEX
```

出力:

```
{
  "instanceSummary": {
    "instanceId": "arn:aws:ec2:us-east-1:80398EXAMPLE:instance/i-902e9fEX",
    "lifecycleEvents": [
      {
        "status": "Succeeded",
        "endTime": 1408480726.569,
        "startTime": 1408480726.437,
        "lifecycleEventName": "ApplicationStop"
      },
      {
        "status": "Succeeded",
        "endTime": 1408480728.016,
        "startTime": 1408480727.665,
        "lifecycleEventName": "DownloadBundle"
      },
      {
        "status": "Succeeded",
        "endTime": 1408480729.744,
        "startTime": 1408480729.125,
        "lifecycleEventName": "BeforeInstall"
      },
      {
        "status": "Succeeded",
        "endTime": 1408480730.979,
        "startTime": 1408480730.844,
        "lifecycleEventName": "Install"
      },
      {
        "status": "Failed",
        "endTime": 1408480732.603,
        "startTime": 1408480732.1,
        "lifecycleEventName": "AfterInstall"
      },
      {
        "status": "Skipped",
        "endTime": 1408480732.606,
        "lifecycleEventName": "ApplicationStart"
      },
      {
        "status": "Skipped",
        "endTime": 1408480732.606,
```



```

        "lifecycleEventName": "ValidateService"
      }
    ],
    "deploymentId": "d-QA4G4F9EX",
    "lastUpdatedAt": 1408480733.152,
    "status": "Failed"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDeploymentInstance](#)」の「」を参照してください。

get-deployment-target

次のコード例は、get-deployment-target を使用する方法を示しています。

AWS CLI

デプロイターゲットに関する情報を返すには

次のget-deployment-target例では、指定されたデプロイに関連付けられているデプロイターゲットに関する情報を返します。

```

aws deploy get-deployment-target \
  --deployment-id "d-A1B2C3111" \
  --target-id "i-a1b2c3d4e5f61111"

```

出力:

```

{
  "deploymentTarget": {
    "deploymentTargetType": "InstanceTarget",
    "instanceTarget": {
      "lastUpdatedAt": 1556918687.504,
      "targetId": "i-a1b2c3d4e5f61111",
      "targetArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-a1b2c3d4e5f61111",
      "status": "Succeeded",
      "lifecycleEvents": [
        {
          "status": "Succeeded",
          "diagnostics": {

```

```
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "ApplicationStop",
    "startTime": 1556918592.162,
    "endTime": 1556918592.247
},
{
    "status": "Succeeded",
    "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "DownloadBundle",
    "startTime": 1556918593.193,
    "endTime": 1556918593.981
},
{
    "status": "Succeeded",
    "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "BeforeInstall",
    "startTime": 1556918594.805,
    "endTime": 1556918681.807
},
{
    "status": "Succeeded",
    "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
    },
    "lifecycleEventName": "Install",
    "startTime": 1556918682.696,
    "endTime": 1556918683.005
}
```

```
    },
    {
      "status": "Succeeded",
      "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
      },
      "lifecycleEventName": "AfterInstall",
      "startTime": 1556918684.135,
      "endTime": 1556918684.216
    },
    {
      "status": "Succeeded",
      "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
      },
      "lifecycleEventName": "ApplicationStart",
      "startTime": 1556918685.211,
      "endTime": 1556918685.295
    },
    {
      "status": "Succeeded",
      "diagnostics": {
        "errorCode": "Success",
        "message": "Succeeded",
        "logTail": "",
        "scriptName": ""
      },
      "lifecycleEventName": "ValidateService",
      "startTime": 1556918686.65,
      "endTime": 1556918686.747
    }
  ],
  "deploymentId": "d-A1B2C3111"
}
}
```

詳細については、「リファレンス[GetDeploymentTarget](#)」の「」を参照してください。AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeploymentTarget](#)」の「」を参照してください。

get-deployment

次の例は、get-deployment を使用する方法を説明しています。

AWS CLI

デプロイに関する情報を取得するには

次のget-deployment例は、ユーザーの AWS アカウントに関連付けられているデプロイに関する情報を表示します。

```
aws deploy get-deployment --deployment-id d-A1B2C3123
```

出力:

```
{
  "deploymentInfo": {
    "applicationName": "WordPress_App",
    "status": "Succeeded",
    "deploymentOverview": {
      "Failed": 0,
      "InProgress": 0,
      "Skipped": 0,
      "Succeeded": 1,
      "Pending": 0
    },
    "deploymentConfigName": "CodeDeployDefault.OneAtATime",
    "creator": "user",
    "description": "My WordPress app deployment",
    "revision": {
      "revisionType": "S3",
      "s3Location": {
        "bundleType": "zip",
        "eTag": "\"dd56cfdEXAMPLE8e768f9d77fEXAMPLE\"",
        "bucket": "CodeDeployDemoBucket",
        "key": "WordPressApp.zip"
      }
    }
  }
}
```

```
    },
    "deploymentId": "d-A1B2C3123",
    "deploymentGroupName": "WordPress_DG",
    "createTime": 1409764576.589,
    "completeTime": 1409764596.101,
    "ignoreApplicationStopFailures": false
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeployment](#)」の「」を参照してください。

get-on-premises-instance

次の例は、get-on-premises-instance を使用する方法を説明しています。

AWS CLI

オンプレミスインスタンスに関する情報を取得するには

次のget-on-premises-instance例では、指定されたオンプレミスインスタンスに関する情報を取得します。

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

出力:

```
{
  "instanceInfo": {
    "iamUserArn": "arn:aws:iam::123456789012:user/AWS/CodeDeploy/AssetTag12010298EX",
    "tags": [
      {
        "Value": "CodeDeployDemo-OnPrem",
        "Key": "Name"
      }
    ],
    "instanceName": "AssetTag12010298EX",
    "registerTime": 1425579465.228,
    "instanceArn": "arn:aws:codedeploy:us-east-1:123456789012:instance/AssetTag12010298EX_4IwLNI2Alh"
  }
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOnPremisesInstance](#)」の「」を参照してください。

install

次のコード例は、install を使用する方法を示しています。

AWS CLI

オンプレミスインスタンスをインストールするには

次のinstall例では、オンプレミス設定ファイルをインスタンス上の指定された場所から、AWS CodeDeploy エージェントが検出する予定のインスタンス上の場所にコピーします。また、インスタンスに AWS CodeDeploy エージェントをインストールします。ユーザーを作成したりIAM、オンプレミスインスタンスを に登録したり AWS CodeDeploy、インスタンス AWS CodeDeploy の にオンプレミスインスタンスタグを関連付けたりすることはありません。

```
aws deploy install \  
  --override-config \  
  --config-file C:\temp\codedeploy.onpremises.yml \  
  --region us-west-2 \  
  --agent-installer s3://aws-codedeploy-us-west-2/latest/codedeploy-agent.msi
```

出力:

```
Creating the on-premises instance configuration file... DONE  
Installing the AWS CodeDeploy Agent... DONE
```

- API 詳細については、「コマンドリファレンス」の「[インストール](#)」を参照してください。

AWS CLI

list-application-revisions

次のコード例は、list-application-revisions を使用する方法を示しています。

AWS CLI

アプリケーションリビジョンに関する情報を取得するには

次のlist-application-revisions例では、指定されたアプリケーションに関連付けられているすべてのアプリケーションリビジョンに関する情報を表示します。

```
aws deploy list-application-revisions \  
  --application-name WordPress_App \  
  --s-3-bucket CodeDeployDemoBucket \  
  --deployed exclude \  
  --s-3-key-prefix WordPress_ \  
  --sort-by LastUsedTime \  
  --sort-order descending
```

出力:

```
{  
  "revisions": [  
    {  
      "revisionType": "S3",  
      "s3Location": {  
        "version": "uTecLusvCB_JqHFXtfUcyfV8bEXAMPLE",  
        "bucket": "CodeDeployDemoBucket",  
        "key": "WordPress_App.zip",  
        "bundleType": "zip"  
      }  
    },  
    {  
      "revisionType": "S3",  
      "s3Location": {  
        "version": "tMk.UxgDpMEVb7V187ZM6wVAWEXAMPLE",  
        "bucket": "CodeDeployDemoBucket",  
        "key": "WordPress_App_2-0.zip",  
        "bundleType": "zip"  
      }  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListApplicationRevisions](#)」の「」を参照してください。

list-applications

次の例は、list-applications を使用する方法を説明しています。

AWS CLI

アプリケーションに関する情報を取得するには

次のlist-applications例では、ユーザーの AWS アカウントに関連付けられているすべてのアプリケーションに関する情報を表示します。

```
aws deploy list-applications
```

出力:

```
{
  "applications": [
    "WordPress_App",
    "MyOther_App"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListApplications](#)」の「」を参照してください。

list-deployment-configs

次のコード例は、list-deployment-configs を使用する方法を示しています。

AWS CLI

デプロイ設定に関する情報を取得するには

次のlist-deployment-configs例は、ユーザーの AWS アカウントに関連付けられているすべてのデプロイ設定に関する情報を表示します。

```
aws deploy list-deployment-configs
```

出力:

```
{
  "deploymentConfigsList": [
    "ThreeQuartersHealthy",
    "CodeDeployDefault.AllAtOnce",
    "CodeDeployDefault.HalfAtATime",
  ]
}
```



```
        "CodeDeployDefault.OneAtATime"
    ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeploymentConfigs](#)」の「」を参照してください。

list-deployment-groups

次のコード例は、list-deployment-groups を使用する方法を示しています。

AWS CLI

デプロイグループに関する情報を取得するには

次のlist-deployment-groups例では、指定されたアプリケーションに関連付けられているすべてのデプロイグループに関する情報を表示します。

```
aws deploy list-deployment-groups --application-name WordPress_App
```

出力:

```
{
  "applicationName": "WordPress_App",
  "deploymentGroups": [
    "WordPress_DG",
    "WordPress_Beta_DG"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeploymentGroups](#)」の「」を参照してください。

list-deployment-instances

次のコード例は、list-deployment-instances を使用する方法を示しています。

AWS CLI

デプロイインスタンスに関する情報を取得するには

次のlist-deployment-instances例では、指定されたデプロイに関連付けられているすべてのデプロイインスタンスに関する情報を表示します。

```
aws deploy list-deployment-instances \  
  --deployment-id d-A1B2C3111 \  
  --instance-status-filter Succeeded
```

出力:

```
{  
  "instancesList": [  
    "i-EXAMPLE11",  
    "i-EXAMPLE22"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeploymentInstances](#)」の「」を参照してください。

list-deployment-targets

次のコード例は、list-deployment-targets を使用する方法を示しています。

AWS CLI

デプロイIDsに関連付けられているターゲットのリストを取得するには

次のlist-deployment-targets例では、「失敗」または「」のステータスを持つデプロイIDsに関連付けられたターゲットのリストを取得しますInProgress。

```
aws deploy list-deployment-targets \  
  --deployment-id "d-A1B2C3111" \  
  --target-filters "{\\"TargetStatus\\":[\\"Failed\\",\\"InProgress\\"]}"
```

出力:

```
{  
  "targetIds": [  
    "i-0f1558aaf90e5f1f9"  
  ]  
}
```

```
}
```

詳細については、「リファレンス[ListDeploymentTargets](#)」の「」を参照してください。AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeploymentTargets](#)」の「」を参照してください。

list-deployments

次のコード例は、list-deployments を使用する方法を示しています。

AWS CLI

デプロイに関する情報を取得するには

次のlist-deployments例では、指定されたアプリケーションとデプロイグループに関連付けられているすべてのデプロイに関する情報を表示します。

```
aws deploy list-deployments \  
  --application-name WordPress_App \  
  --create-time-range start=2014-08-19T00:00:00,end=2014-08-20T00:00:00 \  
  --deployment-group-name WordPress_DG \  
  --include-only-statuses Failed
```

出力:

```
{  
  "deployments": [  
    "d-EXAMPLE11",  
    "d-EXAMPLE22",  
    "d-EXAMPLE33"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeployments](#)」の「」を参照してください。

list-git-hub-account-token-names

次の例は、list-git-hub-account-token-names を使用する方法を説明しています。

AWS CLI

GitHub アカウントへのストアド接続の名を一覧表示するには

次の `list-git-hub-account-token-names` 例では、現在の AWS ユーザーの GitHub アカウントへのストアド接続の名を一覧表示します。

```
aws deploy list-git-hub-account-token-names
```

出力:

```
{
  "tokenNameList": [
    "my-first-token",
    "my-second-token",
    "my-third-token"
  ]
}
```

詳細については、「リファレンス [ListGitHubAccountTokenNames](#)」の「」を参照してください。

AWS CodeDeploy API

- API 詳細については、AWS CLI 「コマンドリファレンス [ListGitHubAccountTokenNames](#)」の「」を参照してください。

list-on-premises-instances

次のコード例は、`list-on-premises-instances` を使用する方法を示しています。

AWS CLI

1 つ以上のオンプレミスインスタンスに関する情報を取得するには

次の `list-on-premises-instances` 例では、に登録 AWS CodeDeploy され、インスタンス AWS CodeDeploy に関連付けられた指定されたオンプレミスインスタンスタグを持つインスタンスで使用可能なオンプレミスインスタンス名のリストを取得します。

```
aws deploy list-on-premises-instances \
  --registration-status Registered \
  --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

出力:

```
{
  "instanceNames": [
    "AssetTag12010298EX"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOnPremisesInstances](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには (アプリケーション)

次のlist-tags-for-resource例では、testApp という名前のアプリケーションに適用されるタグを一覧表示します CodeDeploy。

```
aws deploy list-tags-for-resource \
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp
```

出力:

```
{
  "Tags": [
    {
      "Key": "Type",
      "Value": "testType"
    },
    {
      "Key": "Name",
      "Value": "testName"
    }
  ]
}
```

詳細については、AWS CodeDeploy 「ユーザーガイド」の「[デプロイグループのインスタスのタグ付け CodeDeploy](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListTagsForResource](#)」の「」を参照してください。

push

次の例は、push を使用する方法を説明しています。

AWS CLI

AWS CodeDeploy 互換性のあるアプリケーションリビジョンを Amazon S3 にバンドルしてデプロイするには

次のpush例では、アプリケーションリビジョンをバンドルして Amazon S3 にデプロイし、アプリケーションリビジョンを指定されたアプリケーションに関連付けます。

```
aws deploy push \  
  --application-name WordPress_App \  
  --description "This is my deployment" \  
  --ignore-hidden-files \  
  --s3-location s3://CodeDeployDemoBucket/WordPressApp.zip \  
  --source /tmp/MyLocalDeploymentFolder/
```

出力では、create-deployment コマンドを使用して、アップロードされたアプリケーションリビジョンを使用するデプロイを作成する方法について説明します。

```
To deploy with this revision, run:  
aws deploy create-deployment --application-name WordPress_App  
  --deployment-config-name <deployment-config-name> --  
deployment-group-name <deployment-group-name> --s3-location  
  bucket=CodeDeployDemoBucket,key=WordPressApp.zip,bundleType=zip,eTag="cecc9b8EXAMPLE50a6e71"
```

- API 詳細については、AWS CLI 「コマンドリファレンスの [プッシュ](#)」を参照してください。

register-application-revision

次のコード例は、register-application-revision を使用する方法を示しています。

AWS CLI

既にアップロードされたアプリケーションリビジョンに関する情報を登録するには

次のregister-application-revision例では、Amazon S3 に保存されているアップロード済みのアプリケーションリビジョンに関する情報を に登録します AWS CodeDeploy。

```
aws deploy register-application-revision \  
  --application-name WordPress_App \  
  --description "Revised WordPress application" \  
  --s3-  
location bucket=CodeDeployDemoBucket,key=RevisedWordPressApp.zip,bundleType=zip,eTag=cecc9b8
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterApplicationRevision](#)」の「」を参照してください。

register-on-premises-instance

次の例は、register-on-premises-instance を使用する方法を説明しています。

AWS CLI

オンプレミスインスタンスを登録するには

次のregister-on-premises-instance例では、オンプレミスインスタンスを に登録します AWS CodeDeploy。指定されたIAMユーザーを作成したり、AWS CodeDeploy オンプレミスインスタンスタグに登録されたインスタンスに関連付けたりすることはありません。

```
aws deploy register-on-premises-instance \  
  --instance-name AssetTag12010298EX \  
  --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployDemoUser-OnPrem
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterOnPremisesInstance](#)」の「」を参照してください。

register

次の例は、register を使用する方法を説明しています。

AWS CLI

オンプレミスインスタンスを登録するには

次のregister例では、オンプレミスインスタンスを に登録し AWS CodeDeploy、 AWS CodeDeploy 指定されたオンプレミスインスタンスタグに登録されたインスタンスに関連付け、インスタンスにコピーできるオンプレミス設定ファイルを作成します。ユーザーを作成したり IAM、インスタンスに AWS CodeDeploy エージェントをインストールしたりしません。

```
aws deploy register \  
  --instance-name AssetTag12010298EX \  
  --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployUser-OnPrem \  
  --tags Key=Name, Value=CodeDeployDemo-OnPrem \  
  --region us-west-2
```

出力:

```
Registering the on-premises instance... DONE  
Adding tags to the on-premises instance... DONE  
Copy the on-premises configuration file named codedeploy.onpremises.yml to the on-  
premises instance, and run the following command on the on-premises instance to  
install and configure the AWS CodeDeploy Agent:  
aws deploy install --config-file codedeploy.onpremises.yml
```

- API 詳細については、「コマンドリファレンス」の「[登録](#)」を参照してください。AWS CLI

remove-tags-from-on-premises-instances

次の例は、remove-tags-from-on-premises-instances を使用する方法を説明しています。

AWS CLI

1 つ以上のオンプレミスインスタンスからタグを削除するには

次のremove-tags-from-on-premises-instances例では、 で指定されたオンプレミスタグとオンプレミスインスタンス AWS CodeDeploy の関連付けを解除します。 でオンプレミスインスタンスを登録解除したり AWS CodeDeploy、インスタンスから AWS CodeDeploy エージェントをアンインストールしたり、インスタンスからオンプレミス設定ファイルを削除したり、インスタンスに関連付けられているIAMユーザーを削除したりすることはありません。

```
aws deploy remove-tags-from-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name, Value=CodeDeployDemo-OnPrem
```


このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveTagsFromOnPremisesInstances](#)」の「」を参照してください。

stop-deployment

次のコード例は、stop-deployment を使用する方法を示しています。

AWS CLI

デプロイを停止するには

次のstop-deployment例では、ユーザーの AWS アカウントに関連付けられている進行中のデプロイを停止しようとします。

```
aws デプロイ停止 --deployment-id d-A1B2C3111
```

出力:

```
{
  "status": "Succeeded",
  "statusMessage": "No more commands will be scheduled for execution in the
deployment instances"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [StopDeployment](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを付けるには (アプリケーション)

次のtag-resource例では、キー Name と Type、および 値 testName とを持つ 2 つのタグ testType を testApp で という名前のアプリケーションに追加します CodeDeploy。

```
aws deploy tag-resource \
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp \
```

```
--tags Key=Name,Value=testName Key=Type,Value=testType
```

このコマンドが成功した場合、出力は生成されません。

詳細については、AWS CodeDeploy 「ユーザーガイド」の「[デプロイグループのインスタスのタグ付け CodeDeploy](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

uninstall

次のコード例は、uninstall を使用する方法を示しています。

AWS CLI

オンプレミスインスタンスをアンインストールするには

次のuninstall例では、オンプレミスインスタンスから AWS CodeDeploy エージェントをアンインストールし、インスタンスからオンプレミス設定ファイルを削除します。でインスタンスの登録を解除したり AWS CodeDeploy、インスタンス AWS CodeDeploy からオンプレミスインスタンスタグの関連付けを解除したり、インスタンスに関連付けられているIAMユーザーを削除したりすることはありません。

```
aws deploy uninstall
```

このコマンドでは何も出力されません。

- API 詳細については、「コマンドリファレンス」の「[アンインストール](#)」を参照してください。AWS CLI

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースからタグを削除するには (アプリケーション)

次のuntag-resource例では、testApp という名前のアプリケーションから、キーの名前とタイプを持つ 2 つのタグを削除します CodeDeploy。

```
aws deploy untag-resource \  
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp \  
  --tag-keys Name Type
```

このコマンドが成功した場合、出力は生成されません。

詳細については、AWS CodeDeploy 「ユーザーガイド」の「[デプロイグループのインスタスのタグ付け CodeDeploy](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-application

次の例は、update-application を使用する方法を説明しています。

AWS CLI

アプリケーションの詳細を変更するには

次のupdate-application例では、ユーザーの AWS アカウントに関連付けられているアプリケーションの名前を変更します。

```
aws deploy update-application \  
  --application-name WordPress_App \  
  --new-application-name My_WordPress_App
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateApplication](#)」の「」を参照してください。

update-deployment-group

次のコード例は、update-deployment-group を使用する方法を示しています。

AWS CLI

デプロイグループに関する情報を変更するには

次のupdate-deployment-group例では、指定されたアプリケーションに関連付けられているデプロイグループの設定を変更します。

```
aws deploy update-deployment-group \  
  --application-name WordPress_App \  
  --auto-scaling-groups My_CodeDeployDemo_ASG \  
  --current-deployment-group-name WordPress_DG \  
  --deployment-config-name CodeDeployDefault.AllAtOnce \  
  --ec2-tag-filters Key=Name,Type=KEY_AND_VALUE,Value=My_CodeDeployDemo \  
  --new-deployment-group-name My_WordPress_DepGroup \  
  --service-role-arn arn:aws:iam::80398EXAMPLE:role/CodeDeployDemo-2
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDeploymentGroup](#)」の「」を参照してください。

CodeGuru を使用したレビュー担当者の例 AWS CLI

次のコード例は、AWS Command Line Interface と CodeGuru Reviewer を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-repository

次のコード例は、associate-repository を使用する方法を示しています。

AWS CLI

例 1: Bitbucket リポジトリの関連付けを作成するには

次のassociate-repository例では、既存の Bitbucket リポジトリを使用してリポジトリの関連付けを作成します。

```
aws codeguru-reviewer associate-repository \  
  --repository 'Bitbucket={Owner=sample-owner, Name=mySampleRepo,  
ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

出力:

```
{  
  "RepositoryAssociation": {  
    "ProviderType": "Bitbucket",  
    "Name": "mySampleRepo",  
    "LastUpdatedTimeStamp": 1596216896.979,  
    "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "CreatedTimeStamp": 1596216896.979,  
    "ConnectionArn": "arn:aws:codestar-connections:us-  
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "State": "Associating",  
    "StateReason": "Pending Repository Association",  
    "AssociationArn": "arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "Owner": "sample-owner"  
  }  
}
```

詳細については、[「Amazon CodeGuru Reviewer ユーザーガイド」の「Amazon Reviewer で Bitbucket リポジトリの関連付けを作成する CodeGuru」](#)を参照してください。

例 2: Enterprise GitHub リポジトリの関連付けを作成するには

次の associate-repository 例では、既存の GitHub Enterprise リポジトリを使用してリポジトリの関連付けを作成します。

```
aws codeguru-reviewer associate-repository \  
  --repository 'GitHubEnterpriseServer={Owner=sample-owner, Name=mySampleRepo,  
ConnectionArn=arn:aws:codestar-connections:us-west-2:123456789012:connection/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 }'
```

出力:

```
{  
  "RepositoryAssociation": {  
    "ProviderType": "GitHubEnterpriseServer",
```

```

    "Name": "mySampleRepo",
    "LastUpdatedTimeStamp": 1596216896.979,
    "AssociationId": "association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "CreatedTimeStamp": 1596216896.979,
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "State": "Associating",
    "StateReason": "Pending Repository Association",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Owner": "sample-owner"
  }
}

```

詳細については、[「Amazon Codeguru CodeGuru Reviewer ユーザーガイド」の「Amazon Reviewer で GitHub エンタープライズサーバーリポジトリの関連付けを作成する」](#)を参照してください。

例 3: AWS CodeCommit リポジトリの関連付けを作成するには

次のassociate-repository例では、既存のリポジトリを使用して AWS CodeCommit リポジトリの関連付けを作成します。

```

aws codeguru-reviewer associate-repository \
  --repository CodeCommit={Name=mySampleRepo}

```

出力:

```

{
  "RepositoryAssociation": {
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "My-ecs-beta-repo",
    "LastUpdatedTimeStamp": 1595634764.029,
    "ProviderType": "CodeCommit",
    "CreatedTimeStamp": 1595634764.029,
    "Owner": "544120495673",
    "State": "Associating",
    "StateReason": "Pending Repository Association",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:544120495673:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}

```

詳細については、[「Amazon CodeGuru Reviewer ユーザーガイド」の「Amazon Reviewer で AWS CodeCommit リポジトリの関連付けを作成する」](#)を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateRepository](#)」の「」を参照してください。

create-code-review

次のコード例は、create-code-review を使用する方法を示しています。

AWS CLI

コードレビューを作成するには。

以下は、`my-code-review` という名前の AWS CodeCommit リポジトリの mainline ブランチにコードのレビュー create-code-review を作成します `my-repository-name`。

```
aws codeguru-reviewer create-code-review \  
  --name my-code-review \  
  --repository-association-arn arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --type '{"RepositoryAnalysis": {"RepositoryHead": {"BranchName": "mainline"}}}'
```

出力:

```
{  
  "CodeReview": {  
    "Name": "my-code-review",  
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-review:RepositoryAnalysis-my-code-review",  
    "RepositoryName": "my-repository-name",  
    "Owner": "123456789012",  
    "ProviderType": "CodeCommit",  
    "State": "Pending",  
    "StateReason": "CodeGuru Reviewer has received the request, and a code review is scheduled.",  
    "CreatedTimeStamp": 1618873489.195,  
    "LastUpdatedTimeStamp": 1618873489.195,  
    "Type": "RepositoryAnalysis",  
    "SourceCodeType": {  
      "RepositoryHead": {  
        "BranchName": "mainline"  
      }  
    }  
  }  
}
```

```
    }
  },
  "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
}
```

詳細については、[「Amazon CodeGuru Reviewer ユーザーガイド」の「Amazon Reviewer でコードレビューを作成する」](#)を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCodeReview](#)」の「」を参照してください。

describe-code-review

次の例は、describe-code-review を使用する方法を説明しています。

AWS CLI

コードレビューの詳細を一覧表示します。

以下に、「」という名前の AWS CodeCommit リポジトリの「メインライン」ブランチ内のコードのレビューに関する情報describe-code-reviewを示しますmy-repo-name。

```
aws codeguru-reviewer put-recommendation-feedback \
  --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefg12345678 \
  --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \
  --reactions ThumbsUp
```

出力

```
{
  "CodeReview": {
    "Name": "My-ecs-beta-repo-master-xs6di4kfd4j269dz",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222:code-
review:RepositoryAnalysis-my-repo-name",
    "RepositoryName": "My-ecs-beta-repo",
    "Owner": "123456789012",
```



```
    "ProviderType": "CodeCommit",
    "State": "Pending",
    "StateReason": "CodeGuru Reviewer is reviewing the source code.",
    "CreatedTimeStamp": 1618874226.226,
    "LastUpdatedTimeStamp": 1618874233.689,
    "Type": "RepositoryAnalysis",
    "SourceCodeType": {
      "RepositoryHead": {
        "BranchName": "mainline"
      }
    },
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}
```

詳細については、「Amazon Reviewer ユーザーガイド」の「[コードレビューの詳細を表示する](#)」を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCodeReview](#)」の「」を参照してください。

describe-recommendation-feedback

次のコード例は、describe-recommendation-feedback を使用する方法を示しています。

AWS CLI

レコメンデーションに関するフィードバックに関する情報を表示するには

以下は、レコメンデーションに関するフィードバックに関する情報describe-recommendation-feedbackを示しています。このレコメンデーションには 1 つのThumbsUp反応があります。

```
aws codeguru-reviewer describe-recommendation-feedback \
  --code-review-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefg12345678 \
  --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb
```

出力:

```
{
  "RecommendationFeedback": {
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefg12345678",
    "RecommendationId":
"3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
    "Reactions": [
      "ThumbsUp"
    ],
    "UserId": "aws-user-id",
    "CreatedTimeStamp": 1618877070.313,
    "LastUpdatedTimeStamp": 1618877948.881
  }
}
```

詳細については、「Amazon CodeGuru Reviewer ユーザーガイド」の「[レコメンデーションの表示とフィードバックの提供](#)」および「[ステップ 4: フィードバックの提供](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeRecommendationFeedback](#)」の「」を参照してください。

describe-repository-association

次の例は、describe-repository-association を使用する方法を説明しています。

AWS CLI

例 1: GitHub リポジトリの関連付けに関する情報を返すには

次のdescribe-repository-association例では、GitHub エンタープライズリポジトリを使用して Associated状態にあるリポジトリの関連付けに関する情報を返します。

```
aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{
  "RepositoryAssociation": {
```

```

    "AssociationId": "b822717e-0711-4e8a-bada-0e738289c75e",
    "Name": "mySampleRepo",
    "LastUpdatedTimeStamp": 1588102637.649,
    "ProviderType": "GitHub",
    "CreatedTimeStamp": 1588102615.636,
    "Owner": "sample-owner",
    "State": "Associated",
    "StateReason": "Pull Request Notification configuration successful",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}

```

詳細については、[「Amazon CodeGuru Reviewer ユーザーガイド」の「Amazon Reviewer で GitHub エンタープライズサーバーリポジトリの関連付けを作成する」](#)を参照してください。

CodeGuru

例 2: 失敗したリポジトリの関連付けに関する情報を返すには

次のdescribe-repository-association例では、GitHub エンタープライズリポジトリを使用して Failed状態にあるリポジトリの関連付けに関する情報を返します。

```

aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

出力:

```

{
  "RepositoryAssociation": {
    "ProviderType": "GitHubEnterpriseServer",
    "Name": "mySampleRepo",
    "LastUpdatedTimeStamp": 1596217036.892,
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "CreatedTimeStamp": 1596216896.979,
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "State": "Failed",
    "StateReason": "Failed, Please retry.",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "Owner": "sample-owner"
  }
}

```

```
}
}
```

詳細については、[「Amazon CodeGuru Reviewer ユーザーガイド」の「Amazon Reviewer で GitHub エンタープライズサーバーリポジトリの関連付けを作成する」](#)を参照してください。

CodeGuru

例 3: リポジトリの関連付けの関連付け解除に関する情報を返すには

次のdescribe-repository-association例では、GitHub エンタープライズリポジトリを使用して Disassociating状態にあるリポジトリの関連付けに関する情報を返します。

```
aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{
  "RepositoryAssociation": {
    "ProviderType": "GitHubEnterpriseServer",
    "Name": "mySampleRepo",
    "LastUpdatedTimeStamp": 1596217036.892,
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "CreatedTimeStamp": 1596216896.979,
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "State": "Disassociating",
    "StateReason": "Source code access removal in progress",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "Owner": "sample-owner"
  }
}
```

詳細については、[「Amazon CodeGuru Reviewer ユーザーガイド」の「Amazon Reviewer で GitHub エンタープライズサーバーリポジトリの関連付けを作成する」](#)を参照してください。

CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRepositoryAssociation](#)」の「」を参照してください。

disassociate-repository

次のコード例は、disassociate-repository を使用する方法を示しています。

AWS CLI

リポジトリの関連付けを解除するには

次に、リポジトリを使用している AWS CodeCommit リポジトリの関連付けの関連付けdisassociate-repositoryを解除します。

```
aws codeguru-reviewer disassociate-repository \  
  --association-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "RepositoryAssociation": {  
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "AssociationArn": "arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Name": "my-repository",  
    "Owner": "123456789012",  
    "ProviderType": "CodeCommit",  
    "State": "Disassociating",  
    "LastUpdatedTimeStamp": 1618939174.759,  
    "CreatedTimeStamp": 1595636947.096  
  },  
  "Tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

詳細については、[「Amazon CodeGuru Reviewer ユーザーガイド」の「Reviewer のリポジトリの関連付けを解除する」](#)を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateRepository](#)」の「」を参照してください。

list-code-reviews

次の例は、list-code-reviews を使用する方法を説明しています。

AWS CLI

過去 90 日間に AWS アカウントで作成されたコードレビューを一覧表示するには。

次のlist-code-reviews例では、プルリクエストを使用して過去 90 日間に作成されたコードレビューを一覧表示します。

```
aws codeguru-reviewer list-code-reviews \  
  --type PullRequest
```

出力:

```
{  
  "CodeReviewSummaries": [  
    {  
      "LastUpdatedTimeStamp": 1588897288.054,  
      "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ProviderType": "GitHub",  
      "PullRequestId": "5",  
      "MetricsSummary": {  
        "MeteredLinesOfCodeCount": 24,  
        "FindingsCount": 1  
      },  
      "CreatedTimeStamp": 1588897068.512,  
      "State": "Completed",  
      "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-  
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Owner": "sample-owner",  
      "RepositoryName": "sample-repository-name",  
      "Type": "PullRequest"  
    },  
    {  
      "LastUpdatedTimeStamp": 1588869793.263,  
      "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "ProviderType": "GitHub",  
      "PullRequestId": "4",  
      "MetricsSummary": {  
        "MeteredLinesOfCodeCount": 29,  
        "FindingsCount": 0  
      }  
    }  
  ]  
}
```

```
    },
    "CreatedTimeStamp": 1588869575.949,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
    "LastUpdatedTimeStamp": 1588870511.211,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "ProviderType": "GitHub",
    "PullRequestId": "4",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 2,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588870292.425,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
    "LastUpdatedTimeStamp": 1588118522.452,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
    "ProviderType": "GitHub",
    "PullRequestId": "3",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 29,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588118301.131,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
```

```
    "LastUpdatedTimeStamp": 1588112205.207,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
    "ProviderType": "GitHub",
    "PullRequestId": "2",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 25,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588111987.443,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  },
  {
    "LastUpdatedTimeStamp": 1588104489.981,
    "Name": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
    "ProviderType": "GitHub",
    "PullRequestId": "1",
    "MetricsSummary": {
      "MeteredLinesOfCodeCount": 25,
      "FindingsCount": 0
    },
    "CreatedTimeStamp": 1588104270.223,
    "State": "Completed",
    "CodeReviewArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:code-
review:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
    "Owner": "sample-owner",
    "RepositoryName": "sample-repository-name",
    "Type": "PullRequest"
  }
]
}
```

詳細については、「Amazon CodeGuru Reviewer ユーザーガイド」の「[すべてのコードレビューを表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCodeReviews](#)」の「」を参照してください。

list-recommendation-feedback

次のコード例は、list-recommendation-feedback を使用する方法を示しています。

AWS CLI

関連付けられたリポジトリのレコメンデーションに関するカスタマーレコメンデーションのフィードバックを一覧表示するには

以下は、コードレビューに関するすべてのレコメンデーションに関するお客様からのフィードバックlist-recommendation-feedbackの一覧です。このコードレビューには、顧客からのフィードバックの1つであるThumbsUp「」があります。

```
aws codeguru-reviewer list-recommendation-feedback \  
  --code-review-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-  
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678
```

出力:

```
{  
  "RecommendationFeedbackSummaries": [  
    {  
      "RecommendationId":  
"3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",  
      "Reactions": [  
        "ThumbsUp"  
      ],  
      "UserId": "aws-user-id"  
    }  
  ]  
}
```

詳細については、「Amazon CodeGuru Reviewer ユーザーガイド」の「[ステップ 4: フィードバックの提供](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRecommendationFeedback](#)」の「」を参照してください。

list-recommendations

次のコード例は、list-recommendations を使用する方法を示しています。

AWS CLI

完了したコードレビューのレコメンデーションを一覧表示するには

次のlist-recommendations例では、コンリテッドコードレビューの推奨事項を一覧表示します。このコードレビューには1つのレコメンデーションがあります。

```
aws codeguru-reviewer list-recommendations \  
  --code-review-arn arn:aws:codeguru-reviewer:us-west-2:544120495673:code-  
  review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "RecommendationSummaries": [  
    {  
      "Description": "\n\nProblem  \n You are using a `ConcurrentHashMap`,  
but your usage of `containsKey()` and `get()` may not be thread-safe at lines: **63  
and **64**. In between the check and the `get()` another thread can remove the key  
and the `get()` will return `null`. The remove that can remove the key is at line:  
**59**.\n\nFix  \n Consider calling `get()`, checking instead of your current  
check if the returned object is `null`, and then using that object only, without  
calling `get()` again.\n\nMore info  \n [View an example on GitHub](https://  
github.com/apache/hadoop/blob/f16cf877e565084c66bc63605659b157c4394dc8/hadoop-tools/  
hadoop-aws/src/main/java/org/apache/hadoop/fs/s3a/s3guard/S3Guard.java#L302-L304)  
(external link).",  
      "RecommendationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "StartLine": 63,  
      "EndLine": 64,  
      "FilePath": "src/main/java/com/company/sample/application/  
CreateOrderThread.java"  
    }  
  ]  
}
```

詳細については、「Amazon CodeGuru Reviewer ユーザーガイド」の「[ステップ 4: フィードバックの提供](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRecommendations](#)」の「」を参照してください。

list-repository-associations

次のコード例は、list-repository-associations を使用する方法を示しています。

AWS CLI

AWS アカウントのリポジトリの関連付けを一覧表示するには

次のlist-repository-associations例では、アカウントのリポジトリ関連付けサマリーオブジェクトのリストを返します。返されたリストは、ProviderType、NameState、およびでフィルタリングできますOwner。

```
aws codeguru-reviewer list-repository-associations
```

出力:

```
{
  "RepositoryAssociationSummaries": [
    {
      "LastUpdatedTimeStamp": 1595886609.616,
      "Name": "test",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Owner": "sample-owner",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ProviderType": "Bitbucket"
    },
    {
      "LastUpdatedTimeStamp": 1595636969.035,
      "Name": "CodeDeploy-CodePipeline-ECS-Tutorial",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "Owner": "123456789012",
      "State": "Associated",
      "AssociationArn": "arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "ProviderType": "CodeCommit"
    },
    {
      "LastUpdatedTimeStamp": 1595634785.983,
      "Name": "My-ecs-beta-repo",
      "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
```

```
    "Owner": "123456789012",
    "State": "Associated",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "ProviderType": "CodeCommit"
  },
  {
    "LastUpdatedTimeStamp": 1590712811.77,
    "Name": "MyTestCodeCommit",
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
    "Owner": "123456789012",
    "State": "Associated",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
    "ProviderType": "CodeCommit"
  },
  {
    "LastUpdatedTimeStamp": 1588102637.649,
    "Name": "aws-codeguru-profiler-sample-application",
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
    "Owner": "sample-owner",
    "State": "Associated",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE55555",
    "ProviderType": "GitHub"
  },
  {
    "LastUpdatedTimeStamp": 1588028233.995,
    "Name": "codeguru-profiler-demo-app",
    "AssociationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
    "Owner": "sample-owner",
    "State": "Associated",
    "AssociationArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
    "ProviderType": "GitHub"
  }
]
}
```

詳細については、「Amazon [CodeGuru Reviewer ユーザーガイド](#)」の「[Reviewerですべてのリポジトリの関連付けを表示する](#)」を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRepositoryAssociations](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

関連付けられたリポジトリのタグを一覧表示するには

以下に、関連付けられたリポジトリのタグlist-tags-for-resourceを示します。この関連付けられたリポジトリには 2 つのタグがあります。

```
aws codeguru-reviewer list-tags-for-resource \
  --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{
  "Tags": {
    "Status": "Secret",
    "Team": "Saanvi"
  }
}
```

詳細については、「Amazon [CodeGuru Reviewer ユーザーガイド](#)」の「[Reviewer 関連リポジトリ \(AWS CLI\) のタグを表示する](#)」を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-recommendation-feedback

次の例は、put-recommendation-feedback を使用する方法を説明しています。

AWS CLI

コードレビューにレコメンデーションを追加するには

以下は、コードレビューに関するThumbsUp推奨事項put-recommendation-feedbackです。

```
aws codeguru-reviewer put-recommendation-feedback \
```

```
--code-review-arn \arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678 \  
--recommendation-id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \  
--reactions ThumbsUp
```

このコマンドでは何も出力されません。

詳細については、「Amazon CodeGuru Reviewer ユーザーガイド」の「[ステップ 4: フィードバックの提供](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRecommendationFeedback](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

関連付けられたリポジトリにタグを追加するには

次に、関連付けられたリポジトリに 2 つのタグ tag-resource を追加します。

```
aws codeguru-reviewer tag-resource \  
--resource-arn arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
--tags Status=Secret,Team=Saarvi
```

このコマンドでは何も出力されません。

詳細については、「Amazon [CodeGuru Reviewer ユーザーガイド](#)」の「[Reviewer 関連リポジトリ \(AWS CLI\) にタグを追加](#)」および [CodeGuru 「Reviewer 関連リポジトリ \(AWS CLI\) のタグを追加または更新](#)」を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

関連付けられたリポジトリのタグを解除するには

以下は `untag-resource`、キーが「Secret」と「Team」の2つのタグを関連するリポジトリから削除します。

```
aws codeguru-reviewer untag-resource \  
  --resource-arn arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tag-keys Status Team
```

このコマンドでは何も出力されません。

詳細については、「Amazon [CodeGuru Reviewer ユーザーガイド](#)」の「[Reviewer 関連リポジトリ \(AWS CLI\) からタグを削除する](#)」を参照してください。 CodeGuru

- API 詳細については、AWS CLI 「[コマンドリファレンス UntagResource](#)」の「」を参照してください。

CodePipeline を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CodePipeline。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

acknowledge-job

次のコード例は、`acknowledge-job` を使用する方法を示しています。

AWS CLI

指定されたジョブに関する情報を取得するには

この例では、指定されたジョブが存在する場合、そのジョブのステータスなど、指定されたジョブに関する情報を返します。これは、ジョブワーカーとカスタムアクションにのみ使用されます。nonce の値とジョブ ID を決定するには、aws コードパイプライン を使用します poll-for-jobs。

コマンド:

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

出力:

```
{  
  "status": "InProgress"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AcknowledgeJob](#)」の「」を参照してください。

create-custom-action-type

次のコード例は、create-custom-action-type を使用する方法を示しています。

AWS CLI

カスタムアクションを作成するには

この例では、カスタムアクションの構造を含む、既に作成されたJSONファイル (ここでは MyCustomAction.json という名前) AWS CodePipeline を使用するためのカスタムアクションを作成します。ファイルの構造など、カスタムアクションを作成するための要件の詳細については、AWS CodePipeline 「ユーザーガイド」を参照してください。

```
aws codepipeline create-custom-action-type --cli-input-json file://  
MyCustomAction.json
```

JSON ファイルの内容MyCustomAction.json :


```
{
  "category": "Build",
  "provider": "MyJenkinsProviderName",
  "version": "1",
  "settings": {
    "entityUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/",
    "executionUrlTemplate": "https://192.0.2.4/job/{Config:ProjectName}/
lastSuccessfulBuild/{ExternalExecutionId}/"
  },
  "configurationProperties": [
    {
      "name": "MyJenkinsExampleBuildProject",
      "required": true,
      "key": true,
      "secret": false,
      "queryable": false,
      "description": "The name of the build project must be provided when this
action is added to the pipeline.",
      "type": "String"
    }
  ],
  "inputArtifactDetails": {
    "maximumCount": 1,
    "minimumCount": 0
  },
  "outputArtifactDetails": {
    "maximumCount": 1,
    "minimumCount": 0
  }
}
```

このコマンドは、カスタムアクションの構造を返します。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomActionType](#)」の「」を参照してください。

create-pipeline

次の例は、create-pipeline を使用する方法を説明しています。

AWS CLI

パイプラインを作成するには

この例では、パイプラインの構造を含む、既に作成されたJSONファイル (ここでは `MySecondPipeline.json` という名前) AWS CodePipeline を使用して、パイプラインを作成します。ファイルの構造など、パイプライン作成の要件の詳細については、AWS CodePipeline 「ユーザーガイド」を参照してください。

コマンド:

```
aws codepipeline create-pipeline --cli-input-json file://MySecondPipeline.json
```

JSON ファイルサンプルの内容 :

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "configuration": {
              "S3Bucket": "awscodepipeline-demo-bucket",
              "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
            },
            "runOrder": 1
          }
        ]
      },
      {
        "name": "Beta",
        "actions": [
```

```
{
  "inputArtifacts": [
    {
      "name": "MyApp"
    }
  ],
  "name": "CodePipelineDemoFleet",
  "actionTypeId": {
    "category": "Deploy",
    "owner": "AWS",
    "version": "1",
    "provider": "CodeDeploy"
  },
  "outputArtifacts": [],
  "configuration": {
    "ApplicationName": "CodePipelineDemoApplication",
    "DeploymentGroupName": "CodePipelineDemoFleet"
  },
  "runOrder": 1
}
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "codepipeline-us-east-1-11EXAMPLE11"
},
"name": "MySecondPipeline",
"version": 1
}
}
```

出力:

This command returns the structure of the pipeline.

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePipeline](#)」の「」を参照してください。

delete-custom-action-type

次の例は、delete-custom-action-type を使用する方法を説明しています。

AWS CLI

カスタムアクションを削除するには

この例では、削除するアクションのアクションタイプ、プロバイダー名、バージョン番号を含む、既に作成されたJSONファイル (ここでは `DeleteMyCustomAction.json` という名前) AWS CodePipeline を使用して、 のカスタムアクションを削除します。 `list-action-types` コマンドを使用して、カテゴリ、バージョン、プロバイダーの正しい値を表示します。

コマンド:

```
aws codepipeline delete-custom-action-type --cli-input-json file://  
DeleteMyCustomAction.json
```

JSON ファイルサンプルの内容 :

```
{  
  "category": "Build",  
  "version": "1",  
  "provider": "MyJenkinsProviderName"  
}
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCustomActionType](#)」の「」を参照してください。

delete-pipeline

次のコード例は、`delete-pipeline` を使用する方法を示しています。

AWS CLI

パイプラインを削除するには

この例では、`MySecondPipeline` から という名前のパイプラインを削除します AWS CodePipeline。 `list-pipelines` コマンドを使用して、AWS アカウントに関連付けられたパイプラインのリストを表示します。

コマンド:

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePipeline](#)」の「」を参照してください。

delete-webhook

次のコード例は、delete-webhook を使用する方法を示しています。

AWS CLI

ウェブフックを削除するには

次のdelete-webhook例では、GitHub バージョン 1 のソースアクションのウェブフックを削除します。deregister-webhook-with-third-party コマンドを使用して、ウェブフックを削除する前に登録を解除する必要があります。

```
aws codepipeline delete-webhook \  
  --name my-webhook
```

このコマンドでは何も出力されません。

詳細については、AWS CodePipeline 「ユーザーガイド」の[GitHub 「ソースのウェブフックを削除する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteWebhook](#)」の「」を参照してください。

deregister-webhook-with-third-party

次のコード例は、deregister-webhook-with-third-party を使用する方法を示しています。

AWS CLI

ウェブフックの登録を解除するには

次のderegister-webhook-with-third-party例では、GitHub バージョン 1 ソースアクションのウェブフックを削除します。ウェブフックを削除する前に、その登録を解除する必要があります。

```
aws codepipeline deregister-webhook-with-third-party \  
  --webhook-name my-webhook
```

このコマンドでは何も出力されません。

詳細については、AWS CodePipeline 「ユーザーガイド」の [GitHub 「ソースのウェブフックを削除する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterWebhookWithThirdParty](#)」の「」を参照してください。

disable-stage-transition

次のコード例は、disable-stage-transition を使用する方法を示しています。

AWS CLI

パイプライン内のステージへの移行を無効にするには

この例では、の MyFirstPipeline パイプラインのベータステージへの移行を無効にします AWS CodePipeline。

コマンド:

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-  
name Beta --transition-type Inbound
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableStageTransition](#)」の「」を参照してください。

enable-stage-transition

次の例は、enable-stage-transition を使用する方法を説明しています。

AWS CLI

パイプライン内のステージへの移行を有効にするには

この例では、の MyFirstPipeline パイプラインのベータステージへの移行を有効にします AWS CodePipeline。

コマンド:

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta --transition-type Inbound
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableStageTransition](#)」の「」を参照してください。

get-job-details

次の例は、get-job-details を使用する方法を説明しています。

AWS CLI

ジョブの詳細を取得するには

この例では、ID が f4f4ff82-2d11- で表されるジョブに関する詳細を返しますEXAMPLE。このコマンドはカスタムアクションにのみ使用されます。このコマンドが呼び出されると、は、カスタムアクションで必要な場合、パイプラインのアーティファクトを保存するために使用される Amazon S3 バケットの一時的な認証情報 AWS CodePipeline を返します。このコマンドは、アクションに定義されているシークレット値がある場合も返します。

コマンド:

```
aws codepipeline get-job-details --job-id f4f4ff82-2d11-EXAMPLE
```

出力:

```
{  
  "jobDetails": {
```

```

"accountId": "111111111111",
"data": {
  "actionConfiguration": {
    "__type": "ActionConfiguration",
    "configuration": {
      "ProjectName": "MyJenkinsExampleTestProject"
    }
  },
  "actionTypeId": {
    "__type": "ActionTypeId",
    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "artifactCredentials": {
    "__type": "AWSSessionCredentials",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "sessionToken":
      "fICCQD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwd
      +a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/
      f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
      MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
      +auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs
  },
  "inputArtifacts": [
    {
      "__type": "Artifact",
      "location": {
        "s3Location": {
          "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
          "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
        },
        "type": "S3"
      },
      "name": "MyAppBuild"
    }
  ],
  "outputArtifacts": [],
  "pipelineContext": {
    "__type": "PipelineContext",
    "action": {
      "name": "MyJenkinsTest-Action"
    }
  }
}

```



```
    },
    "pipelineName": "MySecondPipeline",
    "stage": {
      "name": "Testing"
    }
  }
},
"id": "f4f4ff82-2d11-EXAMPLE"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetJobDetails](#)」の「」を参照してください。

get-pipeline-state

次の例は、get-pipeline-state を使用する方法を説明しています。

AWS CLI

パイプラインの状態に関する情報を取得するには

この例では、 という名前のパイプラインの最新の状態を返します MyFirstPipeline。

コマンド:

```
aws codepipeline get-pipeline-state --name MyFirstPipeline
```

出力:

```
{
  "created": 1446137312.204,
  "pipelineName": "MyFirstPipeline",
  "pipelineVersion": 1,
  "stageStates": [
    {
      "actionStates": [
        {
          "actionName": "Source",
          "entityUrl": "https://console.aws.amazon.com/s3/home?#",
          "latestExecution": {
            "lastStatusChange": 1446137358.328,
```

```
        "status": "Succeeded"
      }
    ],
    "stageName": "Source"
  },
  {
    "actionStates": [
      {
        "actionName": "CodePipelineDemoFleet",
        "entityUrl": "https://console.aws.amazon.com/codedeploy/home?#/applications/CodePipelineDemoApplication/deployment-groups/CodePipelineDemoFleet",
        "latestExecution": {
          "externalExecutionId": "d-EXAMPLE",
          "externalExecutionUrl": "https://console.aws.amazon.com/codedeploy/home?#/deployments/d-EXAMPLE",
          "lastStatusChange": 1446137493.131,
          "status": "Succeeded",
          "summary": "Deployment Succeeded"
        }
      }
    ],
    "inboundTransitionState": {
      "enabled": true
    },
    "stageName": "Beta"
  }
],
"updated": 1446137312.204
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPipelineState](#)」の「」を参照してください。

get-pipeline

次のコード例は、get-pipeline を使用する方法を示しています。

AWS CLI

パイプラインの構造を表示するには

この例では、 という名前のパイプラインの構造を返します MyFirstPipeline。

コマンド:

```
aws codepipeline get-pipeline --name MyFirstPipeline
```

出力:

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "configuration": {
              "S3Bucket": "awscodepipeline-demo-bucket",
              "S3ObjectKey": "aws-codepipeline-s3-aws-
codedeploy_linux.zip"
            },
            "runOrder": 1
          }
        ]
      },
      {
        "name": "Beta",
        "actions": [
          {
            "inputArtifacts": [
              {
                "name": "MyApp"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
        }
      ],
      "name": "CodePipelineDemoFleet",
      "actionTypeId": {
        "category": "Deploy",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeDeploy"
      },
      "outputArtifacts": [],
      "configuration": {
        "ApplicationName": "CodePipelineDemoApplication",
        "DeploymentGroupName": "CodePipelineDemoFleet"
      },
      "runOrder": 1
    }
  ]
}
},
"artifactStore": {
  "type": "S3",
  "location": "codepipeline-us-east-1-11EXAMPLE11"
},
"name": "MyFirstPipeline",
"version": 1
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPipeline](#)」の「」を参照してください。

list-action-executions

次のコード例は、list-action-executions を使用する方法を示しています。

AWS CLI

アクション実行を一覧表示するには

次のlist-action-executions例では、アクション実行 ID、入力アーティファクト、出力アーティファクト、実行結果、ステータスなど、パイプラインのアクション実行の詳細を表示します。

```
aws codepipeline list-action-executions \  
  --pipeline-name myPipeline
```

出力:

```
{  
  "actionExecutionDetails": [  
    {  
      "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",  
      "actionExecutionId": "EXAMPLE4-2ee8-4853-bd6a-111111158148",  
      "pipelineVersion": 12,  
      "stageName": "Deploy",  
      "actionName": "Deploy",  
      "startTime": 1598572628.6,  
      "lastUpdateTime": 1598572661.255,  
      "status": "Succeeded",  
      "input": {  
        "actionTypeId": {  
          "category": "Deploy",  
          "owner": "AWS",  
          "provider": "CodeDeploy",  
          "version": "1"  
        },  
        "configuration": {  
          "ApplicationName": "my-application",  
          "DeploymentGroupName": "my-deployment-group"  
        },  
        "resolvedConfiguration": {  
          "ApplicationName": "my-application",  
          "DeploymentGroupName": "my-deployment-group"  
        },  
        "region": "us-east-1",  
        "inputArtifacts": [  
          {  
            "name": "SourceArtifact",  
            "s3location": {  
              "bucket": "artifact-bucket",  
              "key": "myPipeline/SourceArti/key"  
            }  
          }  
        ],  
        "namespace": "DeployVariables"  
      },  
    ],  
  ],  
}
```

```

    "output": {
      "outputArtifacts": [],
      "executionResult": {
        "externalExecutionId": "d-EXAMPLEE5",
        "externalExecutionSummary": "Deployment Succeeded",
        "externalExecutionUrl": "https://myaddress.com"
      },
      "outputVariables": {}
    }
  },
  {
    "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
    "actionExecutionId": "EXAMPLE5-abb4-4192-9031-11111113a7b0",
    "pipelineVersion": 12,
    "stageName": "Source",
    "actionName": "Source",
    "startTime": 1598572624.387,
    "lastUpdateTime": 1598572628.16,
    "status": "Succeeded",
    "input": {
      "actionTypeId": {
        "category": "Source",
        "owner": "AWS",
        "provider": "CodeCommit",
        "version": "1"
      },
      "configuration": {
        "BranchName": "production",
        "PollForSourceChanges": "false",
        "RepositoryName": "my-repo"
      },
      "resolvedConfiguration": {
        "BranchName": "production",
        "PollForSourceChanges": "false",
        "RepositoryName": "my-repo"
      },
      "region": "us-east-1",
      "inputArtifacts": [],
      "namespace": "SourceVariables"
    },
    "output": {
      "outputArtifacts": [
        {
          "name": "SourceArtifact",

```

```
        "s3location": {
            "bucket": "my-bucket",
            "key": "myPipeline/SourceArti/key"
        }
    },
    "executionResult": {
        "externalExecutionId":
"11111111ad99dcd35914c00b7fbea13995EXAMPLE",
        "externalExecutionSummary": "Edited template.yml",
        "externalExecutionUrl": "https://myaddress.com"
    },
    "outputVariables": {
        "AuthorDate": "2020-05-08T17:45:43Z",
        "BranchName": "production",
        "CommitId": "EXAMPLEad99dcd35914c00b7fbea139951111111",
        "CommitMessage": "Edited template.yml",
        "CommitterDate": "2020-05-08T17:45:43Z",
        "RepositoryName": "my-repo"
    }
},
. . . .
```

詳細については、AWS CodePipeline 「ユーザーガイド」の [「アクション実行の表示 \(CLI\)」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListActionExecutions](#)」の「」を参照してください。

list-action-types

次のコード例は、list-action-types を使用する方法を示しています。

AWS CLI

使用可能なアクションタイプを表示するには

単独で使用するコマンドは list-action-types、AWS アカウントで使用できるすべてのアクションの構造を返します。この例では、-- action-owner-filter オプションを使用してカスタムアクションのみを返します。

コマンド:

```
aws codepipeline list-action-types --action-owner-filter Custom
```

出力:

```
{
  "actionTypes": [
    {
      "inputArtifactDetails": {
        "maximumCount": 5,
        "minimumCount": 0
      },
      "actionConfigurationProperties": [
        {
          "secret": false,
          "required": true,
          "name": "MyJenkinsExampleBuildProject",
          "key": true,
          "queryable": true
        }
      ],
      "outputArtifactDetails": {
        "maximumCount": 5,
        "minimumCount": 0
      },
      "id": {
        "category": "Build",
        "owner": "Custom",
        "version": "1",
        "provider": "MyJenkinsProviderName"
      },
      "settings": {
        "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
        "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
      }
    },
    {
      "inputArtifactDetails": {
        "maximumCount": 5,
        "minimumCount": 0
      },
      "actionConfigurationProperties": [
        {
```



```
        "secret": false,
        "required": true,
        "name": "MyJenkinsExampleTestProject",
        "key": true,
        "queryable": true
      }
    ],
    "outputArtifactDetails": {
      "maximumCount": 5,
      "minimumCount": 0
    },
    "id": {
      "category": "Test",
      "owner": "Custom",
      "version": "1",
      "provider": "MyJenkinsProviderName"
    },
    "settings": {
      "entityUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}",
      "executionUrlTemplate": "http://192.0.2.4/job/{Config:ProjectName}/
{ExternalExecutionId}"
    }
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListActionTypes](#)」の「」を参照してください。

list-pipeline-executions

次の例は、list-pipeline-executions を使用する方法を説明しています。

AWS CLI

パイプライン実行履歴を表示するには

次のlist-pipeline-executions例は、AWS アカウントのパイプラインのパイプライン実行履歴を示しています。

```
aws codepipeline list-pipeline-executions \  
  --pipeline-name MyPipeline
```

出力:

```
{
  "pipelineExecutionSummaries": [
    {
      "lastUpdateTime": 1496380678.648,
      "pipelineExecutionId": "7cf7f7cb-3137-539g-j458-d7eu3EXAMPLE",
      "startTime": 1496380258.243,
      "status": "Succeeded"
    },
    {
      "lastUpdateTime": 1496591045.634,
      "pipelineExecutionId": "3137f7cb-8d494hj4-039j-d84l-d7eu3EXAMPLE",
      "startTime": 1496590401.222,
      "status": "Succeeded"
    },
    {
      "lastUpdateTime": 1496946071.6456,
      "pipelineExecutionId": "4992f7jf-7cf7-913k-k334-d7eu3EXAMPLE",
      "startTime": 1496945471.5645,
      "status": "Succeeded"
    }
  ]
}
```

詳細については、AWS CodePipeline 「ユーザーガイド」の「[実行履歴の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPipelineExecutions](#)」の「」を参照してください。

list-pipelines

次の例は、list-pipelines を使用する方法を説明しています。

AWS CLI

パイプラインのリストを表示するには

この例では、ユーザーの AWS アカウントに関連付けられているすべての AWS CodePipeline パイプラインを一覧表示します。

コマンド:

```
aws codepipeline list-pipelines
```

出力:

```
{
  "pipelines": [
    {
      "updated": 1439504274.641,
      "version": 1,
      "name": "MyFirstPipeline",
      "created": 1439504274.641
    },
    {
      "updated": 1436461837.992,
      "version": 2,
      "name": "MySecondPipeline",
      "created": 1436460801.381
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPipelines](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

タグを一覧表示するには

次のlist-tags-for-resource例では、指定されたパイプラインリソースにアタッチされたすべてのタグのリストを取得します。

```
aws codepipeline list-tags-for-resource \
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline
```

出力:

```
{
```

```
"tags": {
  "Project": "ProjectA",
  "IscontainerBased": "true"
}
}
```

詳細については、AWS CodePipeline 「ユーザーガイド」の「[パイプライン \(CLI\) のタグを表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-webhooks

次のコード例は、list-webhooks を使用する方法を示しています。

AWS CLI

ウェブフックを一覧表示するには

次のlist-webhooks例では、指定されたパイプラインリソースにアタッチされたすべてのタグのリストを取得します。

```
aws codepipeline list-webhooks \
  --endpoint-url "https://codepipeline.eu-central-1.amazonaws.com" \
  --region "eu-central-1"
```

出力:

```
{
  "webhooks": [
    {
      "url": "https://webhooks.domain.com/
trigger1111111111EXAMPLE1111111111111111111": {
        "authenticationConfiguration": {
          "SecretToken": "Secret"
        },
        "name": "my-webhook",
        "authentication": "GITHUB_HMAC",
        "targetPipeline": "my-Pipeline",
        "targetAction": "Source",
        "filters": [
```

```
        {
            "jsonPath": "$.ref",
            "matchEquals": "refs/heads/{Branch}"
        }
    ],
    "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-
webhook"
}
]
```

詳細については、[「ユーザーガイド」の「アカウントのウェブフックを一覧表示する」](#)を参照してください。AWS CodePipeline

- API 詳細については、AWS CLI 「コマンドリファレンス[ListWebhooks](#)」の「」を参照してください。

poll-for-jobs

次のコード例は、poll-for-jobs を使用する方法を示しています。

AWS CLI

使用可能なジョブを表示するには

この例では、ジョブワーカーが実行するジョブに関する情報を返します。この例では、定義済みJSONファイル (MyActionTypeInfo.json) を使用して、ジョブワーカーがジョブを処理するアクションタイプに関する情報を提供します。このコマンドはカスタムアクションにのみ使用されます。このコマンドが呼び出されると、はパイプラインのアーティファクトの保存に使用される Amazon S3 バケットの一時的な認証情報 AWS CodePipeline を返します。このコマンドは、アクションに定義されているシークレット値がある場合も返します。

コマンド:

```
aws codepipeline poll-for-jobs --cli-input-json file://MyActionTypeInfo.json
```

JSON ファイルサンプルの内容:

```
{
  "actionTypeId": {
```

```

    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "maxBatchSize": 5,
  "queryParam": {
    "ProjectName": "MyJenkinsTestProject"
  }
}

```

出力:

```

{
  "jobs": [
    {
      "accountId": "111111111111",
      "data": {
        "actionConfiguration": {
          "__type": "ActionConfiguration",
          "configuration": {
            "ProjectName": "MyJenkinsExampleTestProject"
          }
        },
        "actionTypeId": {
          "__type": "ActionTypeId",
          "category": "Test",
          "owner": "Custom",
          "provider": "MyJenkinsProviderName",
          "version": "1"
        },
        "artifactCredentials": {
          "__type": "AWSSessionCredentials",
          "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
          "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFfiCYEXAMPLEKEY",
          "sessionToken":
            "fICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwd
            +a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/
            f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
            MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
            +auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs
          },
        "inputArtifacts": [

```

```
{
  "__type": "Artifact",
  "location": {
    "s3Location": {
      "bucketName": "codepipeline-us-east-1-11EXAMPLE11",
      "objectKey": "MySecondPipeline/MyAppBuild/EXAMPLE"
    },
    "type": "S3"
  },
  "name": "MyAppBuild"
},
],
"outputArtifacts": [],
"pipelineContext": {
  "__type": "PipelineContext",
  "action": {
    "name": "MyJenkinsTest-Action"
  },
  "pipelineName": "MySecondPipeline",
  "stage": {
    "name": "Testing"
  }
},
"id": "ef66c259-64f9-EXAMPLE",
"nonce": "3"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PollForJobs](#)」の「」を参照してください。

put-webhook

次のコード例は、put-webhook を使用する方法を示しています。

AWS CLI

ウェブフックを作成するには

次のput-webhook例では、GitHub バージョン 1 ソースアクションのウェブフックを作成します。ウェブフックを作成したら、register-webhook-with-thirdパーティコマンドを使用してウェブフックを登録する必要があります。

```
aws codepipeline put-webhook \  
  --cli-input-json file://webhook_json.json \  
  --region "eu-central-1"
```

webhook_json.json の内容:

```
{  
  "webhook": {  
    "name": "my-webhook",  
    "targetPipeline": "pipeline_name",  
    "targetAction": "source_action_name",  
    "filters": [  
      {  
        "jsonPath": "$.ref",  
        "matchEquals": "refs/heads/{Branch}"  
      }  
    ],  
    "authentication": "GITHUB_HMAC",  
    "authenticationConfiguration": {  
      "SecretToken": "secret"  
    }  
  }  
}
```

出力:

```
{  
  "webhook": {  
    "url": "https://webhooks.domain.com/  
trigger1111111111EXAMPLE1111111111111111111",  
    "definition": {  
      "authenticationConfiguration": {  
        "SecretToken": "secret"  
      },  
      "name": "my-webhook",  
      "authentication": "GITHUB_HMAC",  
      "targetPipeline": "pipeline_name",  
      "targetAction": "Source",  
    }  
  }  
}
```



```
    "filters": [
      {
        "jsonPath": "$.ref",
        "matchEquals": "refs/heads/{Branch}"
      }
    ],
    "arn": "arn:aws:codepipeline:eu-central-1:123456789012:webhook:my-webhook"
  },
  "tags": [
    {
      "key": "Project",
      "value": "ProjectA"
    }
  ]
}
```

詳細については、AWS CodePipeline 「ユーザーガイド」の [GitHub 「ソースのウェブフックを作成する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutWebhook](#)」の「」を参照してください。

retry-stage-execution

次の例は、retry-stage-execution を使用する方法を説明しています。

AWS CLI

失敗したアクションを再試行するには

次のretry-stage-execution例では、失敗したアクションを持つステージを再試行します。

```
aws codepipeline retry-stage-execution \
  --pipeline-name MyPipeline \
  --stage-name Deploy \
  --pipeline-execution-id b59babff-5f34-EXAMPLE \
  --retry-mode FAILED_ACTIONS
```

出力:

```
{
```

```
"pipelineExecutionId": "b59babff-5f34-EXAMPLE"  
}
```

詳細については、AWS CodePipeline ユーザーガイドの「[失敗したアクションを再試行する \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RetryStageExecution](#)」の「」を参照してください。

start-pipeline-execution

次のコード例は、start-pipeline-execution を使用する方法を示しています。

AWS CLI

パイプラインを介して最新のリリースを実行するには

この例では、MyFirstPipeline「」という名前のパイプラインを介してパイプラインのソースステージに存在する最新のリリースを実行します。

コマンド:

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

出力:

```
{  
  "pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartPipelineExecution](#)」の「」を参照してください。

stop-pipeline-execution

次の例は、stop-pipeline-execution を使用する方法を説明しています。

AWS CLI

パイプラインの実行を停止するには

次のstop-pipeline-execution例では、デフォルトで進行中のアクションが完了するまで待機し、パイプラインを停止します。実行がすでに [停止] 状態になっている場合、[Stop and wait (停止して待機)] を選択することはできません。実行がすでに [停止] 状態になっている場合、[Stop and abandon (停止して中止)] を選択することはできません。

```
aws codepipeline stop-pipeline-execution \  
  --pipeline-name MyFirstPipeline \  
  --pipeline-execution-id d-EXAMPLE \  
  --reason "Stopping pipeline after the build action is done"
```

このコマンドは出力なしを返します。

詳細については、AWS CodePipeline 「ユーザーガイド」の「[パイプライン実行を停止する \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopPipelineExecution](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、提供されたタグのセットをパイプラインに関連付けます。このコマンドを使用して、タグを追加または編集します。

```
aws codepipeline tag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tags key=Project,value=ProjectA key=IscontainerBased,value=true
```

このコマンドでは何も出力されません。

詳細については、AWS CodePipeline 「ユーザーガイド」の「[パイプラインにタグを追加する \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、`untag-resource` を使用する方法を説明しています。

AWS CLI

接続リソースから AWS タグを削除するには

次の`untag-resource`例では、指定されたリソースからタグを削除します。

```
aws codepipeline untag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tag-keys Project IscontainerBased
```

このコマンドでは何も出力されません。

詳細については、AWS CodePipeline 「ユーザーガイド」の「[パイプライン \(CLI\) からタグを削除する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-pipeline

次のコード例は、`update-pipeline` を使用する方法を示しています。

AWS CLI

パイプラインの構造を更新するには

この例では、`--cli-input-json` argument で `update-pipeline` コマンドを使用します。この例では、事前定義されたJSONファイル (`MyFirstPipeline.json`) を使用してパイプラインの構造を更新します。はJSON、ファイルに含まれるパイプライン名 AWS CodePipeline を認識し、パイプライン構造内の変更されたフィールドからの変更を適用してパイプラインを更新します。

事前定義されたJSONファイルを作成するときは、次のガイドラインを使用します。

`get-pipeline` コマンドを使用して取得したパイプライン構造を使用する場合は、JSONファイルのパイプライン構造からメタデータセクションを削除する必要があります (「メタデータ」:{ } 行、「作成済み、」パイプライン ARN、「更新済み」フィールド)。パイプライン名は変更できません。

コマンド:

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

サンプルJSONファイルの内容:

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "MyApp"
              }
            ],
            "configuration": {
              "S3Bucket": "awscodepipeline-demo-bucket2",
              "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
            },
            "runOrder": 1
          }
        ]
      },
      {
        "name": "Beta",
        "actions": [
          {
            "inputArtifacts": [
              {
                "name": "MyApp"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "name": "CodePipelineDemoFleet",
    "actionTypeId": {
      "category": "Deploy",
      "owner": "AWS",
      "version": "1",
      "provider": "CodeDeploy"
    },
    "outputArtifacts": [],
    "configuration": {
      "ApplicationName": "CodePipelineDemoApplication",
      "DeploymentGroupName": "CodePipelineDemoFleet"
    },
    "runOrder": 1
  }
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "codepipeline-us-east-1-11EXAMPLE11"
},
"name": "MyFirstPipeline",
"version": 1
}
}

```

出力:

```

{
  "pipeline": {
    "artifactStore": {
      "location": "codepipeline-us-east-1-11EXAMPLE11",
      "type": "S3"
    },
    "name": "MyFirstPipeline",
    "roleArn": "arn:aws:iam::111111111111:role/AWS-CodePipeline-Service",
    "stages": [
      {
        "actions": [
          {
            "actionTypeId": {
              "__type": "ActionTypeId",

```

```
    "category": "Source",
    "owner": "AWS",
    "provider": "S3",
    "version": "1"
  },
  "configuration": {
    "S3Bucket": "awscodepipeline-demo-bucket2",
    "S3ObjectKey": "aws-codepipeline-s3-aws-codedeploy_linux.zip"
  },
  "inputArtifacts": [],
  "name": "Source",
  "outputArtifacts": [
    {
      "name": "MyApp"
    }
  ],
  "runOrder": 1
}
],
"name": "Source"
},
{
  "actions": [
    {
      "actionTypeId": {
        "__type": "ActionTypeId",
        "category": "Deploy",
        "owner": "AWS",
        "provider": "CodeDeploy",
        "version": "1"
      },
      "configuration": {
        "ApplicationName": "CodePipelineDemoApplication",
        "DeploymentGroupName": "CodePipelineDemoFleet"
      },
      "inputArtifacts": [
        {
          "name": "MyApp"
        }
      ],
      "name": "CodePipelineDemoFleet",
      "outputArtifacts": [],
      "runOrder": 1
    }
  ]
}
```

```
    ],  
    "name": "Beta"  
  }  
],  
"version": 3  
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePipeline](#)」の「」を参照してください。

AWS CodeStar を使用した通知の例 AWS CLI

次のコード例は、AWS CodeStar 通知 AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-notification-rule

次のコード例は、create-notification-rule を使用する方法を示しています。

AWS CLI

通知ルールを作成するには

次のcreate-notification-rule例では、という名前のJSONファイルを使用して、指定された AWS アカウントMyDemoRepoで という名前のリポジトリMyNotificationRuleに という名前の通知ルールrule.jsonを作成します。ブランチとタグが作成されると、FULL詳細タイプの通知が指定されたターゲット Amazon SNSトピックに送信されます。


```
aws codestar-notifications create-notification-rule \  
--cli-input-json file://rule.json
```

rule.json の内容:

```
{  
  "Name": "MyNotificationRule",  
  "EventTypeIds": [  
    "codecommit-repository-branches-and-tags-created"  
  ],  
  "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",  
  "Targets": [  
    {  
      "TargetType": "SNS",  
      "TargetAddress": "arn:aws:sns:us-  
east-1:123456789012:MyNotificationTopic"  
    }  
  ],  
  "Status": "ENABLED",  
  "DetailType": "FULL"  
}
```

出力:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

詳細については、[「デベロッパーツールコンソールユーザーガイド」の「通知ルールを作成するAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNotificationRule](#)」の「」を参照してください。

delete-notification-rule

次の例は、delete-notification-rule を使用する方法を説明しています。

AWS CLI

通知ルールを削除するには

次のdelete-notification-rule例では、指定された通知ルールを削除します。

```
aws codestar-notifications delete-notification-rule \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE
```

出力:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

詳細については、[「デベロッパーツールコンソールユーザーガイド」の「通知ルールの削除AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNotificationRule](#)」の「」を参照してください。

delete-target

次の例は、delete-target を使用する方法を説明しています。

AWS CLI

通知ルールターゲットを削除するには

次のdelete-target例では、指定したターゲットをターゲットとして使用するように設定されたすべての通知ルールから削除し、ターゲットを削除します。

```
aws codestar-notifications delete-target \  
  --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic \  
  --force-unsubscribe-all
```

このコマンドでは何も出力されません。

詳細については、[「デベロッパーツールコンソールユーザーガイド」の「通知ルールターゲットの削除AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTarget](#)」の「」を参照してください。

describe-notification-rule

次のコード例は、describe-notification-rule を使用する方法を示しています。

AWS CLI

通知ルールの詳細を取得するには

次のdescribe-notification-rule例では、指定された通知ルールの詳細を取得します。

```
aws codestar-notifications describe-notification-rule \  
  --arn arn:aws:codestar-notifications:us-west-2:123456789012:notificationrule/  
dc82df7a-EXAMPLE
```

出力:

```
{  
  "LastModifiedTimestamp": 1569199844.857,  
  "EventTypes": [  
    {  
      "ServiceName": "CodeCommit",  
      "EventTypeName": "Branches and tags: Created",  
      "ResourceType": "Repository",  
      "EventTypeId": "codecommit-repository-branches-and-tags-created"  
    }  
  ],  
  "Status": "ENABLED",  
  "DetailType": "FULL",  
  "Resource": "arn:aws:codecommit:us-west-2:123456789012:MyDemoRepo",  
  "Arn": "arn:aws:codestar-notifications:us-west-w:123456789012:notificationrule/  
dc82df7a-EXAMPLE",  
  "Targets": [  
    {  
      "TargetStatus": "ACTIVE",  
      "TargetAddress": "arn:aws:sns:us-  
west-2:123456789012:MyNotificationTopic",  
      "TargetType": "SNS"  
    }  
  ],  
  "Name": "MyNotificationRule",  
  "CreatedTimestamp": 1569199844.857,  
  "CreatedBy": "arn:aws:iam::123456789012:user/Mary_Major"  
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の「[通知ルールの表示](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNotificationRule](#)」の「」を参照してください。

list-event-types

次のコード例は、list-event-types を使用する方法を示しています。

AWS CLI

通知ルールのイベントタイプのリストを取得するには

次のlist-event-types例では、CodeDeploy アプリケーションで使用できるすべての通知イベントタイプのフィルタリングされたリストを取得します。代わりにフィルターを使用しない場合、コマンドはすべてのリソースタイプに対するすべての通知イベントタイプを返します。

```
aws codestar-notifications list-event-types \  
  --filters Name=SERVICE_NAME,Value=CodeDeploy
```

出力:

```
{  
  "EventTypes": [  
    {  
      "EventTypeId": "codedeploy-application-deployment-succeeded",  
      "ServiceName": "CodeDeploy",  
      "EventTypeName": "Deployment: Succeeded",  
      "ResourceType": "Application"  
    },  
    {  
      "EventTypeId": "codedeploy-application-deployment-failed",  
      "ServiceName": "CodeDeploy",  
      "EventTypeName": "Deployment: Failed",  
      "ResourceType": "Application"  
    },  
    {  
      "EventTypeId": "codedeploy-application-deployment-started",  
      "ServiceName": "CodeDeploy",  
      "EventTypeName": "Deployment: Started",  
      "ResourceType": "Application"  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

詳細については、[「デベロッパーツールコンソールユーザーガイド」の「通知ルールの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEventTypes](#)」の「」を参照してください。

list-notification-rules

次のコード例は、list-notification-rules を使用する方法を示しています。

AWS CLI

通知ルールのリストを取得するには

次のlist-notification-rules例では、指定された AWS リージョン内のすべての通知ルールのリストを取得します。

```
aws codestar-notifications list-notification-rules --region us-east-1
```

出力:

```
{  
  "NotificationRules": [  
    {  
      "Id": "dc82df7a-EXAMPLE",  
      "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"  
    },  
    {  
      "Id": "8d1f0983-EXAMPLE",  
      "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/8d1f0983-EXAMPLE"  
    }  
  ]  
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の[「通知ルールの表示」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListNotificationRules](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

通知ルールにアタッチされたタグのリストを取得するには

次のlist-tags-for-resource例では、指定された通知ルールにアタッチされたすべてのタグのリストを取得します。この例では、通知ルールには現在タグが関連付けられていません。

```
aws codestar-notifications list-tags-for-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE
```

出力:

```
{  
  "Tags": {}  
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の「[通知ルールの作成](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-targets

次の例は、list-targets を使用する方法を説明しています。

AWS CLI

通知ルールターゲットのリストを取得するには

次のlist-targets例では、指定された AWS リージョン内のすべての通知ルールターゲットのリストを取得します。

```
aws codestar-notifications list-targets \  
  --region us-east-1
```

出力:

```
{  
  "Targets": [  
    {  
      "TargetAddress": "arn:aws:sns:us-  
east-1:123456789012:MySNSTopicForNotificationRules",  
      "TargetType": "SNS",  
      "TargetStatus": "ACTIVE"  
    },  
    {  
      "TargetAddress": "arn:aws:sns:us-  
east-1:123456789012:MySNSTopicForNotificationsAboutMyDemoRepo",  
      "TargetType": "SNS",  
      "TargetStatus": "ACTIVE"  
    }  
  ]  
}
```

詳細については、AWS「[デベロッパーツールコンソールユーザーガイド](#)」の「[通知ルールターゲットの表示](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスListTargets](#)」の「」を参照してください。

subscribe

次の例は、subscribe を使用する方法を説明しています。

AWS CLI

通知ルールにターゲットを追加するには

次のsubscribe例では、指定された通知ルールのターゲットとして Amazon SNS トピックを追加します。

```
aws codestar-notifications subscribe \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE \  
  --target-arn arn:aws:sns:us-east-1:123456789012:MySNSTopicForNotificationsAboutMyDemoRepo
```

```
--target TargetType=SNS,TargetAddress=arn:aws:sns:us-east-1:123456789012:MyNotificationTopic
```

出力:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

詳細については、AWS「デベロッパーツールコンソールユーザーガイド」の[「通知ルールのターゲットとしての Amazon SNS トピックの追加または削除」](#)を参照してください。

- API 詳細については、「コマンドリファレンス」の[「サブスクライブ」](#)を参照してください。
AWS CLI

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

通知ルールにタグを追加するには

次のtag-resource例では、のキー名Teamと の値が指定された通知ルールLi_Juanにタグを追加します。

```
aws codestar-notifications tag-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE \  
  --tags Team=Li_Juan
```

出力:

```
{  
  "Tags": {  
    "Team": "Li_Juan"  
  }  
}
```


詳細については、[「デベロッパーツールコンソールユーザーガイド」の「通知ルールの作成AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

unsubscribe

次の例は、unsubscribe を使用する方法を説明しています。

AWS CLI

通知ルールからターゲットを削除するには

次のunsubscribe例では、指定された通知ルールから Amazon SNSトピックをターゲットとして削除します。

```
aws codestar-notifications unsubscribe \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE \  
  --target TargetType=SNS,TargetAddress=arn:aws:sns:us-  
east-1:123456789012:MyNotificationTopic
```

出力:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
  "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"  
}
```

詳細については、AWS 「デベロッパーツールコンソールユーザーガイド」の[「通知ルールのターゲットとしての Amazon SNSトピックの追加または削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス」の[「サブスクリプション解除」](#)を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

通知ルールからタグを削除するには

次の`untag-resource`例では、指定された通知ルール`Team`からキー名のタグを削除します。

```
aws codestar-notifications untag-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE \  
  --tag-keys Team
```

このコマンドでは何も出力されません。

詳細については、[「デベロッパーツールコンソールユーザーガイド」の「通知ルールの編集AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-notification-rule

次の例は、`update-notification-rule` を使用する方法を説明しています。

AWS CLI

通知ルールを更新するには

次の`update-notification-rule`例では、`update.json`という名前のJSONファイル`123456789012`を使用して、AWS アカウント`MyNotificationRule`で `update.json`という名前の通知ルールを更新します。

```
aws codestar-notifications update-notification-rule \  
  --cli-input-json file://update.json
```

`update.json` の内容:

```
{  
  "Name": "MyUpdatedNotificationRule",  
  "EventTypeIds": [  
    "codecommit-repository-branches-and-tags-created"  ]  
}
```

```
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
      {
        "TargetType": "SNS",
        "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
      }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
  }
}
```

出力:

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

詳細については、[「デベロッパーツールコンソールユーザーガイド」の「通知ルールの編集AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateNotificationRule](#)」の「」を参照してください。

CodeConnections を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています CodeConnections。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-connection

次の例は、create-connection を使用する方法を説明しています。

AWS CLI

接続を作成するには

次のcreate-connection例は、サードパーティーリポジトリへの接続を作成する方法を示しています。この例では、サードパーティープロバイダーが Bitbucket である接続を作成します。

または を通じて AWS CLI作成された接続 AWS CloudFormation は、デフォルトで保留中のステータスです。CLI または の接続を作成したら AWS CloudFormation、コンソールを使用して接続を編集し、ステータスを使用可能にします。

```
aws codestar-connections create-connection \  
  --provider-type Bitbucket \  
  --connection-name MyConnection
```

出力:

```
{  
  "ConnectionArn": "arn:aws:codestar-connections:us-  
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"  
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の「[接続を作成する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConnection](#)」の「」を参照してください。

create-host

次のコード例は、create-host を使用する方法を示しています。

AWS CLI

ホストを作成するには

次のcreate-host例は、サードパーティープロバイダーがインストールされているインフラストラクチャのエンドポイントを表すホストを作成する方法を示しています。この例では、サードパーティーがインストールしたプロバイダーが GitHub Enterprise Server であるホストを作成します。

を通じて AWS CLI作成されたホストは、デフォルトで保留中のステータスです。でホストを作成したらCLI、コンソールまたは CLI を使用して、ステータスを使用可能にするようにホストを設定します。

```
aws codestar-connections create-host \  
  --name MyHost \  
  --provider-type GitHubEnterpriseServer \  
  --provider-endpoint "https://my-instance.dev"
```

出力:

```
{  
  "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-  
Host-28aef605"  
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の「[ホストの作成 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateHost](#)」の「」を参照してください。

delete-connection

次の例は、delete-connection を使用する方法を説明しています。

AWS CLI

接続を削除するには

次のdelete-connection例は、接続を削除する方法を示しています。

```
aws codestar-connections delete-connection \  
  --connection-arn arn:aws:codestar-connections:us-west-2:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

このコマンドでは何も出力されません。

詳細については、[「デベロッパーツールコンソールユーザーガイド」の「接続の削除 \(CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConnection](#)」の「」を参照してください。

delete-host

次の例は、delete-host を使用する方法を説明しています。

AWS CLI

ホストを削除するには

次のdelete-host例は、ホストを削除する方法を示しています。ホストを削除する前に、ホストに関連付けられたすべての接続を削除する必要があります。

```
aws codestar-connections delete-host \  
  --host-arn "arn:aws:codestar-connections:us-east-1 :123456789012:host/My-  
Host-28aef605"
```

このコマンドでは何も出力されません。

詳細については、「デベロッパーツールコンソールユーザーガイド」の[「ホストの削除 \(CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteHost](#)」の「」を参照してください。

get-connection

次のコード例は、get-connection を使用する方法を示しています。

AWS CLI

接続に関する情報を取得するには

次のget-connection例は、接続の詳細を示しています。

```
aws codestar-connections get-connection \  
  --connection-id "arn:aws:codestar-connections:us-east-1 :123456789012:connection/My-  
Connection-28aef605"
```

```
--connection-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

出力:

```
{
  "Connection": {
    "ConnectionName": "MyConnection",
    "ConnectionArn": "arn:aws:codestar-connections:us-east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
    "ProviderType": "Bitbucket",
    "OwnerAccountId": "123456789012",
    "ConnectionStatus": "AVAILABLE"
  }
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の[「接続の詳細を表示する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConnection](#)」の「」を参照してください。

get-host

次の例は、get-host を使用する方法を説明しています。

AWS CLI

ホストに関する情報を取得するには

次のget-host例は、ホストの詳細を示しています。

```
aws codestar-connections get-host \
  --host-arn arn:aws:codestar-connections:us-east-1:123456789012:host/MyHost-28aef605
```

出力:

```
{
  "Name": "MyHost",
  "Status": "AVAILABLE",
```

```
"ProviderType": "GitHubEnterpriseServer",  
"ProviderEndpoint": "https://test-instance-1.dev/"  
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の[「ホストの詳細の表示 \(CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetHost](#)」の「」を参照してください。

list-connections

次の例は、list-connections を使用する方法を説明しています。

AWS CLI

接続を一覧表示するには

次のlist-connections例では、Bitbucket プロバイダータイプのアカウント内のすべての接続のリストを取得します。

```
aws codestar-connections list-connections \  
--provider-type Bitbucket \  
--max-results 5 \  
--next-token: next-token
```

出力:

```
{  
  "Connections": [  
    {  
      "ConnectionName": "my-connection",  
      "ProviderType": "Bitbucket",  
      "Status": "PENDING",  
      "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",  
      "OwnerAccountId": "123456789012"  
    },  
    {  
      "ConnectionName": "my-other-connection",  
      "ProviderType": "Bitbucket",  

```



```
    "Status": "AVAILABLE",
    "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
    "OwnerAccountId": "123456789012"
  },
],
"NextToken": "next-token"
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の「[接続 \(CLI\) を一覧表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListConnections](#)」の「」を参照してください。

list-hosts

次の例は、list-hosts を使用する方法を説明しています。

AWS CLI

ホストを一覧表示するには

次のlist-hosts例では、アカウント内のすべてのホストのリストを取得します。

```
aws codestar-connections list-hosts
```

出力:

```
{
  "Hosts": [
    {
      "Name": "My-Host",
      "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605",
      "ProviderType": "GitHubEnterpriseServer",
      "ProviderEndpoint": "https://my-instance.test.dev",
      "Status": "AVAILABLE"
    }
  ]
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の[「ホストのリスト \(CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListHosts](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

タグを一覧表示するには

次のlist-tags-for-resource例では、指定された接続リソースにアタッチされたすべてのタグのリストを取得します。

```
aws codestar-connections list-tags-for-resource \
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

出力:

```
{
  "Tags": [
    {
      "Key": "Project",
      "Value": "ProjectA"
    },
    {
      "Key": "ReadOnly",
      "Value": "true"
    }
  ]
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の[「接続リソースのタグを表示する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、提供されたタグのセットを接続に関連付けます。このコマンドを使用して、タグを追加または編集します。

```
aws codestar-connections tag-resource \  
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \  
  --tags Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
```

このコマンドでは何も出力されません。

詳細については、「デベロッパーツールコンソールユーザーガイド」の「[接続リソースにタグを追加する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

接続リソースから AWS タグを削除するには

以下では、指定されたリソースからタグuntag-resourceを削除します。

```
aws codestar-connections untag-resource \  
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f \  
  --tag-keys Project ReadOnly
```

出力:

```
{
```

```
"Tags": []
}
```

詳細については、「デベロッパーツールコンソールユーザーガイド」の「[接続リソースからタグを削除する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

を使用した Amazon Cognito Identity の例 AWS CLI

次のコード例は、Amazon Cognito Identity AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-identity-pool

次のコード例は、create-identity-pool を使用する方法を示しています。

AWS CLI

Cognito アイデンティティプールプロバイダーを含むアイデンティティプールを作成するには

この例では、 という名前の ID プールを作成します MyIdentityPool。これには Cognito アイデンティティプールプロバイダーが含まれます。認証されていないアイデンティティは許可されません。

コマンド:

```
aws cognito-identity create-identity-pool --identity-pool-name MyIdentityPool --no-allow-unauthenticated-identities --cognito-identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-west-2_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

出力:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
    {
      "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_11111111",
      "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateIdentityPool](#)」の「」を参照してください。

delete-identities

次のコード例は、delete-identities を使用する方法を示しています。

AWS CLI

アイデンティティプールを削除するには

この例では、ID プールを削除します。

コマンド:

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-west-2:11111111-1111-1111-1111-111111111111"
```

出力:

```
{
```

```
"UnprocessedIdentityIds": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIdentities](#)」の「」を参照してください。

delete-identity-pool

次の例は、delete-identity-pool を使用する方法を説明しています。

AWS CLI

アイデンティティプールを削除するには

次の delete-identity-pool 例では、指定したアイデンティティプールを削除します。

コマンド:

```
aws cognito-identity delete-identity-pool \  
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIdentityPool](#)」の「」を参照してください。

describe-identity-pool

次の例は、describe-identity-pool を使用する方法を説明しています。

AWS CLI

ID プールを記述するには

この例では、ID プールについて説明します。

コマンド:

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-  
west-2:11111111-1111-1111-1111-111111111111"
```

出力:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
    {
      "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_1111111111",
      "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeIdentityPool](#)」の「」を参照してください。

get-identity-pool-roles

次の例は、get-identity-pool-roles を使用する方法を説明しています。

AWS CLI

ID プールロールを取得するには

この例では、ID プールロールを取得します。

コマンド:

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

出力:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "Roles": {
    "authenticated": "arn:aws:iam::111111111111:role/Cognito_MyIdentityPoolAuth_Role",
    "unauthenticated": "arn:aws:iam::111111111111:role/Cognito_MyIdentityPoolUnauth_Role"
  }
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIdentityPoolRoles](#)」の「」を参照してください。

list-identity-pools

次のコード例は、list-identity-pools を使用する方法を示しています。

AWS CLI

アイデンティティプールを一覧表示するには

この例ではアイデンティティプールを一覧表示します。最大 20 個のアイデンティティが一覧表示されます。

コマンド:

```
aws cognito-identity list-identity-pools --max-results 20
```

出力:

```
{
  "IdentityPools": [
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "MyIdentityPool"
    },
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "AnotherIdentityPool"
    },
    {
      "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
      "IdentityPoolName": "IdentityPoolRegionA"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListIdentityPools](#)」の「」を参照してください。

set-identity-pool-roles

次のコード例は、set-identity-pool-roles を使用する方法を示しています。

AWS CLI

ID プールロールを設定するには

次のset-identity-pool-roles例では、ID プールロールを設定します。

```
aws cognito-identity set-identity-pool-roles \  
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111" \  
  --roles authenticated="arn:aws:iam::111111111111:role/  
Cognito_MyIdentityPoolAuth_Role"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetIdentityPoolRoles](#)」の「」を参照してください。

update-identity-pool

次のコード例は、update-identity-pool を使用する方法を示しています。

AWS CLI

ID プールを更新するには

この例では、ID プールを更新します。名前を に設定します MyIdentityPool。Cognito を ID プロバイダーとして追加します。認証されていない ID は許可されません。

コマンド:

```
aws cognito-identity update-identity-pool --identity-pool-id "us-  
west-2:11111111-1111-1111-1111-111111111111" --identity-pool-  
name "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-  
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-  
west-2_11111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

出力:

```
{  
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",  
  "IdentityPoolName": "MyIdentityPool",
```

```
"AllowUnauthenticatedIdentities": false,
"CognitoIdentityProviders": [
  {
    "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
    "ClientId": "3n4b5urk1ft4f13mg5e62d9ado",
    "ServerSideTokenCheck": false
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateIdentityPool](#)」の「」を参照してください。

を使用した Amazon Cognito Identity Provider の例 AWS CLI

次のコード例は、Amazon Cognito Identity Provider AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-custom-attributes

次の例は、add-custom-attributes を使用する方法を説明しています。

AWS CLI

カスタム属性を追加するには

この例では、カスタム属性 CustomAttr1 をユーザープールに追加します。文字列タイプであり、1 文字以上 15 文字以下が必要です。この値は必須ではありません。

コマンド:

```
aws cognito-idp add-custom-attributes --user-pool-id us-west-2_aaaaaaaaa --custom-attributes Name="CustomAttr1",AttributeDataType="String",DeveloperOnlyAttribute=false,Required=false,S
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddCustomAttributes](#)」の「」を参照してください。

admim-disable-user

次のコード例は、admim-disable-user を使用する方法を示しています。

AWS CLI

ユーザーを無効にするには

この例では、ユーザー `jane@example.com` を無効にします。

コマンド:

```
aws cognito-idp admin-disable-user --user-pool-id us-west-2_aaaaaaaaa --username jane@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdmimDisableUser](#)」の「」を参照してください。

admim-enable-user

次の例は、admim-enable-user を使用する方法を説明しています。

AWS CLI

ユーザーを有効にするには

この例では、ユーザー名 `jane@example.com` を有効にします。

コマンド:

```
aws cognito-idp admin-enable-user --user-pool-id us-west-2_aaaaaaaaa --username jane@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminEnableUser](#)」の「」を参照してください。

admin-add-user-to-group

次の例は、admin-add-user-to-group を使用する方法を説明しています。

AWS CLI

ユーザーをグループに追加するには

この例では、ユーザー Jane をグループに追加します MyGroup。

コマンド:

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --  
username Jane --group-name MyGroup
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminAddUserToGroup](#)」の「」を参照してください。

admin-confirm-sign-up

次のコード例は、admin-confirm-sign-up を使用する方法を示しています。

AWS CLI

ユーザー登録を確認するには

この例では、ユーザー jane@example.com を確認します。

コマンド:

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminConfirmSignUp](#)」の「」を参照してください。

admin-create-user

次のコード例は、`admin-create-user` を使用する方法を示しています。

AWS CLI

ユーザーを作成するには

次の`admin-create-user`例では、指定された設定の E メールアドレスと電話番号を持つユーザーを作成します。

```
aws cognito-idp admin-create-user \  
  --user-pool-id us-west-2_aaaaaaaaa \  
  --username diego \  
  --user-attributes Name=email,Value=diego@example.com  
Name=phone_number,Value="+15555551212" \  
  --message-action SUPPRESS
```

出力:

```
{  
  "User": {  
    "Username": "diego",  
    "Attributes": [  
      {  
        "Name": "sub",  
        "Value": "7325c1de-b05b-4f84-b321-9adc6e61f4a2"  
      },  
      {  
        "Name": "phone_number",  
        "Value": "+15555551212"  
      },  
      {  
        "Name": "email",  
        "Value": "diego@example.com"  
      }  
    ],  
    "UserCreateDate": 1548099495.428,  
    "UserLastModifiedDate": 1548099495.428,  
    "Enabled": true,  
    "UserStatus": "FORCE_CHANGE_PASSWORD"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminCreateUser](#)」の「」を参照してください。

admin-delete-user-attributes

次の例は、`admin-delete-user-attributes` を使用する方法を説明しています。

AWS CLI

ユーザー属性を削除するには

この例では、ユーザー `diego@example.com` のカスタム属性 `CustomAttr1` を削除します。

コマンド:

```
aws cognito-idp admin-delete-user-attributes --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com --user-attribute-names "custom:CustomAttr1"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminDeleteUserAttributes](#)」の「」を参照してください。

admin-delete-user

次の例は、`admin-delete-user` を使用する方法を説明しています。

AWS CLI

ユーザーを削除するには

この例では、ユーザーを削除します。

コマンド:

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminDeleteUser](#)」の「」を参照してください。

admin-forget-device

次の例は、admin-forget-device を使用する方法を説明しています。

AWS CLI

デバイスを忘れるには

この例では、ユーザー名 jane@example.com のデバイスを忘れます

コマンド:

```
aws cognito-idp admin-forget-device --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com --device-key us-west-2_abcd_1234-5678
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminForgetDevice](#)」の「」を参照してください。

admin-get-device

次の例は、admin-get-device を使用する方法を説明しています。

AWS CLI

デバイスを取得するには

この例では、ユーザー名 jane@example.com のデバイスを取得します。

コマンド:

```
aws cognito-idp admin-get-device --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com --device-key us-west-2_abcd_1234-5678
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminGetDevice](#)」の「」を参照してください。

admin-get-user

次の例は、admin-get-user を使用する方法を説明しています。

AWS CLI

ユーザーを取得するには

この例では、ユーザー名 `jane@example.com` に関する情報を取得します。

コマンド:

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

出力:

```
{  
  "Username": "4320de44-2322-4620-999b-5e2e1c8df013",  
  "Enabled": true,  
  "UserStatus": "FORCE_CHANGE_PASSWORD",  
  "UserCreateDate": 1548108509.537,  
  "UserAttributes": [  
    {  
      "Name": "sub",  
      "Value": "4320de44-2322-4620-999b-5e2e1c8df013"  
    },  
    {  
      "Name": "email_verified",  
      "Value": "true"  
    },  
    {  
      "Name": "phone_number_verified",  
      "Value": "true"  
    },  
    {  
      "Name": "phone_number",  
      "Value": "+01115551212"  
    },  
    {  
      "Name": "email",  
      "Value": "jane@example.com"  
    }  
  ],  
  "UserLastModifiedDate": 1548108509.537  
}
```


- API 詳細については、AWS CLI 「コマンドリファレンス [AdminGetUser](#)」の「」を参照してください。

admin-initiate-auth

次の例は、admin-initiate-auth を使用する方法を説明しています。

AWS CLI

認可を開始するには

この例では、ユーザー名 jane@example.com の ADMIN_NO_SRP_AUTH フローを使用して承認を開始します。

クライアントでは、サーバーベースの認証 (ADMIN_NO_SRP_AUTH) API のサインインが有効になっている必要があります。

戻り値のセッション情報を使用して admin-respond-to-auth-challenge を呼び出します。

コマンド:

```
aws cognito-idp admin-initiate-auth --user-pool-id us-west-2_aaaaaaaaa --client-id 3n4b5urk1ft4fl3mg5e62d9ado --auth-flow ADMIN_NO_SRP_AUTH --auth-parameters USERNAME=jane@example.com,PASSWORD=password
```

出力:

```
{
  "ChallengeName": "NEW_PASSWORD_REQUIRED",
  "Session": "SESSION",
  "ChallengeParameters": {
    "USER_ID_FOR_SRP": "84514837-dcbc-4af1-abff-f3c109334894",
    "requiredAttributes": "[]",
    "userAttributes": "{\"email_verified\": \"true\", \"phone_number_verified\": \"true\", \"phone_number\": \"+01xxx5550100\", \"email\": \"jane@example.com\"}"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminInitiateAuth](#)」の「」を参照してください。

admin-list-devices

次のコード例は、admin-list-devices を使用する方法を示しています。

AWS CLI

ユーザーのデバイスを一覧表示するには

この例では、ユーザー名 jane@example.com のデバイスを一覧表示します。

コマンド:

```
aws cognito-idp admin-list-devices --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminListDevices](#)」の「」を参照してください。

admin-list-groups-for-user

次の例は、admin-list-groups-for-user を使用する方法を説明しています。

AWS CLI

ユーザーのグループを一覧表示するには

この例では、ユーザー名 jane@example.com のグループを一覧表示します。

コマンド:

```
aws cognito-idp admin-list-groups-for-user --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

出力:

```
{  
  "Groups": [  
    {  
      "Description": "Sample group",  
      "Precedence": 1,  
      "LastModifiedDate": 1548097827.125,  
    }  
  ]  
}
```

```
    "RoleArn": "arn:aws:iam::111111111111:role/SampleRole",
    "GroupName": "SampleGroup",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "CreationDate": 1548097827.125
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminListGroupForUser](#)」の「」を参照してください。

admin-list-user-auth-events

次のコード例は、admin-list-user-auth-events を使用する方法を示しています。

AWS CLI

ユーザーの承認イベントを一覧表示するには

この例では、ユーザー名 diego@example.com の承認イベントを一覧表示します。

コマンド:

```
aws cognito-idp admin-list-user-auth-events --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminListUserAuthEvents](#)」の「」を参照してください。

admin-remove-user-from-group

次のコード例は、admin-remove-user-from-group を使用する方法を示しています。

AWS CLI

グループからユーザーを削除するには

この例では、 から jane@example.com を削除します SampleGroup。

コマンド:

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaa --  
username jane@example.com --group-name SampleGroup
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminRemoveUserFromGroup](#)」の「」を参照してください。

admin-reset-user-password

次の例は、admin-reset-user-password を使用する方法を説明しています。

AWS CLI

ユーザーパスワードをリセットするには

この例では、diego@example.com のパスワードをリセットします。

コマンド:

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaa --  
username diego@example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminResetUserPassword](#)」の「」を参照してください。

admin-set-user-mfa-preference

次のコード例は、admin-set-user-mfa-preference を使用する方法を示しています。

AWS CLI

ユーザーMFA設定を設定するには

この例では、ユーザー名 diego@example.com のSMSMFAプリファレンスを設定します。

コマンド:

```
aws cognito-idp admin-set-user-mfa-preference --user-pool-id us-west-2_aaaaaaaa --  
username diego@example.com --sms-mfa-settings Enabled=false,PreferredMfa=false
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdminSetUserMfaPreference](#)」の「」を参照してください。

admin-set-user-settings

次の例は、`admin-set-user-settings` を使用する方法を説明しています。

AWS CLI

ユーザー設定を設定するには

この例では、ユーザー名 `diego@example.com` のMFA配信設定を `EMAIL` に設定します。

コマンド:

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --username diego@example.com --mfa-options DeliveryMedium=EMAIL
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminSetUserSettings](#)」の「」を参照してください。

admin-update-auth-event-feedback

次の例は、`admin-update-auth-event-feedback` を使用する方法を説明しています。

AWS CLI

認証イベントにフィードバックを提供するには

この例では、`event-id` で識別される認証イベントのフィードバック値を `Valid` に設定します。

コマンド:

```
aws cognito-idp admin-update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa --username diego@example.com --event-id c2c2cf89-c0d3-482d-aba6-99d78a5b0bfe --feedback-value Valid
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminUpdateAuthEventFeedback](#)」の「」を参照してください。

admin-update-device-status

次のコード例は、`admin-update-device-status` を使用する方法を示しています。

AWS CLI

デバイスのステータスを更新するには

この例では、device-key で識別されるデバイスの device remembered ステータスを not_remembered に設定します。

コマンド:

```
aws cognito-idp admin-update-device-status --user-pool-id us-west-2_aaaaaaaaa
--username diego@example.com --device-key xxxx --device-remembered-
status not_remembered
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminUpdateDeviceStatus](#)」の「」を参照してください。

admin-update-user-attributes

次の例は、admin-update-user-attributes を使用する方法を説明しています。

AWS CLI

ユーザー属性を更新するには

この例では、ユーザー diego@example.com のカスタムユーザー属性 CustomAttr1 を更新します。

コマンド:

```
aws cognito-idp admin-update-user-attributes --user-pool-id us-
west-2_aaaaaaaaa --username diego@example.com --user-attributes
Name="custom:CustomAttr1",Value="Purple"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AdminUpdateUserAttributes](#)」の「」を参照してください。

change-password

次のコード例は、change-password を使用する方法を示しています。

AWS CLI

パスワードを変更するには

この例では、パスワードを変更します。

コマンド:

```
aws cognito-idp change-password --previous-password OldPassword --proposed-  
password NewPassword --access-token ACCESS_TOKEN
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ChangePassword](#)」の「」を参照してください。

confirm-forgot-password

次のコード例は、confirm-forgot-password を使用する方法を示しています。

AWS CLI

忘れたパスワードを確認するには

この例では、ユーザー名 diego@example.com のパスワードを忘れたことを確認します。

コマンド:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --  
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfirmForgotPassword](#)」の「」を参照してください。

confirm-sign-up

次のコード例は、confirm-sign-up を使用する方法を示しています。

AWS CLI

サインアップを確認するには

この例では、ユーザー名 diego@example.com のサインアップを確認します。

コマンド:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4f13mg5e62d9ado --  
username=diego@example.com --confirmation-code CONF_CODE
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ConfirmSignUp](#)」の「」を参照してください。

create-group

次のコード例は、create-group を使用する方法を示しています。

AWS CLI

グループを作成するには

この例では、説明を含むグループを作成します。

コマンド:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-  
name MyNewGroup --description "New group."
```

出力:

```
{  
  "Group": {  
    "GroupName": "MyNewGroup",  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "Description": "New group.",  
    "LastModifiedDate": 1548270073.795,  
    "CreationDate": 1548270073.795  
  }  
}
```

ロールと優先順位を持つグループを作成するには

この例では、説明を含むグループを作成します。また、ロールと優先順位も含まれます。

コマンド:


```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaaa --group-name MyNewGroupWithRole --description "New group with a role." --role-arn arn:aws:iam::111111111111:role/MyNewGroupRole --precedence 2
```

出力:

```
{
  "Group": {
    "GroupName": "MyNewGroupWithRole",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "Description": "New group with a role.",
    "RoleArn": "arn:aws:iam::111111111111:role/MyNewGroupRole",
    "Precedence": 2,
    "LastModifiedDate": 1548270211.761,
    "CreationDate": 1548270211.761
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateGroup](#)」の「」を参照してください。

create-user-import-job

次の例は、create-user-import-job を使用する方法を説明しています。

AWS CLI

ユーザーインポートジョブを作成するには

この例では、という名前のユーザーインポートジョブを作成します MyImportJob。

ユーザーのインポートの詳細については、CSV「ファイルからユーザープールにユーザーをインポートする」を参照してください。

コマンド:

```
aws cognito-idp create-user-import-job --user-pool-id us-west-2_aaaaaaaaaa --job-name MyImportJob --cloud-watch-logs-role-arn arn:aws:iam::111111111111:role/CognitoCloudWatchLogsRole
```

出力:

```
{
  "UserImportJob": {
    "JobName": "MyImportJob",
    "JobId": "import-qQ0DCt2fRh",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548271795.471,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}
```

署名付き を使用して、.csv ファイルを curl でアップロードしますURL。

コマンド:

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
"PRE_SIGNED_URL"
```

- API 詳細については、AWS CLI 「 コマンドリファレンス [CreateUserImportJob](#)」の「」を参照してください。

create-user-pool-client

次の例は、create-user-pool-client を使用する方法を説明しています。

AWS CLI

ユーザープールクライアントを作成するには

この例では、__ と USER_ADMINNOPASSWORD__AUTH の 2 つの明示的な承認フローを持つ新しいユーザープールクライアントを作成しますSRPAUTH。

コマンド:

```
aws cognito-idp create-user-pool-client --user-pool-id us-west-2_aaaaaaaaaa
--client-name MyNewClient --no-generate-secret --explicit-auth-
flows "USER_PASSWORD_AUTH" "ADMIN_NO_SRP_AUTH"
```

出力:

```
{
  "UserPoolClient": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientName": "MyNewClient",
    "ClientId": "6p3bs000no6a4ue1idruvd05ad",
    "LastModifiedDate": 1548697449.497,
    "CreationDate": 1548697449.497,
    "RefreshTokenValidity": 30,
    "ExplicitAuthFlows": [
      "USER_PASSWORD_AUTH",
      "ADMIN_NO_SRP_AUTH"
    ],
    "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUserPoolClient](#)」の「」を参照してください。

create-user-pool-domain

次のコード例は、create-user-pool-domain を使用する方法を示しています。

AWS CLI

ユーザープールドメインを作成するには

この例では、__ と USER_ADMINNOPASSWORD__AUTH の 2 つの明示的な承認フローを持つ新しいユーザープールドメインを作成しますSRPAUTH。

コマンド:

```
aws cognito-idp create-user-pool-domain --user-pool-id us-west-2_aaaaaaaaa --  
domain my-new-domain
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUserPoolDomain](#)」の「」を参照してください。

create-user-pool

次の例は、create-user-pool を使用する方法を説明しています。

AWS CLI

最小構成のユーザープールを作成するには

この例では、デフォルト値 MyUserPool を使用して という名前のユーザープールを作成します。必要な属性やアプリケーションクライアントはありません。MFA および の高度なセキュリティは無効になっています。

コマンド:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

出力:

```
{
  "UserPool": {
    "SchemaAttributes": [
      {
        "Name": "sub",
        "StringAttributeConstraints": {
          "MinLength": "1",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": true,
        "AttributeDataType": "String",
        "Mutable": false
      },
      {
        "Name": "name",
        "StringAttributeConstraints": {
          "MinLength": "0",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
        "Mutable": true
      }
    ],
  }
}
```

```
{
  "Name": "given_name",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "family_name",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "middle_name",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "nickname",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
}
```

```
{
  "Name": "preferred_username",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "profile",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "picture",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "website",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
}
```

```
{
  "Name": "email",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "AttributeDataType": "Boolean",
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "Name": "email_verified",
  "Mutable": true
},
{
  "Name": "gender",
  "StringAttributeConstraints": {
    "MinLength": "0",
    "MaxLength": "2048"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "birthdate",
  "StringAttributeConstraints": {
    "MinLength": "10",
    "MaxLength": "10"
  },
  "DeveloperOnlyAttribute": false,
  "Required": false,
  "AttributeDataType": "String",
  "Mutable": true
},
{
  "Name": "zoneinfo",
  "StringAttributeConstraints": {
    "MinLength": "0",
```

```
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "locale",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "phone_number",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "phone_number_verified",
    "Mutable": true
},
{
    "Name": "address",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
```



```
        "AttributeDataType": "String",
        "Mutable": true
    },
    {
        "Name": "updated_at",
        "NumberAttributeConstraints": {
            "MinValue": "0"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "Number",
        "Mutable": true
    }
],
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"LastModifiedDate": 1547833345.777,
"AdminCreateUserConfig": {
    "UnusedAccountValidityDays": 7,
    "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {},
"Policies": {
    "PasswordPolicy": {
        "RequireLowercase": true,
        "RequireSymbols": true,
        "RequireNumbers": true,
        "MinimumLength": 8,
        "RequireUppercase": true
    }
},
"CreationDate": 1547833345.777,
"EstimatedNumberOfUsers": 0,
"Id": "us-west-2_aaaaaaaaa",
"LambdaConfig": {}
}
}
```

2つの必須属性でユーザープールを作成するには

この例では、ユーザープールを作成します MyUserPool。プールは、Eメールをユーザー名属性として受け入れるように設定されています。また、Amazon Simple Email Service を使用して、Eメールの送信元アドレスを検証済みのアドレスに設定します。

コマンド:

```
aws cognito-idp create-user-pool --pool-name MyUserPool --username-attributes "email" --email-configuration=SourceArn="arn:aws:ses:us-east-1:111111111111:identity/jane@example.com",ReplyToEmailAddress="jane@example.com"
```

出力:

```
{
  "UserPool": {
    "SchemaAttributes": [
      {
        "Name": "sub",
        "StringAttributeConstraints": {
          "MinLength": "1",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": true,
        "AttributeDataType": "String",
        "Mutable": false
      },
      {
        "Name": "name",
        "StringAttributeConstraints": {
          "MinLength": "0",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
        "Mutable": true
      },
      {
        "Name": "given_name",
        "StringAttributeConstraints": {
          "MinLength": "0",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
```

```
    "Mutable": true
  },
  {
    "Name": "family_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "middle_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "nickname",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "preferred_username",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
```

```
    "Mutable": true
  },
  {
    "Name": "profile",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "picture",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "website",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "email",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
```

```
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "email_verified",
    "Mutable": true
  },
  {
    "Name": "gender",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "birthdate",
    "StringAttributeConstraints": {
      "MinLength": "10",
      "MaxLength": "10"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "zoneinfo",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "locale",
```

```
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "phone_number",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "phone_number_verified",
    "Mutable": true
  },
  {
    "Name": "address",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "updated_at",
    "NumberAttributeConstraints": {
      "MinValue": "0"
    },
    "DeveloperOnlyAttribute": false,
```

```
        "Required": false,
        "AttributeDataType": "Number",
        "Mutable": true
    }
],
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"LastModifiedDate": 1547837788.189,
"AdminCreateUserConfig": {
    "UnusedAccountValidityDays": 7,
    "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {
    "ReplyToEmailAddress": "jane@example.com",
    "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
},
"Policies": {
    "PasswordPolicy": {
        "RequireLowercase": true,
        "RequireSymbols": true,
        "RequireNumbers": true,
        "MinimumLength": 8,
        "RequireUppercase": true
    }
},
"UsernameAttributes": [
    "email"
],
"CreationDate": 1547837788.189,
"EstimatedNumberOfUsers": 0,
"Id": "us-west-2_aaaaaaaaa",
"LambdaConfig": {}
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateUserPool](#)」の「」を参照してください。

delete-group

次のコード例は、delete-group を使用する方法を示しています。

AWS CLI

グループを削除するには

この例では、グループを削除します。

コマンド:

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroupName
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGroup](#)」の「」を参照してください。

delete-identity-provider

次の例は、delete-identity-provider を使用する方法を説明しています。

AWS CLI

ID プロバイダーを削除するには

この例では、ID プロバイダーを削除します。

コマンド:

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaaa --provider-name Facebook
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIdentityProvider](#)」の「」を参照してください。

delete-resource-server

次の例は、delete-resource-server を使用する方法を説明しています。

AWS CLI

リソースサーバーを削除するには

この例では、weather.example.com という名前のリソースサーバーを削除します。

コマンド:


```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --  
identifier weather.example.com
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResourceServer](#)」の「」を参照してください。

delete-user-attributes

次のコード例は、delete-user-attributes を使用する方法を示しています。

AWS CLI

ユーザー属性を削除するには

この例では、ユーザー属性 FAVORITE 「_ANIMAL」を削除します。

コマンド:

```
aws cognito-idp delete-user-attributes --access-token ACCESS_TOKEN --user-attribute-  
names "FAVORITE_ANIMAL"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUserAttributes](#)」の「」を参照してください。

delete-user-pool-client

次の例は、delete-user-pool-client を使用する方法を説明しています。

AWS CLI

ユーザープールクライアントを削除するには

この例では、ユーザープールクライアントを削除します。

コマンド:

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-  
id 38fjsnc484p94kpbsnet7mpld0
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUserPoolClient](#)」の「」を参照してください。

delete-user-pool-domain

次の例は、delete-user-pool-domain を使用する方法を説明しています。

AWS CLI

ユーザープールドメインを削除するには

次のdelete-user-pool-domain例では、 という名前のユーザープールドメインを削除します。 my-domain

```
aws cognito-idp delete-user-pool-domain \  
  --user-pool-id us-west-2_aaaaaaaaa \  
  --domain my-domain
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUserPoolDomain](#)」の「」を参照してください。

delete-user-pool

次のコード例は、delete-user-pool を使用する方法を示しています。

AWS CLI

ユーザープールを削除するには

この例では、ユーザープール ID、us-west-2_aaaaaaaaa を使用してユーザープールを削除します。

コマンド:

```
aws cognito-idp delete-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUserPool](#)」の「」を参照してください。

delete-user

次の例は、delete-user を使用する方法を説明しています。

AWS CLI

ユーザーを削除するには

この例では、ユーザーを削除します。

コマンド:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteUser](#)」の「」を参照してください。

describe-identity-provider

次の例は、describe-identity-provider を使用する方法を説明しています。

AWS CLI

ID プロバイダーを記述するには

この例では、Facebook という名前の ID プロバイダーについて説明します。

コマンド:

```
aws cognito-idp describe-identity-provider --user-pool-id us-west-2_aaaaaaaaa --  
provider-name Facebook
```

出力:

```
{  
  "IdentityProvider": {  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "ProviderName": "Facebook",  
    "ProviderType": "Facebook",  
    "ProviderDetails": {  
      "attributes_url": "https://graph.facebook.com/me?fields=",  
      "attributes_url_add_attributes": "true",  
      "authorize_scopes": "myscope",  
      "authorize_url": "https://www.facebook.com/v2.9/dialog/oauth",  
      "client_id": "11111",  
      "client_secret": "11111",  
      "token_request_method": "GET",
```

```
    "token_url": "https://graph.facebook.com/v2.9/oauth/access_token"
  },
  "AttributeMapping": {
    "username": "id"
  },
  },
  "IdpIdentifiers": [],
  "LastModifiedDate": 1548105901.736,
  "CreationDate": 1548105901.736
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeIdentityProvider](#)」の「」を参照してください。

describe-resource-server

次のコード例は、describe-resource-server を使用する方法を示しています。

AWS CLI

リソースサーバーを記述するには

この例では、リソースサーバー weather.example.com について説明します。

コマンド:

```
aws cognito-idp describe-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

出力:

```
{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Identifier": "weather.example.com",
    "Name": "Weather",
    "Scopes": [
      {
        "ScopeName": "weather.update",
        "ScopeDescription": "Update weather forecast"
      },
      {
        "ScopeName": "weather.read",
```

```
        "ScopeDescription": "Read weather forecasts"
      },
      {
        "ScopeName": "weather.delete",
        "ScopeDescription": "Delete a weather forecast"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeResourceServer](#)」の「」を参照してください。

describe-risk-configuration

次のコード例は、describe-risk-configuration を使用する方法を示しています。

AWS CLI

リスク設定を記述するには

この例では、us-west-2_aaaaaaaaaa プールに関連するリスク設定について説明します。

コマンド:

```
aws cognito-idp describe-risk-configuration --user-pool-id us-west-2_aaaaaaaaaa
```

出力:

```
{
  "RiskConfiguration": {
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "CompromisedCredentialsRiskConfiguration": {
      "EventFilter": [
        "SIGN_IN",
        "SIGN_UP",
        "PASSWORD_CHANGE"
      ],
      "Actions": {
        "EventAction": "BLOCK"
      }
    },
    "AccountTakeoverRiskConfiguration": {
```

```
"NotifyConfiguration": {
  "From": "diego@example.com",
  "ReplyTo": "diego@example.com",
  "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
diego@example.com",
  "BlockEmail": {
    "Subject": "Blocked sign-in attempt",
    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
{country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
link</a> to let us know</pre>\n</body>\n</html>",
    "TextBody": "We blocked an unrecognized sign-in to your account
with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
{city}, {country}\nIf this sign-in was not by you, you should change your password
and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
you can follow {one-click-link-valid} to let us know"
  },
  "NoActionEmail": {
    "Subject": "New sign-in attempt",
    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
{country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
link</a> to let us know</pre>\n</body>\n</html>",
    "TextBody": "We observed an unrecognized sign-in to your account
with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
{city}, {country}\nIf this sign-in was not by you, you should change your password
and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
you can follow {one-click-link-valid} to let us know"
  },
  "MfaEmail": {
    "Subject": "New sign-in attempt",
    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML email
context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We required
you to use multi-factor authentication for the following sign-in attempt:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
{country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
```

```
password and notify us by clicking on <a href={one-click-link-invalid}>this link</a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this link</a> to let us know</pre>\n</body>\n</html>",
    "TextBody": "We required you to use multi-factor authentication for the following sign-in attempt:\nTime: {login-time}\nDevice: {device-name}\nLocation: {city}, {country}\nIf this sign-in was not by you, you should change your password and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you, you can follow {one-click-link-valid} to let us know"
  }
},
"Actions": {
  "LowAction": {
    "Notify": true,
    "EventAction": "NO_ACTION"
  },
  "MediumAction": {
    "Notify": true,
    "EventAction": "MFA_IF_CONFIGURED"
  },
  "HighAction": {
    "Notify": true,
    "EventAction": "MFA_IF_CONFIGURED"
  }
}
}
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRiskConfiguration](#)」の「」を参照してください。

describe-user-import-job

次のコード例は、describe-user-import-job を使用する方法を示しています。

AWS CLI

ユーザーインポートジョブを記述するには

この例では、ユーザー入力ジョブについて説明します。

ユーザーのインポートの詳細については、CSV「ファイルからユーザープールにユーザーをインポートする」を参照してください。

コマンド:

```
aws cognito-idp describe-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-id import-TZqNQvDRnW
```

出力:

```
{
  "UserImportJob": {
    "JobName": "import-Test1",
    "JobId": "import-TZqNQvDRnW",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED URL",
    "CreationDate": 1548271708.512,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUserImportJob](#)」の「」を参照してください。

describe-user-pool-client

次の例は、describe-user-pool-client を使用する方法を説明しています。

AWS CLI

ユーザープールクライアントを記述するには

この例では、ユーザープールクライアントについて説明します。

コマンド:

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-id 38fjsnc484p94kqpsnet7mp1d0
```


出力:

```
{
  "UserPoolClient": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientName": "MyApp",
    "ClientId": "38fjsnc484p94kpbsnet7mpld0",
    "ClientSecret": "CLIENT_SECRET",
    "LastModifiedDate": 1548108676.163,
    "CreationDate": 1548108676.163,
    "RefreshTokenValidity": 30,
    "ReadAttributes": [
      "address",
      "birthdate",
      "custom:CustomAttr1",
      "custom:CustomAttr2",
      "email",
      "email_verified",
      "family_name",
      "gender",
      "given_name",
      "locale",
      "middle_name",
      "name",
      "nickname",
      "phone_number",
      "phone_number_verified",
      "picture",
      "preferred_username",
      "profile",
      "updated_at",
      "website",
      "zoneinfo"
    ],
    "WriteAttributes": [
      "address",
      "birthdate",
      "custom:CustomAttr1",
      "custom:CustomAttr2",
      "email",
      "family_name",
      "gender",
      "given_name",
      "locale",
```

```
        "middle_name",
        "name",
        "nickname",
        "phone_number",
        "picture",
        "preferred_username",
        "profile",
        "updated_at",
        "website",
        "zoneinfo"
    ],
    "ExplicitAuthFlows": [
        "ADMIN_NO_SRP_AUTH",
        "USER_PASSWORD_AUTH"
    ],
    "AllowedOAuthFlowsUserPoolClient": false
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeUserPoolClient](#)」の「」を参照してください。

describe-user-pool-domain

次の例は、describe-user-pool-domain を使用する方法を説明しています。

AWS CLI

ユーザープールクライアントを記述するには

この例では、my-domain という名前のユーザープールドメインについて説明します。

コマンド:

```
aws cognito-idp describe-user-pool-domain --domain my-domain
```

出力:

```
{
  "DomainDescription": {
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "AWSAccountId": "111111111111",
    "Domain": "my-domain",
```

```
"S3Bucket": "aws-cognito-prod-pdx-assets",
"CloudFrontDistribution": "aaaaaaaaaaaaa.cloudfront.net",
"Version": "20190128175402",
"Status": "ACTIVE",
"CustomDomainConfig": {}
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeUserPoolDomain](#)」の「」を参照してください。

describe-user-pool

次の例は、describe-user-pool を使用する方法を説明しています。

AWS CLI

ユーザープールを記述するには

この例では、ユーザープール ID us-west-2_aaaaaaaaa を持つユーザープールについて説明します。

コマンド:

```
aws cognito-idp describe-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

出力:

```
{
  "UserPool": {
    "SmsVerificationMessage": "Your verification code is {####}. ",
    "SchemaAttributes": [
      {
        "Name": "sub",
        "StringAttributeConstraints": {
          "MinLength": "1",
          "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": true,
        "AttributeDataType": "String",
        "Mutable": false
      }
    ]
  }
}
```

```
  },
  {
    "Name": "name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "given_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "family_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "middle_name",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  }
}
```

```
  },
  {
    "Name": "nickname",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "preferred_username",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "profile",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "picture",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  }
}
```

```
  },
  {
    "Name": "website",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "email",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": true,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "Name": "email_verified",
    "Mutable": true
  },
  {
    "Name": "gender",
    "StringAttributeConstraints": {
      "MinLength": "0",
      "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
  },
  {
    "Name": "birthdate",
    "StringAttributeConstraints": {
```

```
        "MinLength": "10",
        "MaxLength": "10"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "zoneinfo",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "locale",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "Name": "phone_number",
    "StringAttributeConstraints": {
        "MinLength": "0",
        "MaxLength": "2048"
    },
    "DeveloperOnlyAttribute": false,
    "Required": false,
    "AttributeDataType": "String",
    "Mutable": true
},
{
    "AttributeDataType": "Boolean",
    "DeveloperOnlyAttribute": false,
```

```

        "Required": false,
        "Name": "phone_number_verified",
        "Mutable": true
    },
    {
        "Name": "address",
        "StringAttributeConstraints": {
            "MinLength": "0",
            "MaxLength": "2048"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "String",
        "Mutable": true
    },
    {
        "Name": "updated_at",
        "NumberAttributeConstraints": {
            "MinValue": "0"
        },
        "DeveloperOnlyAttribute": false,
        "Required": false,
        "AttributeDataType": "Number",
        "Mutable": true
    }
],
"EmailVerificationSubject": "Your verification code",
"MfaConfiguration": "OFF",
"Name": "MyUserPool",
"EmailVerificationMessage": "Your verification code is {#####}. ",
"SmsAuthenticationMessage": "Your authentication code is {#####}. ",
"LastModifiedDate": 1547763720.822,
"AdminCreateUserConfig": {
    "InviteMessageTemplate": {
        "EmailMessage": "Your username is {username} and temporary password is
{#####}. ",
        "EmailSubject": "Your temporary password",
        "SMSMessage": "Your username is {username} and temporary password is
{#####}. "
    },
    "UnusedAccountValidityDays": 7,
    "AllowAdminCreateUserOnly": false
},
"EmailConfiguration": {

```



```
    "ReplyToEmailAddress": "myemail@mydomain.com"
    "SourceArn": "arn:aws:ses:us-east-1:000000000000:identity/
myemail@mydomain.com"
  },
  "AutoVerifiedAttributes": [
    "email"
  ],
  "Policies": {
    "PasswordPolicy": {
      "RequireLowercase": true,
      "RequireSymbols": true,
      "RequireNumbers": true,
      "MinimumLength": 8,
      "RequireUppercase": true
    }
  },
  "UserPoolTags": {},
  "UsernameAttributes": [
    "email"
  ],
  "CreationDate": 1547763720.822,
  "EstimatedNumberOfUsers": 1,
  "Id": "us-west-2_aaaaaaaaaa",
  "LambdaConfig": {}
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUserPool](#)」の「」を参照してください。

forget-device

次のコード例は、forget-device を使用する方法を示しています。

AWS CLI

デバイスを忘れるには

この例では、デバイスを忘れませう。

コマンド:

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ForgotDevice](#)」の「」を参照してください。

forgot-password

次のコード例は、forgot-password を使用する方法を示しています。

AWS CLI

パスワードの変更を強制するには

次のforgot-password例では、パスワードを変更するメッセージを jane@example.com に送信します。

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpbsnet7mpld0 --  
username jane@example.com
```

出力:

```
{  
  "CodeDeliveryDetails": {  
    "Destination": "j***@e***.com",  
    "DeliveryMedium": "EMAIL",  
    "AttributeName": "email"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ForgotPassword](#)」の「」を参照してください。

get-csv-header

次の例は、get-csv-header を使用する方法を説明しています。

AWS CLI

csv ヘッダーを作成するには

この例では、csv ヘッダーを作成します。

ユーザーのインポートの詳細については、CSV「ファイルからユーザープールにユーザーをインポートする」を参照してください。

コマンド:

```
aws cognito-idp get-csv-header --user-pool-id us-west-2_aaaaaaaa
```

出力:

```
{
  "UserPoolId": "us-west-2_aaaaaaaa",
  "CSVHeader": [
    "name",
    "given_name",
    "family_name",
    "middle_name",
    "nickname",
    "preferred_username",
    "profile",
    "picture",
    "website",
    "email",
    "email_verified",
    "gender",
    "birthdate",
    "zoneinfo",
    "locale",
    "phone_number",
    "phone_number_verified",
    "address",
    "updated_at",
    "cognito:mfa_enabled",
    "cognito:username"
  ]
}
```

... CSV ファイルからユーザープールへのユーザーのインポート : <https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-using-import-tool.html>

- API 詳細については、AWS CLI「コマンドリファレンス [GetCsvHeader](#)」の「」を参照してください。

get-group

次の例は、get-group を使用する方法を説明しています。

AWS CLI

グループに関する情報を取得するには

この例では、という名前のグループに関する情報を取得します MyGroup。

コマンド:

```
aws cognito-idp get-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

出力:

```
{
  "Group": {
    "GroupName": "MyGroup",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Description": "A sample group.",
    "LastModifiedDate": 1548270073.795,
    "CreationDate": 1548270073.795
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroup](#)」の「」を参照してください。

get-signing-certificate

次の例は、get-signing-certificate を使用する方法を説明しています。

AWS CLI

署名証明書を取得するには

この例では、ユーザープールの署名証明書を取得します。

コマンド:

```
aws cognito-idp get-signing-certificate --user-pool-id us-west-2_aaaaaaaaa
```

出力:

```
{
  "Certificate": "CERTIFICATE_DATA"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSigningCertificate](#)」の「」を参照してください。

get-ui-customization

次の例は、get-ui-customization を使用する方法を説明しています。

AWS CLI

UI カスタマイズ情報を取得するには

この例では、ユーザープールの UI カスタマイズ情報を取得します。

コマンド:

```
aws cognito-idp get-ui-customization --user-pool-id us-west-2_aaaaaaaaa
```

出力:

```
{
  "UICustomization": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientId": "ALL",
    "ImageUrl": "https://aaaaaaaaaaaaa.cloudfront.net/us-west-2_aaaaaaaaa/
ALL/20190128231240/assets/images/image.jpg",
    "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n\n.banner-customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color:
lightgray;\n}\n\n.label-customizable {\n\tfont-weight: 300;\n}\n\n.textDescription-
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n\n.legalText-
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n\n.submitButton-customizable
{\n\tfont-size: 14px;\n\tfont-weight: bold;\n\tmargin: 20px 0px 10px 0px;\n
\theight: 40px;\n\twidth: 100%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n}\n\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
#286090;\n}\n\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
```

```

\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
#D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
#555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n.inputField-
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-
customizable {\n\theight: 40px;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-
bottom: 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color:
#46b8da;\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
#31b0d5;\n}\n.socialButton-customizable {\n\theight: 40px;\n\ttext-align: left;
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #faf;\n}\n",
    "CSSVersion": "20190128231240"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUiCustomization](#)」の「」を参照してください。

list-user-import-jobs

次の例は、list-user-import-jobs を使用する方法を説明しています。

AWS CLI

ユーザーインポートジョブを一覧表示するには

この例では、ユーザーのインポートジョブを一覧表示します。

ユーザーのインポートの詳細については、CSV「ファイルからユーザープールにユーザーをインポートする」を参照してください。

コマンド:

```
aws cognito-idp list-user-import-jobs --user-pool-id us-west-2_aaaaaaaaaa --max-
results 20
```

出力:

```
{
  "UserImportJobs": [
    {
      "JobName": "Test2",
```

```

    "JobId": "import-d00nwGA3mV",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548272793.069,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  },
  {
    "JobName": "Test1",
    "JobId": "import-qQ0DCt2fRh",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548271795.471,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  },
  {
    "JobName": "import-Test1",
    "JobId": "import-TZqNQvDRnW",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548271708.512,
    "StartDate": 1548277247.962,
    "CompletionDate": 1548277248.912,
    "Status": "Failed",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 1,
    "CompletionMessage": "Too many users have failed or been skipped during
the import."
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUserImportJobs](#)」の「」を参照してください。

list-user-pools

次のコード例は、list-user-pools を使用する方法を示しています。

AWS CLI

ユーザープールを一覧表示するには

次の例では、最大 20 のユーザープールを一覧表示します。

コマンド:

```
aws cognito-idp list-user-pools --max-results 20
```

出力:

```
{
  "UserPools": [
    {
      "CreationDate": 1547763720.822,
      "LastModifiedDate": 1547763720.822,
      "LambdaConfig": {},
      "Id": "us-west-2_aaaaaaaaa",
      "Name": "MyUserPool"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUserPools](#)」の「」を参照してください。

list-users-in-group

次のコード例は、list-users-in-group を使用する方法を示しています。

AWS CLI

グループ内のユーザーを一覧表示するには

この例では、グループのユーザーを一覧表示します MyGroup。

コマンド:

```
aws cognito-idp list-users-in-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

出力:

```
{
  "Users": [
    {
      "Username": "acf10624-80bb-401a-ac61-607bee2110ec",
      "Attributes": [
        {
          "Name": "sub",
          "Value": "acf10624-80bb-401a-ac61-607bee2110ec"
        },
        {
          "Name": "custom:CustomAttr1",
          "Value": "New Value!"
        },
        {
          "Name": "email",
          "Value": "jane@example.com"
        }
      ],
      "UserCreateDate": 1548102770.284,
      "UserLastModifiedDate": 1548103204.893,
      "Enabled": true,
      "UserStatus": "CONFIRMED"
    },
    {
      "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
      "Attributes": [
        {
          "Name": "sub",
          "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
        },
        {
          "Name": "email_verified",
          "Value": "true"
        }
      ],
    }
  ]
}
```

```
    {
      "Name": "email",
      "Value": "diego@example.com"
    }
  ],
  "UserCreateDate": 1548089817.683,
  "UserLastModifiedDate": 1548089817.683,
  "Enabled": true,
  "UserStatus": "FORCE_CHANGE_PASSWORD"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUsersInGroup](#)」の「」を参照してください。

list-users

次のコード例は、list-users を使用する方法を示しています。

AWS CLI

ユーザーを一覧表示するには

この例では最大 20 のユーザーを一覧表示します。

コマンド:

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

出力:

```
{
  "Users": [
    {
      "Username": "22704aa3-fc10-479a-97eb-2af5806bd327",
      "Enabled": true,
      "UserStatus": "FORCE_CHANGE_PASSWORD",
      "UserCreateDate": 1548089817.683,
      "UserLastModifiedDate": 1548089817.683,
      "Attributes": [
```

```
    {
      "Name": "sub",
      "Value": "22704aa3-fc10-479a-97eb-2af5806bd327"
    },
    {
      "Name": "email_verified",
      "Value": "true"
    },
    {
      "Name": "email",
      "Value": "mary@example.com"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUsers](#)」の「」を参照してください。

resend-confirmation-code

次の例は、resend-confirmation-code を使用する方法を説明しています。

AWS CLI

確認コードを再送信するには

次の resend-confirmation-code 例では、ユーザー jane に確認コードを送信します。

```
aws cognito-idp resend-confirmation-code \
  --client-id 12a3b456c7de890f11g123hijk \
  --username jane
```

出力:

```
{
  "CodeDeliveryDetails": {
    "Destination": "j***@e***.com",
    "DeliveryMedium": "EMAIL",
    "AttributeName": "email"
  }
}
```

```
}
```

詳細については、「Amazon Cognito デベロッパーガイド」の「[ユーザーアカウントのサインアップと確認](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResendConfirmationCode](#)」の「」を参照してください。

respond-to-auth-challenge

次の例は、respond-to-auth-challenge を使用する方法を説明しています。

AWS CLI

認可チャレンジに応答するには

この例では initiate-auth で開始された認可チャレンジに応答します。これは NEW_PASSWORD_REQUIRED チャレンジに対するレスポンスです。ユーザー jane@example.com のパスワードを設定します。

コマンド:

```
aws cognito-idp respond-to-auth-challenge --client-id 3n4b5urk1ft4f13mg5e62d9ado  
--challenge-name NEW_PASSWORD_REQUIRED --challenge-responses  
USERNAME=jane@example.com,NEW_PASSWORD="password" --session "SESSION_TOKEN"
```

出力:

```
{  
  "ChallengeParameters": {},  
  "AuthenticationResult": {  
    "AccessToken": "ACCESS_TOKEN",  
    "ExpiresIn": 3600,  
    "TokenType": "Bearer",  
    "RefreshToken": "REFRESH_TOKEN",  
    "IdToken": "ID_TOKEN",  
    "NewDeviceMetadata": {  
      "DeviceKey": "us-west-2_fec070d2-fa88-424a-8ec8-b26d7198eb23",  
      "DeviceGroupKey": "-wt2ha1Zd"  
    }  
  }  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RespondToAuthChallenge](#)」の「」を参照してください。

set-risk-configuration

次の例は、set-risk-configuration を使用する方法を説明しています。

AWS CLI

リスク設定を設定するには

この例では、ユーザープールのリスク設定を設定します。サインアップイベントアクションを NO_ に設定します ACTION。

コマンド:

```
aws cognito-idp set-risk-configuration --user-pool-id us-west-2_aaaaaaaaa --  
compromised-credentials-risk-  
configuration EventFilter=SIGN_UP,Actions={EventAction=NO_ACTION}
```

出力:

```
{  
  "RiskConfiguration": {  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "CompromisedCredentialsRiskConfiguration": {  
      "EventFilter": [  
        "SIGN_UP"  
      ],  
      "Actions": {  
        "EventAction": "NO_ACTION"  
      }  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetRiskConfiguration](#)」の「」を参照してください。

set-ui-customization

次のコード例は、set-ui-customization を使用する方法を示しています。

AWS CLI

UI のカスタマイズを設定するには

この例では、ユーザープールCSSの設定をカスタマイズします。

コマンド:

```
aws cognito-idp set-ui-customization --user-pool-id us-west-2_aaaaaaaaa --
css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-
customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color: lightgray;
\n}\n.label-customizable {\n\tfont-weight: 300;\n}\n.textDescription-customizable
{\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-
size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n\tpadding-
bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-customizable
{\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
{\n\tfont-size: 14px;\n\tfont-weight: bold;\n\tmargin: 20px 0px 10px 0px;\n
\theight: 40px;\n\twidth: 100%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
#286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
#D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
#555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n.inputField-
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-
customizable {\n\theight: 40px;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-
bottom: 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color:
#46b8da;\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
#31b0d5;\n}\n.socialButton-customizable {\n\theight: 40px;\n\ttext-align: left;
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #faf;\n}\n"
```

出力:

```
{
  "UICustomization": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientId": "ALL",
```

```

    "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color:
  lightgray;\n}\n.label-customizable {\n\tfont-weight: 300;\n}\n.textDescription-
  customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
  \tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-
  customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
  {\n\tfont-size: 14px;\n\tfont-weight: bold;\n\tmargin: 20px 0px 10px 0px;\n
  \theight: 40px;\n\twidth: 100%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
  #286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
  #D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
  #555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n.inputField-
  customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-
  customizable {\n\theight: 40px;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-
  bottom: 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color:
  #46b8da;\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
  #31b0d5;\n}\n.socialButton-customizable {\n\theight: 40px;\n\ttext-align: left;
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n.redirect-customizable {\n\ttext-
  align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
  customizable {\n\tbackground-color: #faf;\n}\n",
    "CSSVersion": "20190129172214"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetUiCustomization](#)」の「」を参照してください。

set-user-mfa-preference

次の例は、set-user-mfa-preference を使用する方法を説明しています。

AWS CLI

ユーザーMFA設定を設定するには

次のset-user-mfa-preference例では、MFA配信オプションを変更します。MFA 配信媒体に変更しますSMS。

```
aws cognito-idp set-user-mfa-preference \
```

```
--access-token "eyJra12345EXAMPLE" \  
--software-token-mfa-settings Enabled=true,PreferredMfa=true \  
--sms-mfa-settings Enabled=false,PreferredMfa=false
```

このコマンドでは何も出力されません。

詳細については、Amazon Cognito デベロッパーガイド」の「[ユーザープールMFAへの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetUserMfaPreference](#)」の「」を参照してください。

set-user-settings

次の例は、set-user-settings を使用する方法を説明しています。

AWS CLI

ユーザー設定を設定するには

この例では、MFA配信設定を に設定しますEMAIL。

コマンド:

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-  
options DeliveryMedium=EMAIL
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetUserSettings](#)」の「」を参照してください。

sign-up

次の例は、sign-up を使用する方法を説明しています。

AWS CLI

ユーザーをサインアップするには

この例では jane@example.com をサインアップします。

コマンド:


```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4f13mg5e62d9ado --  
username jane@example.com --password PASSWORD --user-attributes  
Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

出力:

```
{  
  "UserConfirmed": false,  
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SignUp](#)」の「」を参照してください。

start-user-import-job

次のコード例は、start-user-import-job を使用する方法を示しています。

AWS CLI

ユーザーインポートジョブを開始するには

この例では、ユーザー入力ジョブを開始します。

ユーザーのインポートの詳細については、CSV「ファイルからユーザープールにユーザーをインポートする」を参照してください。

コマンド:

```
aws cognito-idp start-user-import-job --user-pool-id us-west-2_aaaaaaaaaa --job-  
id import-TZqNQvDRnW
```

出力:

```
{  
  "UserImportJob": {  
    "JobName": "import-Test10",  
    "JobId": "import-lmpxS0uIzH",  
    "UserPoolId": "us-west-2_aaaaaaaaaa",  
    "PreSignedUrl": "PRE_SIGNED_URL",
```

```

    "CreationDate": 1548278378.928,
    "StartDate": 1548278397.334,
    "Status": "Pending",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartUserImportJob](#)」の「」を参照してください。

stop-user-import-job

次のコード例は、stop-user-import-job を使用する方法を示しています。

AWS CLI

ユーザーインポートジョブを停止するには

この例では、ユーザー入カジョブを停止します。

ユーザーのインポートの詳細については、CSV「ファイルからユーザープールにユーザーをインポートする」を参照してください。

コマンド:

```
aws cognito-idp stop-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-id import-TZqNQvDRnW
```

出力:

```

{
  "UserImportJob": {
    "JobName": "import-Test5",
    "JobId": "import-Fx0kARISFL",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548278576.259,
  }
}

```

```
    "StartDate": 1548278623.366,  
    "CompletionDate": 1548278626.741,  
    "Status": "Stopped",  
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/  
CognitoCloudWatchLogsRole",  
    "ImportedUsers": 0,  
    "SkippedUsers": 0,  
    "FailedUsers": 0,  
    "CompletionMessage": "The Import Job was stopped by the developer."  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StopUserImportJob](#)」の「」を参照してください。

update-auth-event-feedback

次の例は、update-auth-event-feedback を使用する方法を説明しています。

AWS CLI

認証イベントフィードバックを更新するには

この例では、承認イベントフィードバックを更新します。イベント「Valid」をマークします。

コマンド:

```
aws cognito-idp update-auth-event-feedback --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com --event-id EVENT_ID --feedback-token FEEDBACK_TOKEN --  
feedback-value "Valid"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAuthEventFeedback](#)」の「」を参照してください。

update-device-status

次のコード例は、update-device-status を使用する方法を示しています。

AWS CLI

デバイスのステータスを更新するには

この例では、デバイスのステータスを「not_remembered」に更新します。

コマンド:

```
aws cognito-idp update-device-status --access-token ACCESS_TOKEN --device-key DEVICE_KEY --device-remembered-status "not_remembered"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDeviceStatus](#)」の「」を参照してください。

update-group

次の例は、update-group を使用する方法を説明しています。

AWS CLI

グループを更新するには

この例では、の説明と優先順位を更新します MyGroup。

コマンド:

```
aws cognito-idp update-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup --description "New description" --precedence 2
```

出力:

```
{
  "Group": {
    "GroupName": "MyGroup",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Description": "New description",
    "RoleArn": "arn:aws:iam::111111111111:role/MyRole",
    "Precedence": 2,
    "LastModifiedDate": 1548800862.812,
    "CreationDate": 1548097827.125
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateGroup](#)」の「」を参照してください。

update-resource-server

次のコード例は、`update-resource-server` を使用する方法を示しています。

AWS CLI

リソースサーバーを更新するには

この例では、リソースサーバーの天気を更新します。新しいスコープが追加されます。

コマンド:

```
aws cognito-idp update-resource-server --user-pool-id us-west-2_aaaaaaaaa
--identifier weather.example.com --name Weather --scopes
ScopeName=NewScope,ScopeDescription="New scope description"
```

出力:

```
{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Identifier": "weather.example.com",
    "Name": "Happy",
    "Scopes": [
      {
        "ScopeName": "NewScope",
        "ScopeDescription": "New scope description"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResourceServer](#)」の「」を参照してください。

update-user-attributes

次の例は、`update-user-attributes` を使用する方法を説明しています。

AWS CLI

ユーザー属性を更新するには

この例では、ユーザー属性「nickname」を更新します。

コマンド:

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes
Name="nickname",Value="Dan"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserAttributes](#)」の「」を参照してください。

update-user-pool-client

次の例は、update-user-pool-client を使用する方法を説明しています。

AWS CLI

ユーザープールクライアントを更新するには

この例では、ユーザープールクライアントの名前を更新します。また、書き込み可能な属性「nickname」も追加されます。

コマンド:

```
aws cognito-idp update-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --
client-id 3n4b5urk1ft4fl3mg5e62d9ado --client-name "NewClientName" --write-
attributes "nickname"
```

出力:

```
{
  "UserPoolClient": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientName": "NewClientName",
    "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
    "LastModifiedDate": 1548802761.334,
    "CreationDate": 1548178931.258,
    "RefreshTokenValidity": 30,
    "WriteAttributes": [
      "nickname"
    ],
    "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

```
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateUserPoolClient](#)」の「」を参照してください。

update-user-pool

次の例は、update-user-pool を使用する方法を説明しています。

AWS CLI

ユーザープールを更新するには

次のupdate-user-pool例では、使用可能な各設定オプションの構文例を使用してユーザープールを変更します。ユーザープールを更新するには、以前に設定したすべてのオプションを指定する必要があります。指定しないと、デフォルト値にリセットされます。

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_EXAMPLE \
  --policies PasswordPolicy=
  \{MinimumLength=6,RequireUppercase=true,RequireLowercase=true,RequireNumbers=true,RequireSym
  \
  --deletion-protection ACTIVE \
  --lambda-config PreSignUp="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-presignup-
function",PreTokenGeneration="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-pretoken-function" \
  --auto-verified-attributes "phone_number" "email" \
  --verification-message-template \{"SmsMessage\":"Your code is
#####"\,"EmailMessage\":"Your code is {#####}"\,"EmailSubject\":"Your
verification code"\,"EmailMessageByLink\":"Click {##here##} to verify
your email address.""\,"EmailSubjectByLink\":"Your verification link"\,
\DefaultEmailOption\:"CONFIRM_WITH_LINK\}\ \
  --sms-authentication-message "Your code is {#####}" \
  --user-attribute-update-settings
  AttributesRequireVerificationBeforeUpdate="email","phone_number" \
  --mfa-configuration "OPTIONAL" \
  --device-
configuration ChallengeRequiredOnNewDevice=true,DeviceOnlyRememberedOnUserPrompt=true
  \
  --email-configuration SourceArn="arn:aws:ses:us-
west-2:123456789012:identity/admin@example.com",ReplyToEmailAddress="amdin
```

```
+noreply@example.com",EmailSendingAccount=DEVELOPER,From="admin@amazon.com",ConfigurationSet=
configuration-set" \
  --sms-configuration SnsCallerArn="arn:aws:iam::123456789012:role/service-role/
SNS-SMS-Role",ExternalId="12345",SnsRegion="us-west-2" \
  --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\{"Welcome {username}. Your confirmation code is
{#####}\",EmailMessage=\{"Welcome {username}. Your confirmation code is
{#####}\",EmailSubject=\{"Welcome to MyMobileGame"\}\} \
  --user-pool-tags "Function"="MyMobileGame","Developers"="Berlin" \
  --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\{"Welcome {username}. Your confirmation code is
{#####}\",EmailMessage=\{"Welcome {username}. Your confirmation code is
{#####}\",EmailSubject=\{"Welcome to MyMobileGame"\}\} \
  --user-pool-add-ons AdvancedSecurityMode="AUDIT" \
  --account-recovery-setting RecoveryMechanisms=
\[\{\Priority=1,Name="verified_email"\},\{\Priority=2,Name="verified_phone_number"\}\]
```

このコマンドでは何も出力されません。

詳細については、Amazon Cognito デベロッパーガイド」の「[ユーザープール設定の更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserPool](#)」の「」を参照してください。

を使用した Amazon Comprehend の例 AWS CLI

次のコード例は、Amazon Comprehend AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-detect-dominant-language

次のコード例は、batch-detect-dominant-language を使用する方法を示しています。

AWS CLI

複数の入力テキストの主な言語を検出するには

次のbatch-detect-dominant-language例では、複数の入力テキストを分析し、それぞれの主要な言語を返します。事前トレーニング済みモデルの信頼スコアも、予測ごとに出力されます。

```
aws comprehend batch-detect-dominant-language \  
  --text-list "Physics is the natural science that involves the study of matter  
  and its motion and behavior through space and time, along with related concepts  
  such as energy and force."
```

出力:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "Languages": [  
        {  
          "LanguageCode": "en",  
          "Score": 0.9986501932144165  
        }  
      ]  
    }  
  ],  
  "ErrorList": []  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[主要言語](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDetectDominantLanguage](#)」の「」を参照してください。

batch-detect-entities

次の例は、batch-detect-entities を使用する方法を説明しています。

AWS CLI

複数の入力テキストからエンティティを検出するには

次のbatch-detect-entities例では、複数の入力テキストを分析し、それぞれの名前付きエンティティを返します。予測ごとに、事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend batch-detect-entities \  
  --language-code en \  
  --text-list "Dear Jane, Your AnyCompany Financial Services LLC credit card  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to  
Alice at AnySpa@example.com."
```

出力:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "Entities": [  
        {  
          "Score": 0.9985517859458923,  
          "Type": "PERSON",  
          "Text": "Jane",  
          "BeginOffset": 5,  
          "EndOffset": 9  
        },  
        {  
          "Score": 0.9767839312553406,  
          "Type": "ORGANIZATION",  
          "Text": "AnyCompany Financial Services, LLC",  
          "BeginOffset": 16,  
          "EndOffset": 50  
        },  
        {  
          "Score": 0.9856694936752319,  
          "Type": "OTHER",
```

```
        "Text": "1111-XXXX-1111-XXXX",
        "BeginOffset": 71,
        "EndOffset": 90
    },
    {
        "Score": 0.9652159810066223,
        "Type": "QUANTITY",
        "Text": ".53",
        "BeginOffset": 116,
        "EndOffset": 119
    },
    {
        "Score": 0.9986667037010193,
        "Type": "DATE",
        "Text": "July 31st",
        "BeginOffset": 135,
        "EndOffset": 144
    }
]
},
{
    "Index": 1,
    "Entities": [
        {
            "Score": 0.720084547996521,
            "Type": "ORGANIZATION",
            "Text": "Sunshine Spa",
            "BeginOffset": 33,
            "EndOffset": 45
        },
        {
            "Score": 0.9865870475769043,
            "Type": "LOCATION",
            "Text": "123 Main St",
            "BeginOffset": 47,
            "EndOffset": 58
        },
        {
            "Score": 0.5895616412162781,
            "Type": "LOCATION",
            "Text": "Anywhere",
            "BeginOffset": 60,
            "EndOffset": 68
        }
    ]
},
```

```
    {
      "Score": 0.6809214353561401,
      "Type": "PERSON",
      "Text": "Alice",
      "BeginOffset": 75,
      "EndOffset": 80
    },
    {
      "Score": 0.9979087114334106,
      "Type": "OTHER",
      "Text": "AnySpa@example.com",
      "BeginOffset": 84,
      "EndOffset": 99
    }
  ]
},
"ErrorList": []
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[\[Entities\] \(エンティティ\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDetectEntities](#)」の「」を参照してください。

batch-detect-key-phrases

次のコード例は、batch-detect-key-phrases を使用する方法を示しています。

AWS CLI

複数のテキスト入力のキーフレーズを検出するには

次のbatch-detect-key-phrases例では、複数の入力テキストを分析し、それぞれのキー名詞フレーズを返します。各予測の事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend batch-detect-key-phrases \
  --language-code en \
  --text-list "Hello Zhang Wei, I am John, writing to you about the trip for
next Saturday." "Dear Jane, Your AnyCompany Financial Services LLC credit card
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July
```

31st." "Please send customer feedback to Sunshine Spa, 123 Main St, Anywhere or to Alice at AnySpa@example.com."

出力:

```
{
  "ResultList": [
    {
      "Index": 0,
      "KeyPhrases": [
        {
          "Score": 0.99700927734375,
          "Text": "Zhang Wei",
          "BeginOffset": 6,
          "EndOffset": 15
        },
        {
          "Score": 0.9929308891296387,
          "Text": "John",
          "BeginOffset": 22,
          "EndOffset": 26
        },
        {
          "Score": 0.9997230172157288,
          "Text": "the trip",
          "BeginOffset": 49,
          "EndOffset": 57
        },
        {
          "Score": 0.9999470114707947,
          "Text": "next Saturday",
          "BeginOffset": 62,
          "EndOffset": 75
        }
      ]
    },
    {
      "Index": 1,
      "KeyPhrases": [
        {
          "Score": 0.8358274102210999,
          "Text": "Dear Jane",
          "BeginOffset": 0,
```

```
        "EndOffset": 9
      },
      {
        "Score": 0.989359974861145,
        "Text": "Your AnyCompany Financial Services",
        "BeginOffset": 11,
        "EndOffset": 45
      },
      {
        "Score": 0.8812323808670044,
        "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
        "BeginOffset": 47,
        "EndOffset": 90
      },
      {
        "Score": 0.9999381899833679,
        "Text": "a minimum payment",
        "BeginOffset": 95,
        "EndOffset": 112
      },
      {
        "Score": 0.9997439980506897,
        "Text": ".53",
        "BeginOffset": 116,
        "EndOffset": 119
      },
      {
        "Score": 0.996875524520874,
        "Text": "July 31st",
        "BeginOffset": 135,
        "EndOffset": 144
      }
    ]
  },
  {
    "Index": 2,
    "KeyPhrases": [
      {
        "Score": 0.9990295767784119,
        "Text": "customer feedback",
        "BeginOffset": 12,
        "EndOffset": 29
      },
      {
```

```
        "Score": 0.9994127750396729,  
        "Text": "Sunshine Spa",  
        "BeginOffset": 33,  
        "EndOffset": 45  
    },  
    {  
        "Score": 0.9892991185188293,  
        "Text": "123 Main St",  
        "BeginOffset": 47,  
        "EndOffset": 58  
    },  
    {  
        "Score": 0.9969810843467712,  
        "Text": "Alice",  
        "BeginOffset": 75,  
        "EndOffset": 80  
    },  
    {  
        "Score": 0.9703696370124817,  
        "Text": "AnySpa@example.com",  
        "BeginOffset": 84,  
        "EndOffset": 99  
    }  
  ]  
},  
"ErrorList": []  
}
```

詳細については、「Amazon Comprehend 開発者ガイド」の「[キーフレーズ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDetectKeyPhrases](#)」の「」を参照してください。

batch-detect-sentiment

次の例は、batch-detect-sentiment を使用する方法を説明しています。

AWS CLI

複数の入力テキストの一般的な感情を検出するには

次のbatch-detect-sentiment例では、複数の入力テキストを分析し、それぞれの一般的な感情 (POSITIVE、NEGATIVE、MIXED、または NEUTRAL) を返します。

```
aws comprehend batch-detect-sentiment \  
  --text-list "That movie was very boring, I can't believe it was over four hours long." "It is a beautiful day for hiking today." "My meal was okay, I'm excited to try other restaurants." \  
  --language-code en
```

出力:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "Sentiment": "NEGATIVE",  
      "SentimentScore": {  
        "Positive": 0.00011316669406369328,  
        "Negative": 0.9995445609092712,  
        "Neutral": 0.00014722718333359808,  
        "Mixed": 0.00019498742767609656  
      }  
    },  
    {  
      "Index": 1,  
      "Sentiment": "POSITIVE",  
      "SentimentScore": {  
        "Positive": 0.9981263279914856,  
        "Negative": 0.00015240783977787942,  
        "Neutral": 0.0013876151060685515,  
        "Mixed": 0.00033366199932061136  
      }  
    },  
    {  
      "Index": 2,  
      "Sentiment": "MIXED",  
      "SentimentScore": {  
        "Positive": 0.15930435061454773,  
        "Negative": 0.11471917480230331,  
        "Neutral": 0.26897063851356506,  
        "Mixed": 0.45700588822364807  
      }  
    }  
  ]  
}
```



```

    ],
    "ErrorList": []
  }

```

詳細については、Amazon Comprehend デベロッパーガイド」の「[感情](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDetectSentiment](#)」の「」を参照してください。

batch-detect-syntax

次の例は、batch-detect-syntax を使用する方法を説明しています。

AWS CLI

複数の入力テキスト内の単語の構文と発話の一部を調べるには

次のbatch-detect-syntax例では、複数の入力テキストの構文を分析し、音声のさまざまな部分を返します。予測ごとに、事前トレーニング済みモデルの信頼スコアも出力されます。

```

aws comprehend batch-detect-syntax \
  --text-list "It is a beautiful day." "Can you please pass the salt?" "Please pay
the bill before the 31st." \
  --language-code en

```

出力:

```

{
  "ResultList": [
    {
      "Index": 0,
      "SyntaxTokens": [
        {
          "TokenId": 1,
          "Text": "It",
          "BeginOffset": 0,
          "EndOffset": 2,
          "PartOfSpeech": {
            "Tag": "PRON",
            "Score": 0.9999740719795227
          }
        }
      ],
    },
    {

```

```
    "TokenId": 2,
    "Text": "is",
    "BeginOffset": 3,
    "EndOffset": 5,
    "PartOfSpeech": {
      "Tag": "VERB",
      "Score": 0.999937117099762
    }
  },
  {
    "TokenId": 3,
    "Text": "a",
    "BeginOffset": 6,
    "EndOffset": 7,
    "PartOfSpeech": {
      "Tag": "DET",
      "Score": 0.9999926686286926
    }
  },
  {
    "TokenId": 4,
    "Text": "beautiful",
    "BeginOffset": 8,
    "EndOffset": 17,
    "PartOfSpeech": {
      "Tag": "ADJ",
      "Score": 0.9987891912460327
    }
  },
  {
    "TokenId": 5,
    "Text": "day",
    "BeginOffset": 18,
    "EndOffset": 21,
    "PartOfSpeech": {
      "Tag": "NOUN",
      "Score": 0.9999778866767883
    }
  },
  {
    "TokenId": 6,
    "Text": ".",
    "BeginOffset": 21,
    "EndOffset": 22,
```

```
        "PartOfSpeech": {
            "Tag": "PUNCT",
            "Score": 0.9999974966049194
        }
    ]
},
{
    "Index": 1,
    "SyntaxTokens": [
        {
            "TokenId": 1,
            "Text": "Can",
            "BeginOffset": 0,
            "EndOffset": 3,
            "PartOfSpeech": {
                "Tag": "AUX",
                "Score": 0.9999770522117615
            }
        },
        {
            "TokenId": 2,
            "Text": "you",
            "BeginOffset": 4,
            "EndOffset": 7,
            "PartOfSpeech": {
                "Tag": "PRON",
                "Score": 0.9999986886978149
            }
        },
        {
            "TokenId": 3,
            "Text": "please",
            "BeginOffset": 8,
            "EndOffset": 14,
            "PartOfSpeech": {
                "Tag": "INTJ",
                "Score": 0.9681622385978699
            }
        },
        {
            "TokenId": 4,
            "Text": "pass",
            "BeginOffset": 15,
```

```
        "EndOffset": 19,
        "PartOfSpeech": {
            "Tag": "VERB",
            "Score": 0.9999874830245972
        }
    },
    {
        "TokenId": 5,
        "Text": "the",
        "BeginOffset": 20,
        "EndOffset": 23,
        "PartOfSpeech": {
            "Tag": "DET",
            "Score": 0.9999827146530151
        }
    },
    {
        "TokenId": 6,
        "Text": "salt",
        "BeginOffset": 24,
        "EndOffset": 28,
        "PartOfSpeech": {
            "Tag": "NOUN",
            "Score": 0.9995040893554688
        }
    },
    {
        "TokenId": 7,
        "Text": "?",
        "BeginOffset": 28,
        "EndOffset": 29,
        "PartOfSpeech": {
            "Tag": "PUNCT",
            "Score": 0.999998152256012
        }
    }
]
},
{
    "Index": 2,
    "SyntaxTokens": [
        {
            "TokenId": 1,
            "Text": "Please",
```

```
        "BeginOffset": 0,  
        "EndOffset": 6,  
        "PartOfSpeech": {  
            "Tag": "INTJ",  
            "Score": 0.9997857809066772  
        }  
    },  
    {  
        "TokenId": 2,  
        "Text": "pay",  
        "BeginOffset": 7,  
        "EndOffset": 10,  
        "PartOfSpeech": {  
            "Tag": "VERB",  
            "Score": 0.9999252557754517  
        }  
    },  
    {  
        "TokenId": 3,  
        "Text": "the",  
        "BeginOffset": 11,  
        "EndOffset": 14,  
        "PartOfSpeech": {  
            "Tag": "DET",  
            "Score": 0.9999842643737793  
        }  
    },  
    {  
        "TokenId": 4,  
        "Text": "bill",  
        "BeginOffset": 15,  
        "EndOffset": 19,  
        "PartOfSpeech": {  
            "Tag": "NOUN",  
            "Score": 0.9999588131904602  
        }  
    },  
    {  
        "TokenId": 5,  
        "Text": "before",  
        "BeginOffset": 20,  
        "EndOffset": 26,  
        "PartOfSpeech": {  
            "Tag": "ADP",
```

```
        "Score": 0.9958304762840271
      }
    },
    {
      "TokenId": 6,
      "Text": "the",
      "BeginOffset": 27,
      "EndOffset": 30,
      "PartOfSpeech": {
        "Tag": "DET",
        "Score": 0.9999947547912598
      }
    },
    {
      "TokenId": 7,
      "Text": "31st",
      "BeginOffset": 31,
      "EndOffset": 35,
      "PartOfSpeech": {
        "Tag": "NOUN",
        "Score": 0.9924124479293823
      }
    },
    {
      "TokenId": 8,
      "Text": ".",
      "BeginOffset": 35,
      "EndOffset": 36,
      "PartOfSpeech": {
        "Tag": "PUNCT",
        "Score": 0.9999955892562866
      }
    }
  ]
},
"ErrorList": []
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[構文分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDetectSyntax](#)」の「」を参照してください。

batch-detect-targeted-sentiment

次の例は、batch-detect-targeted-sentiment を使用する方法を説明しています。

AWS CLI

複数の入力テキストの感情と各名前付きエンティティを検出するには

次のbatch-detect-targeted-sentiment例では、複数の入力テキストを分析し、名前付きエンティティと各エンティティにアタッチされた一般的な感情を返します。予測ごとに、事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend batch-detect-targeted-sentiment \  
  --language-code en \  
  --text-list "That movie was really boring, the original was way more  
entertaining" "The trail is extra beautiful today." "My meal was just okay."
```

出力:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "Entities": [  
        {  
          "DescriptiveMentionIndex": [  
            0  
          ],  
          "Mentions": [  
            {  
              "Score": 0.9999009966850281,  
              "GroupScore": 1.0,  
              "Text": "movie",  
              "Type": "MOVIE",  
              "MentionSentiment": {  
                "Sentiment": "NEGATIVE",  
                "SentimentScore": {  
                  "Positive": 0.13887299597263336,  
                  "Negative": 0.8057460188865662,  
                  "Neutral": 0.05525200068950653,  
                  "Mixed": 0.00012799999967683107  
                }  
              }  
            }  
          ],  
        }  
      ],  
    }  
  ],  
}
```

```
        "BeginOffset": 5,
        "EndOffset": 10
      }
    ]
  },
  {
    "DescriptiveMentionIndex": [
      0
    ],
    "Mentions": [
      {
        "Score": 0.9921110272407532,
        "GroupScore": 1.0,
        "Text": "original",
        "Type": "MOVIE",
        "MentionSentiment": {
          "Sentiment": "POSITIVE",
          "SentimentScore": {
            "Positive": 0.9999989867210388,
            "Negative": 9.99999974752427e-07,
            "Neutral": 0.0,
            "Mixed": 0.0
          }
        }
      },
      {
        "BeginOffset": 34,
        "EndOffset": 42
      }
    ]
  }
]
},
{
  "Index": 1,
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "Score": 0.7545599937438965,
          "GroupScore": 1.0,
          "Text": "trail",
          "Type": "OTHER",
```



```

        "MentionSentiment": {
            "Sentiment": "POSITIVE",
            "SentimentScore": {
                "Positive": 1.0,
                "Negative": 0.0,
                "Neutral": 0.0,
                "Mixed": 0.0
            }
        },
        "BeginOffset": 4,
        "EndOffset": 9
    }
]
},
{
    "DescriptiveMentionIndex": [
        0
    ],
    "Mentions": [
        {
            "Score": 0.9999960064888,
            "GroupScore": 1.0,
            "Text": "today",
            "Type": "DATE",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                    "Positive": 9.000000318337698e-06,
                    "Negative": 1.9999999949504854e-06,
                    "Neutral": 0.9999859929084778,
                    "Mixed": 3.999999989900971e-06
                }
            }
        },
        {
            "BeginOffset": 29,
            "EndOffset": 34
        }
    ]
}
]
},
{
    "Index": 2,
    "Entities": [
        {

```

```
"DescriptiveMentionIndex": [
  0
],
"Mentions": [
  {
    "Score": 0.9999880194664001,
    "GroupScore": 1.0,
    "Text": "My",
    "Type": "PERSON",
    "MentionSentiment": {
      "Sentiment": "NEUTRAL",
      "SentimentScore": {
        "Positive": 0.0,
        "Negative": 0.0,
        "Neutral": 1.0,
        "Mixed": 0.0
      }
    }
  },
  "BeginOffset": 0,
  "EndOffset": 2
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9995260238647461,
      "GroupScore": 1.0,
      "Text": "meal",
      "Type": "OTHER",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Positive": 0.04695599898695946,
          "Negative": 0.003226999891921878,
          "Neutral": 0.6091709733009338,
          "Mixed": 0.34064599871635437
        }
      }
    },
    "BeginOffset": 3,
    "EndOffset": 7
  ]
}
```

```

    }
  ]
}
],
"ErrorList": []
}

```

詳細については、[「Amazon Comprehend デベロッパーガイド」の「ターゲット感情」](#)を参照してください。 Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDetectTargetedSentiment](#)」の「」を参照してください。

classify-document

次の例は、classify-document を使用する方法を説明しています。

AWS CLI

モデル固有のエンドポイントを使用してドキュメントを分類するには

次のclassify-document例では、カスタムモデルのエンドポイントを使用してドキュメントを分類します。この例のモデルは、スパムまたは非スパム、または「ham」としてラベル付けされた SMS メッセージを含むデータセットでトレーニングされました。

```

aws comprehend classify-document \
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint \
  --text "CONGRATULATIONS! TXT 1235550100 to win $5000"

```

出力:

```

{
  "Classes": [
    {
      "Name": "spam",
      "Score": 0.9998599290847778
    },
    {

```

```
        "Name": "ham",
        "Score": 0.00014001205272506922
      }
    ]
  }
```

詳細については、「Amazon Comprehend 開発者ガイド」の「[カスタム分類](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ClassifyDocument](#)」の「」を参照してください。

contains-pii-entities

次のコード例は、contains-pii-entities を使用する方法を示しています。

AWS CLI

PII情報の存在について入力テキストを分析するには

次のcontains-pii-entities例では、入力テキストを分析して個人を特定できる情報 (PII) の有無を確認し、名前、住所、銀行口座番号、電話番号など、識別されたPIIエンティティタイプのラベルを返します。

```
aws comprehend contains-pii-entities \
  --language-code en \
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
  credit card
  account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
  July 31st. Based on your autopay settings,
  we will withdraw your payment on the due date from your bank account number
  XXXXXX1111 with the routing number XXXXX0000.
  Customer feedback for Sunshine Spa, 100 Main St, Anywhere. Send comments to
  Alice at AnySpa@example.com."
```

出力:

```
{
  "Labels": [
    {
      "Name": "NAME",
      "Score": 1.0
    }
  ]
}
```

```
    },
    {
      "Name": "EMAIL",
      "Score": 1.0
    },
    {
      "Name": "BANK_ACCOUNT_NUMBER",
      "Score": 0.9995794296264648
    },
    {
      "Name": "BANK_ROUTING",
      "Score": 0.9173126816749573
    },
    {
      "Name": "CREDIT_DEBIT_NUMBER",
      "Score": 1.0
    }
  }
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[個人を特定できる情報 \(PII\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ContainsPiiEntities](#)」の「」を参照してください。

create-dataset

次のコード例は、create-dataset を使用する方法を示しています。

AWS CLI

フライホイールデータセットを作成するには

次のcreate-dataset例では、フライホイールのデータセットを作成します。このデータセットは、--dataset-type タグで指定された追加のトレーニングデータとして使用されます。

```
aws comprehend create-dataset \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity \  
  --dataset-name example-dataset \  
  --dataset-type "TRAIN" \  
  --input-data-config file://inputConfig.json
```

file://inputConfig.json の内容:

```
{
  "DataFormat": "COMPREHEND_CSV",
  "DocumentClassifierInputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/training-data.csv"
  }
}
```

出力:

```
{
  "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity/dataset/example-dataset"
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[Flywheel Overview](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDataset](#)」の「」を参照してください。

create-document-classifier

次の例は、create-document-classifier を使用する方法を説明しています。

AWS CLI

ドキュメントを分類するドキュメント分類子を作成するには

次の create-document-classifier の例では、ドキュメント分類子モデルのトレーニングプロセスを開始します。トレーニングデータファイル「training.csv」は、--input-data-config タグにあります。training.csv は 2 列のドキュメントで、1 番目の列にはラベルまたは分類が、2 番目の列にはドキュメントが表示されます。

```
aws comprehend create-document-classifier \
  --document-classifier-name example-classifier \
  --data-access-arn arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/123456abcdeb0e11022f22a11EXAMPLE \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
  --language-code en
```

出力:

```
{
  "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

詳細については、「Amazon Comprehend 開発者ガイド」の「[カスタム分類](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDocumentClassifier](#)」の「」を参照してください。

create-endpoint

次のコード例は、create-endpoint を使用する方法を示しています。

AWS CLI

カスタムモデルのエンドポイントを作成するには

次のcreate-endpoint例では、以前にトレーニングされたカスタムモデルの同期推論用のエンドポイントを作成します。

```
aws comprehend create-endpoint \
  --endpoint-name example-classifier-endpoint-1 \
  --model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier \
  --desired-inference-units 1
```

出力:

```
{
  "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint-1"
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEndpoint](#)」の「」を参照してください。

create-entity-recognizer

次の例は、create-entity-recognizer を使用する方法を説明しています。

AWS CLI

カスタムエンティティレコグナイザーを作成するには

次のcreate-entity-recognizer例では、カスタムエンティティレコグナイザーモデルのトレーニングプロセスを開始します。この例では、トレーニングドキュメント、raw_text.csv、CSVエンティティリストを含むCSVファイルを使用してモデルをentity_list.csvトレーニングします。には、テキストとタイプという列entity-list.csvが含まれます。

```
aws comprehend create-entity-recognizer \  
  --recognizer-name example-entity-recognizer \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --input-data-config "EntityTypes=[{Type=DEVICE}], Documents={S3Uri=s3://DOC-  
EXAMPLE-BUCKET/trainingdata/raw_text.csv}, EntityList={S3Uri=s3://DOC-EXAMPLE-BUCKET/  
trainingdata/entity_list.csv}" \  
  --language-code en
```

出力:

```
{  
  "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:example-  
entity-recognizer/entityrecognizer1"  
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[カスタムエンティティ認識](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEntityRecognizer](#)」の「」を参照してください。

create-flywheel

次のコード例は、create-flywheel を使用する方法を示しています。

AWS CLI

フライホイールを作成するには

次のcreate-flywheel例では、ドキュメント分類またはエンティティ認識モデルの継続的なトレーニングを調整するフライホイールを作成します。この例のフライホイールは、--active-model-arn タグで指定された既存のトレーニング済みモデルを管理するために作成されます。フライホイールが作成されると、--input-data-lake タグにデータレイクが作成されます。

```
aws comprehend create-flywheel \  
  --flywheel-name example-flywheel \  
  --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-model/version/1 \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role \  
  --data-lake-s3-uri "s3://DOC-EXAMPLE-BUCKET"
```

出力:

```
{  
  "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel"  
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[Flywheel Overview](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFlywheel](#)」の「」を参照してください。

delete-document-classifier

次の例は、delete-document-classifier を使用する方法を説明しています。

AWS CLI

カスタムドキュメント分類子を削除するには

次の delete-document-classifier の例では、カスタムドキュメント分類子モデルを削除します。

```
aws comprehend delete-document-classifier \  
  --document-classifier-name example-classifier
```

```
--document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDocumentClassifier](#)」の「」を参照してください。

delete-endpoint

次の例は、delete-endpoint を使用する方法を説明しています。

AWS CLI

カスタムモデルのエンドポイントを削除するには

次のdelete-endpoint例では、モデル固有のエンドポイントを削除します。モデルを削除するには、すべてのエンドポイントを削除する必要があります。

```
aws comprehend delete-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint-1
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEndpoint](#)」の「」を参照してください。

delete-entity-recognizer

次の例は、delete-entity-recognizer を使用する方法を説明しています。

AWS CLI

カスタムエンティティレコグナイザーモデルを削除するには

次のdelete-entity-recognizer例では、カスタムエンティティレコグナイザーモデルを削除します。

```
aws comprehend delete-entity-recognizer \  
  --entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/example-entity-recognizer-1
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

- API 詳細については、「コマンドリファレンス[DeleteEntityRecognizer](#)」の「」を参照してください。AWS CLI

delete-flywheel

次のコード例は、delete-flywheel を使用する方法を示しています。

AWS CLI

フライホイールを削除するには

次のdelete-flywheel例では、フライホイールを削除します。データレイクまたはフライホイールに関連付けられたモデルは削除されません。

```
aws comprehend delete-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Flywheel 概要](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFlywheel](#)」の「」を参照してください。

delete-resource-policy

次のコード例は、delete-resource-policy を使用する方法を示しています。

AWS CLI

リソースベースのポリシーを削除するには

次のdelete-resource-policy例では、Amazon Comprehend リソースからリソースベースのポリシーを削除します。

```
aws comprehend delete-resource-policy \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier-1/version/1
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Comprehend デベロッパーガイド」の AWS 「アカウント間のカスタムモデルのコピー」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResourcePolicy](#)」の「」を参照してください。

describe-dataset

次の例は、describe-dataset を使用する方法を説明しています。

AWS CLI

フライホイールデータセットを記述するには

次のdescribe-dataset例では、フライホイールデータセットのプロパティを取得します。

```
aws comprehend describe-dataset \  
  --dataset-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset
```

出力:

```
{  
  "DatasetProperties": {  
    "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset",  
    "DatasetName": "example-dataset",  
    "DatasetType": "TRAIN",
```

```
"DatasetS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/
schemaVersion=1/12345678A123456Z/datasets/example-dataset/20230616T203710Z/",
  "Status": "CREATING",
  "CreationTime": "2023-06-16T20:37:10.400000+00:00"
}
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[Flywheel Overview](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDataset](#)」の「」を参照してください。

describe-document-classification-job

次の例は、describe-document-classification-job を使用する方法を説明しています。

AWS CLI

ドキュメント分類ジョブを記述するには

次の describe-document-classification-job の例では、非同期ドキュメント分類ジョブのプロパティを取得します。

```
aws comprehend describe-document-classification-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "DocumentClassificationJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "exampleclassificationjob",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
    "EndTime": "2023-06-14T17:15:58.582000+00:00",
    "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/mymodel/version/1",
    "InputDataConfig": {
```

```

        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
        "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-
CLN-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}

```

詳細については、「Amazon Comprehend 開発者ガイド」の「[カスタム分類](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDocumentClassificationJob](#)」の「」を参照してください。

describe-document-classifier

次のコード例は、describe-document-classifier を使用する方法を示しています。

AWS CLI

ドキュメント分類子を記述するには

次の describe-document-classifier の例では、カスタムドキュメント分類子モデルのプロパティを取得します。

```

aws comprehend describe-document-classifier \
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier-1

```

出力:

```

{
  "DocumentClassifierProperties": {
    "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier-1",
    "LanguageCode": "en",
    "Status": "TRAINED",
    "SubmitTime": "2023-06-13T19:04:15.735000+00:00",

```

```
"EndTime": "2023-06-13T19:42:31.752000+00:00",
"TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
"TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
"InputDataConfig": {
  "DataFormat": "COMPREHEND_CSV",
  "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
},
"OutputDataConfig": {},
"ClassifierMetadata": {
  "NumberOfLabels": 3,
  "NumberOfTrainedDocuments": 5016,
  "NumberOfTestDocuments": 557,
  "EvaluationMetrics": {
    "Accuracy": 0.9856,
    "Precision": 0.9919,
    "Recall": 0.9459,
    "F1Score": 0.9673,
    "MicroPrecision": 0.9856,
    "MicroRecall": 0.9856,
    "MicroF1Score": 0.9856,
    "HammingLoss": 0.0144
  }
},
"DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role",
"Mode": "MULTI_CLASS"
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[カスタムモデルの作成と管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDocumentClassifier](#)」の「」を参照してください。

describe-dominant-language-detection-job

次の例は、describe-dominant-language-detection-job を使用する方法を説明しています。

AWS CLI

主要な言語検出検出ジョブを記述します。

次のdescribe-dominant-language-detection-job例では、非同期優勢言語検出ジョブのプロパティを取得します。

```
aws comprehend describe-dominant-language-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{  
  "DominantLanguageDetectionJobProperties": {  
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-  
detection-job/123456abcdeb0e11022f22a11EXAMPLE",  
    "JobName": "languageanalysis1",  
    "JobStatus": "IN_PROGRESS",  
    "SubmitTime": "2023-06-09T18:10:38.037000+00:00",  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-  
LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"  
    },  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role"  
  }  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDominantLanguageDetectionJob](#)」の「」を参照してください。

describe-endpoint

次の例は、describe-endpoint を使用する方法を説明しています。

AWS CLI

特定のエンドポイントを記述するには

次のdescribe-endpoint例では、モデル固有のエンドポイントのプロパティを取得します。

```
aws comprehend describe-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint
```

出力:

```
{  
  "EndpointProperties": {  
    "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier-endpoint/example-classifier-endpoint",  
    "Status": "IN_SERVICE",  
    "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
exampleclassifier1",  
    "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/exampleclassifier1",  
    "DesiredInferenceUnits": 1,  
    "CurrentInferenceUnits": 1,  
    "CreationTime": "2023-06-13T20:32:54.526000+00:00",  
    "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"  
  }  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEndpoint](#)」の「」を参照してください。

describe-entities-detection-job

次の例は、describe-entities-detection-job を使用する方法を説明しています。

AWS CLI

エンティティ検出ジョブを記述するには

次のdescribe-entities-detection-job例では、非同期エンティティ検出ジョブのプロパティを取得します。

```
aws comprehend describe-entities-detection-job \  
  --job-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
exampleclassifier1
```

```
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "EntitiesDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "example-entity-detector",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
    "EndTime": "2023-06-08T21:40:23.509000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/thefolder/111122223333-
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEntitiesDetectionJob](#)」の「」を参照してください。

describe-entity-recognizer

次の例は、describe-entity-recognizer を使用する方法を説明しています。

AWS CLI

エンティティレコグナイザーを記述するには

次のdescribe-entity-recognizer例では、カスタムエンティティレコグナイザーモデルのプロパティを取得します。

```
aws comprehend describe-entity-recognizer \  
  entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-  
recognizer/business-recongizer-1/version/1
```

出力:

```
{  
  "EntityRecognizerProperties": {  
    "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:entity-  
recognizer/business-recongizer-1/version/1",  
    "LanguageCode": "en",  
    "Status": "TRAINED",  
    "SubmitTime": "2023-06-14T20:44:59.631000+00:00",  
    "EndTime": "2023-06-14T20:59:19.532000+00:00",  
    "TrainingStartTime": "2023-06-14T20:48:52.811000+00:00",  
    "TrainingEndTime": "2023-06-14T20:58:11.473000+00:00",  
    "InputDataConfig": {  
      "DataFormat": "COMPREHEND_CSV",  
      "EntityTypes": [  
        {  
          "Type": "BUSINESS"  
        }  
      ],  
      "Documents": {  
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/dataset/",  
        "InputFormat": "ONE_DOC_PER_LINE"  
      },  
      "EntityList": {  
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/entity.csv"  
      }  
    },  
    "RecognizerMetadata": {  
      "NumberOfTrainedDocuments": 1814,  
      "NumberOfTestDocuments": 486,  
      "EvaluationMetrics": {  
        "Precision": 100.0,  
        "Recall": 100.0,  
        "F1Score": 100.0  
      },  
      "EntityTypes": [  
        {  
          "Type": "BUSINESS",  
          "EvaluationMetrics": {
```

```
        "Precision": 100.0,
        "Recall": 100.0,
        "F1Score": 100.0
      },
      "NumberOfTrainMentions": 1520
    ]
  ],
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role",
  "VersionName": "1"
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[カスタムエンティティ認識](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEntityRecognizer](#)」の「」を参照してください。

describe-events-detection-job

次の例は、describe-events-detection-job を使用する方法を説明しています。

AWS CLI

イベント検出ジョブを記述します。

次のdescribe-events-detection-job例では、非同期イベント検出ジョブのプロパティを取得します。

```
aws comprehend describe-events-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "EventsDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
```

```
"JobName": "events_job_1",
"JobStatus": "IN_PROGRESS",
"SubmitTime": "2023-06-12T18:45:56.054000+00:00",
"InputDataConfig": {
  "S3Uri": "s3://DOC-EXAMPLE-BUCKET/EventsData",
  "InputFormat": "ONE_DOC_PER_LINE"
},
"OutputDataConfig": {
  "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-
EVENTS-123456abcdeb0e11022f22a11EXAMPLE/output/"
},
"LanguageCode": "en",
"DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
"TargetEventTypes": [
  "BANKRUPTCY",
  "EMPLOYMENT",
  "CORPORATE_ACQUISITION",
  "CORPORATE_MERGER",
  "INVESTMENT_GENERAL"
]
}
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEventsDetectionJob](#)」の「」を参照してください。

describe-flywheel-iteration

次のコード例は、describe-flywheel-iteration を使用する方法を示しています。

AWS CLI

フライホイールの反復を記述するには

次のdescribe-flywheel-iteration例では、フライホイール反復のプロパティを取得します。

```
aws comprehend describe-flywheel-iteration \
```

```
--flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel \  
--flywheel-iteration-id 20232222AEXAMPLE
```

出力:

```
{  
  "FlywheelIterationProperties": {  
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity",  
    "FlywheelIterationId": "20232222AEXAMPLE",  
    "CreationTime": "2023-06-16T21:10:26.385000+00:00",  
    "EndTime": "2023-06-16T23:33:16.827000+00:00",  
    "Status": "COMPLETED",  
    "Message": "FULL_ITERATION: Flywheel iteration performed all functions successfully.",  
    "EvaluatedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/1",  
    "EvaluatedModelMetrics": {  
      "AverageF1Score": 0.7742663922375772,  
      "AveragePrecision": 0.8287636394041166,  
      "AverageRecall": 0.7427084833645399,  
      "AverageAccuracy": 0.8795394154118689  
    },  
    "TrainedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/Comprehend-Generated-v1-bb52d585",  
    "TrainedModelMetrics": {  
      "AverageF1Score": 0.9767700253081214,  
      "AveragePrecision": 0.9767700253081214,  
      "AverageRecall": 0.9767700253081214,  
      "AverageAccuracy": 0.9858281665190434  
    },  
    "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/flywheel-entity/schemaVersion=1/20230616T200543Z/evaluation/20230616T211026Z/"  
  }  
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[Flywheel 概要](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFlywheelIteration](#)」の「」を参照してください。

describe-flywheel

次の例は、describe-flywheel を使用する方法を説明しています。

AWS CLI

フライホイールを記述するには

次のdescribe-flywheel例では、フライホイールのプロパティを取得します。この例では、フライホイールに関連付けられたモデルは、ドキュメントをスパムまたは非スパム、または「ham」として分類するようにトレーニングされたカスタム分類子モデルです。

```
aws comprehend describe-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel
```

出力:

```
{  
  "FlywheelProperties": {  
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel",  
    "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-model/version/1",  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role",  
    "TaskConfig": {  
      "LanguageCode": "en",  
      "DocumentClassificationConfig": {  
        "Mode": "MULTI_CLASS",  
        "Labels": [  
          "ham",  
          "spam"  
        ]  
      }  
    },  
    "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/example-flywheel/  
schemaVersion=1/20230616T200543Z/",  
    "DataSecurityConfig": {},  
    "Status": "ACTIVE",  
    "ModelType": "DOCUMENT_CLASSIFIER",  
    "CreationTime": "2023-06-16T20:05:43.242000+00:00",  
    "LastModifiedTime": "2023-06-16T20:21:43.567000+00:00"  
  }  
}
```

```
}  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Flywheel Overview](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFlywheel](#)」の「」を参照してください。

describe-key-phrases-detection-job

次の例は、describe-key-phrases-detection-job を使用する方法を説明しています。

AWS CLI

キーフレーズ検出ジョブを記述するには

次のdescribe-key-phrases-detection-job例では、非同期キーフレーズ検出ジョブのプロパティを取得します。

```
aws comprehend describe-key-phrases-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{  
  "KeyPhrasesDetectionJobProperties": {  
    "JobId": "69aa080c00fc68934a6a98f10EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-  
job/69aa080c00fc68934a6a98f10EXAMPLE",  
    "JobName": "example-key-phrases-detection-job",  
    "JobStatus": "COMPLETED",  
    "SubmitTime": 1686606439.177,  
    "EndTime": 1686606806.157,  
    "InputDataConfig": {  
      "S3Uri": "s3://dereksbucket1001/EventsData/",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://dereksbucket1002/testfolder/111122223333-  
KP-69aa080c00fc68934a6a98f10EXAMPLE/output/output.tar.gz"  
    }  
  }  
}
```



```
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testrole"
  }
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeKeyPhrasesDetectionJob](#)」の「」を参照してください。

describe-pii-entities-detection-job

次の例は、describe-pii-entities-detection-job を使用する方法を説明しています。

AWS CLI

PIIエンティティ検出ジョブを記述するには

次のdescribe-pii-entities-detection-job例では、非同期 pii エンティティ検出ジョブのプロパティを取得します。

```
aws comprehend describe-pii-entities-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "PiiEntitiesDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "example-pii-entities-job",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
    "EndTime": "2023-06-08T21:40:23.509000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    }
  },
}
```

```
"OutputDataConfig": {
  "S3Uri": "s3://DOC-EXAMPLE-BUCKET/thefolder/111122223333-
NER-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
},
"LanguageCode": "en",
"DataAccessRoleArn": "arn:aws:iam::12345678012:role/service-role/
AmazonComprehendServiceRole-example-role"
}
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePiiEntitiesDetectionJob](#)」の「」を参照してください。

describe-resource-policy

次のコード例は、describe-resource-policy を使用する方法を示しています。

AWS CLI

モデルにアタッチされたリソースポリシーを記述するには

次のdescribe-resource-policy例では、モデルにアタッチされたリソースベースのポリシーのプロパティを取得します。

```
aws comprehend describe-resource-policy \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

出力:

```
{
  "ResourcePolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::444455556666:root\"},\"Action\":
\"comprehend:ImportModel\",\"Resource\":\"*\"}]}",
  "CreationTime": "2023-06-19T18:44:26.028000+00:00",
  "LastModifiedTime": "2023-06-19T18:53:02.002000+00:00",
  "PolicyRevisionId": "baa675d069d07afaa2aa3106ae280f61"
}
```

詳細については、[「Amazon Comprehend デベロッパーガイド」の AWS 「アカウント間のカスタムモデルのコピー」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeResourcePolicy](#)」の「」を参照してください。

describe-sentiment-detection-job

次のコード例は、describe-sentiment-detection-job を使用する方法を示しています。

AWS CLI

感情検出ジョブを記述するには

次のdescribe-sentiment-detection-job例では、非同期感情検出ジョブのプロパティを取得します。

```
aws comprehend describe-sentiment-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{  
  "SentimentDetectionJobProperties": {  
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
    "JobName": "movie_review_analysis",  
    "JobStatus": "IN_PROGRESS",  
    "SubmitTime": "2023-06-09T23:16:15.956000+00:00",  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",  
      "InputFormat": "ONE_DOC_PER_LINE"  
    },  
    "OutputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-  
TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"  
    },  
    "LanguageCode": "en",  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-servicerole"  
  }  
}
```

```
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSentimentDetectionJob](#)」の「」を参照してください。

describe-targeted-sentiment-detection-job

次の例は、describe-targeted-sentiment-detection-job を使用する方法を説明しています。

AWS CLI

ターゲット感情検出ジョブを記述するには

次のdescribe-targeted-sentiment-detection-job例では、非同期ターゲット感情検出ジョブのプロパティを取得します。

```
aws comprehend describe-targeted-sentiment-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "TargetedSentimentDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "movie_review_analysis",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
  }
}
```

```
"DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
  }
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTargetedSentimentDetectionJob](#)」の「」を参照してください。

describe-topics-detection-job

次のコード例は、describe-topics-detection-job を使用する方法を示しています。

AWS CLI

トピック検出ジョブを説明するには

次の describe-topics-detection-job の例では、非同期トピック検出ジョブのプロパティを取得します。

```
aws comprehend describe-topics-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "TopicsDetectionJobProperties": {
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobName": "example_topics_detection",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-
TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
  },
}
```

```
    "NumberOfTopics": 10,  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-examplerole"  
  }  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTopicsDetectionJob](#)」の「」を参照してください。

detect-dominant-language

次の例は、detect-dominant-language を使用する方法を説明しています。

AWS CLI

入力テキストの主要言語を検出するには

以下の detect-dominant-language は、入力テキストを分析し、主要言語を特定します。事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend detect-dominant-language \  
  --text "It is a beautiful day in Seattle."
```

出力:

```
{  
  "Languages": [  
    {  
      "LanguageCode": "en",  
      "Score": 0.9877256155014038  
    }  
  ]  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[主要言語](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectDominantLanguage](#)」の「」を参照してください。

detect-entities

次の例は、detect-entities を使用する方法を説明しています。

AWS CLI

入力テキストで名前付きエンティティを検出するには

次の detect-entities の例では、入力テキストを分析し、名前付きエンティティを返します。予測ごとに、事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend detect-entities \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st. Based on your autopay settings, \  
we will withdraw your payment on the due date from your bank account number  
XXXXXX1111 with the routing number XXXXX0000. \  
Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to  
Alice at AnySpa@example.com."
```

出力:

```
{  
  "Entities": [  
    {  
      "Score": 0.9994556307792664,  
      "Type": "PERSON",  
      "Text": "Zhang Wei",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9981022477149963,  
      "Type": "PERSON",  
      "Text": "John",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.9986887574195862,  
      "Type": "ORGANIZATION",
```

```
    "Text": "AnyCompany Financial Services, LLC",
    "BeginOffset": 33,
    "EndOffset": 67
  },
  {
    "Score": 0.9959119558334351,
    "Type": "OTHER",
    "Text": "1111-XXXX-1111-XXXX",
    "BeginOffset": 88,
    "EndOffset": 107
  },
  {
    "Score": 0.9708039164543152,
    "Type": "QUANTITY",
    "Text": ".53",
    "BeginOffset": 133,
    "EndOffset": 136
  },
  {
    "Score": 0.9987268447875977,
    "Type": "DATE",
    "Text": "July 31st",
    "BeginOffset": 152,
    "EndOffset": 161
  },
  {
    "Score": 0.9858865737915039,
    "Type": "OTHER",
    "Text": "XXXXXX1111",
    "BeginOffset": 271,
    "EndOffset": 281
  },
  {
    "Score": 0.9700471758842468,
    "Type": "OTHER",
    "Text": "XXXXX0000",
    "BeginOffset": 306,
    "EndOffset": 315
  },
  {
    "Score": 0.9591118693351746,
    "Type": "ORGANIZATION",
    "Text": "Sunshine Spa",
    "BeginOffset": 340,
```



```
    "EndOffset": 352
  },
  {
    "Score": 0.9797496795654297,
    "Type": "LOCATION",
    "Text": "123 Main St",
    "BeginOffset": 354,
    "EndOffset": 365
  },
  {
    "Score": 0.994929313659668,
    "Type": "PERSON",
    "Text": "Alice",
    "BeginOffset": 394,
    "EndOffset": 399
  },
  {
    "Score": 0.9949769377708435,
    "Type": "OTHER",
    "Text": "AnySpa@example.com",
    "BeginOffset": 403,
    "EndOffset": 418
  }
]
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[\[Entities\] \(エンティティ\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectEntities](#)」の「」を参照してください。

detect-key-phrases

次の例は、detect-key-phrases を使用する方法を説明しています。

AWS CLI

入力テキスト内のキーフレーズを検出するには

次の detect-key-phrases の例では、入力テキストを分析し、主要な名詞フレーズを特定します。予測ごとに、事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend detect-key-phrases \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
  account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by  
July 31st. Based on your autopay settings, \  
  we will withdraw your payment on the due date from your bank account number  
XXXXXX1111 with the routing number XXXXX0000. \  
  Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to  
Alice at AnySpa@example.com."
```

出力:

```
{  
  "KeyPhrases": [  
    {  
      "Score": 0.8996376395225525,  
      "Text": "Zhang Wei",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9992469549179077,  
      "Text": "John",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.988385021686554,  
      "Text": "Your AnyCompany Financial Services",  
      "BeginOffset": 28,  
      "EndOffset": 62  
    },  
    {  
      "Score": 0.8740853071212769,  
      "Text": "LLC credit card account 1111-XXXX-1111-XXXX",  
      "BeginOffset": 64,  
      "EndOffset": 107  
    },  
    {  
      "Score": 0.9999437928199768,  
      "Text": "a minimum payment",  
      "BeginOffset": 112,  
    }  
  ]  
}
```

```
    "EndOffset": 129
  },
  {
    "Score": 0.9998900890350342,
    "Text": ".53",
    "BeginOffset": 133,
    "EndOffset": 136
  },
  {
    "Score": 0.9979453086853027,
    "Text": "July 31st",
    "BeginOffset": 152,
    "EndOffset": 161
  },
  {
    "Score": 0.9983011484146118,
    "Text": "your autopay settings",
    "BeginOffset": 172,
    "EndOffset": 193
  },
  {
    "Score": 0.9996572136878967,
    "Text": "your payment",
    "BeginOffset": 211,
    "EndOffset": 223
  },
  {
    "Score": 0.9995037317276001,
    "Text": "the due date",
    "BeginOffset": 227,
    "EndOffset": 239
  },
  {
    "Score": 0.9702621698379517,
    "Text": "your bank account number XXXXXX1111",
    "BeginOffset": 245,
    "EndOffset": 280
  },
  {
    "Score": 0.9179925918579102,
    "Text": "the routing number XXXXX0000.Customer feedback",
    "BeginOffset": 286,
    "EndOffset": 332
  },
  },
```

```
{
  "Score": 0.9978160858154297,
  "Text": "Sunshine Spa",
  "BeginOffset": 337,
  "EndOffset": 349
},
{
  "Score": 0.9706913232803345,
  "Text": "123 Main St",
  "BeginOffset": 351,
  "EndOffset": 362
},
{
  "Score": 0.9941995143890381,
  "Text": "comments",
  "BeginOffset": 379,
  "EndOffset": 387
},
{
  "Score": 0.9759287238121033,
  "Text": "Alice",
  "BeginOffset": 391,
  "EndOffset": 396
},
{
  "Score": 0.8376792669296265,
  "Text": "AnySpa@example.com",
  "BeginOffset": 400,
  "EndOffset": 415
}
]
```

詳細については、「Amazon Comprehend 開発者ガイド」の「[キーフレーズ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectKeyPhrases](#)」の「」を参照してください。

detect-pii-entities

次のコード例は、detect-pii-entities を使用する方法を示しています。

AWS CLI

入力テキストの PII エンティティを検出するには

次のdetect-pii-entities例では、入力テキストを分析し、個人を特定できる情報 () を含むエンティティを識別しますPII。予測ごとに、事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend detect-pii-entities \  
  --language-code en \  
  --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC  
credit card \  
  account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by  
July 31st. Based on your autopay settings, \  
  we will withdraw your payment on the due date from your bank account number  
XXXXXX1111 with the routing number XXXXX0000. \  
  Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to  
Alice at AnySpa@example.com."
```

出力:

```
{  
  "Entities": [  
    {  
      "Score": 0.9998322129249573,  
      "Type": "NAME",  
      "BeginOffset": 6,  
      "EndOffset": 15  
    },  
    {  
      "Score": 0.9998878240585327,  
      "Type": "NAME",  
      "BeginOffset": 22,  
      "EndOffset": 26  
    },  
    {  
      "Score": 0.9994089603424072,  
      "Type": "CREDIT_DEBIT_NUMBER",  
      "BeginOffset": 88,  
      "EndOffset": 107  
    },  
    {  
      "Score": 0.9999760985374451,
```

```
    "Type": "DATE_TIME",
    "BeginOffset": 152,
    "EndOffset": 161
  },
  {
    "Score": 0.9999449253082275,
    "Type": "BANK_ACCOUNT_NUMBER",
    "BeginOffset": 271,
    "EndOffset": 281
  },
  {
    "Score": 0.9999847412109375,
    "Type": "BANK_ROUTING",
    "BeginOffset": 306,
    "EndOffset": 315
  },
  {
    "Score": 0.999925434589386,
    "Type": "ADDRESS",
    "BeginOffset": 354,
    "EndOffset": 365
  },
  {
    "Score": 0.9989161491394043,
    "Type": "NAME",
    "BeginOffset": 394,
    "EndOffset": 399
  },
  {
    "Score": 0.9994171857833862,
    "Type": "EMAIL",
    "BeginOffset": 403,
    "EndOffset": 418
  }
]
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[個人を特定できる情報 \(PII\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectPiiEntities](#)」の「」を参照してください。

detect-sentiment

次のコード例は、detect-sentiment を使用する方法を示しています。

AWS CLI

入力テキストの感情を検出するには

次の detect-sentiment の例では、入力テキストを分析し、一般的な感情 (POSITIVE、NEUTRAL、MIXED、または NEGATIVE) の推論を返します。

```
aws comprehend detect-sentiment \  
  --language-code en \  
  --text "It is a beautiful day in Seattle"
```

出力:

```
{  
  "Sentiment": "POSITIVE",  
  "SentimentScore": {  
    "Positive": 0.9976957440376282,  
    "Negative": 9.653854067437351e-05,  
    "Neutral": 0.002169104292988777,  
    "Mixed": 3.857641786453314e-05  
  }  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[感情](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectSentiment](#)」の「」を参照してください。

detect-syntax

次のコード例は、detect-syntax を使用する方法を示しています。

AWS CLI

入力テキスト内の品詞を検出するには

次の detect-syntax の例では、入力テキストの構文を分析し、さまざまな品詞を返します。予測ごとに、事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend detect-syntax \  
--language-code en \  
--text "It is a beautiful day in Seattle."
```

出力:

```
{  
  "SyntaxTokens": [  
    {  
      "TokenId": 1,  
      "Text": "It",  
      "BeginOffset": 0,  
      "EndOffset": 2,  
      "PartOfSpeech": {  
        "Tag": "PRON",  
        "Score": 0.9999740719795227  
      }  
    },  
    {  
      "TokenId": 2,  
      "Text": "is",  
      "BeginOffset": 3,  
      "EndOffset": 5,  
      "PartOfSpeech": {  
        "Tag": "VERB",  
        "Score": 0.999901294708252  
      }  
    },  
    {  
      "TokenId": 3,  
      "Text": "a",  
      "BeginOffset": 6,  
      "EndOffset": 7,  
      "PartOfSpeech": {  
        "Tag": "DET",  
        "Score": 0.9999938607215881  
      }  
    },  
    {  
      "TokenId": 4,  
      "Text": "beautiful",  
      "BeginOffset": 8,  
      "EndOffset": 17,  
    }  
  ]  
}
```



```
    "PartOfSpeech": {
      "Tag": "ADJ",
      "Score": 0.9987351894378662
    }
  },
  {
    "TokenId": 5,
    "Text": "day",
    "BeginOffset": 18,
    "EndOffset": 21,
    "PartOfSpeech": {
      "Tag": "NOUN",
      "Score": 0.9999796748161316
    }
  },
  {
    "TokenId": 6,
    "Text": "in",
    "BeginOffset": 22,
    "EndOffset": 24,
    "PartOfSpeech": {
      "Tag": "ADP",
      "Score": 0.9998047947883606
    }
  },
  {
    "TokenId": 7,
    "Text": "Seattle",
    "BeginOffset": 25,
    "EndOffset": 32,
    "PartOfSpeech": {
      "Tag": "PROPN",
      "Score": 0.9940530061721802
    }
  }
]
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[構文分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectSyntax](#)」の「」を参照してください。

detect-targeted-sentiment

次の例は、detect-targeted-sentiment を使用する方法を説明しています。

AWS CLI

入力テキスト内の名前付きエンティティのターゲット感情を検出するには

次のdetect-targeted-sentiment例では、入力テキストを分析し、各エンティティに関連付けられたターゲット感情に加えて、名前付きエンティティを返します。各予測の事前トレーニング済みモデルの信頼スコアも出力されます。

```
aws comprehend detect-targeted-sentiment \  
  --language-code en \  
  --text "I do not enjoy January because it is too cold but August is the perfect temperature"
```

出力:

```
{  
  "Entities": [  
    {  
      "DescriptiveMentionIndex": [  
        0  
      ],  
      "Mentions": [  
        {  
          "Score": 0.9999979734420776,  
          "GroupScore": 1.0,  
          "Text": "I",  
          "Type": "PERSON",  
          "MentionSentiment": {  
            "Sentiment": "NEUTRAL",  
            "SentimentScore": {  
              "Positive": 0.0,  
              "Negative": 0.0,  
              "Neutral": 1.0,  
              "Mixed": 0.0  
            }  
          },  
          "BeginOffset": 0,  
          "EndOffset": 1  
        }  
      ]  
    }  
  ]  
}
```

```
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9638869762420654,
      "GroupScore": 1.0,
      "Text": "January",
      "Type": "DATE",
      "MentionSentiment": {
        "Sentiment": "NEGATIVE",
        "SentimentScore": {
          "Positive": 0.0031610000878572464,
          "Negative": 0.9967250227928162,
          "Neutral": 0.00011100000119768083,
          "Mixed": 1.9999999949504854e-06
        }
      }
    },
    {
      "BeginOffset": 15,
      "EndOffset": 22
    }
  ]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9664419889450073,
      "GroupScore": 1.0,
      "Text": "August",
      "Type": "DATE",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Positive": 0.9999549984931946,
          "Negative": 3.999999989900971e-06,
          "Neutral": 4.099999932805076e-05,
          "Mixed": 0.0
        }
      }
    }
  ]
}
```

```

    }
  },
  "BeginOffset": 50,
  "EndOffset": 56
}
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "Score": 0.9803199768066406,
      "GroupScore": 1.0,
      "Text": "temperature",
      "Type": "ATTRIBUTE",
      "MentionSentiment": {
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Positive": 1.0,
          "Negative": 0.0,
          "Neutral": 0.0,
          "Mixed": 0.0
        }
      }
    },
    {
      "BeginOffset": 77,
      "EndOffset": 88
    }
  ]
}
]
}
}

```

詳細については、[「Amazon Comprehend デベロッパーガイド」](#)の「ターゲット感情」を参照してください。 Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectTargetedSentiment](#)」の「」を参照してください。

import-model

次の例は、import-model を使用する方法を説明しています。

AWS CLI

モデルをインポートするには

次のimport-model例では、別の AWS アカウントからモデルをインポートします。アカウントのドキュメント分類子モデル444455556666には、アカウントがモデル111122223333をインポートできるようにするリソースベースのポリシーがあります。

```
aws comprehend import-model \  
  --source-model-arn arn:aws:comprehend:us-west-2:444455556666:document-  
classifier/example-classifier
```

出力:

```
{  
  "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier"  
}
```

詳細については、[「Amazon Comprehend デベロッパーガイド」の AWS 「アカウント間のカスタムモデルのコピー」](#)を参照してください。 Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportModel](#)」の「」を参照してください。

list-datasets

次のコード例は、list-datasets を使用する方法を示しています。

AWS CLI

すべてのフライホイールデータセットを一覧表示するには

次のlist-datasets例では、フライホイールに関連付けられているすべてのデータセットを一覧表示します。

```
aws comprehend list-datasets \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity
```

出力:

```
{
  "DatasetPropertiesList": [
    {
      "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
flywheel-entity/dataset/example-dataset-1",
      "DatasetName": "example-dataset-1",
      "DatasetType": "TRAIN",
      "DatasetS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/
schemaVersion=1/20230616T200543Z/datasets/example-dataset-1/20230616T203710Z/",
      "Status": "CREATING",
      "CreationTime": "2023-06-16T20:37:10.400000+00:00"
    },
    {
      "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
flywheel-entity/dataset/example-dataset-2",
      "DatasetName": "example-dataset-2",
      "DatasetType": "TRAIN",
      "DatasetS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/
schemaVersion=1/20230616T200543Z/datasets/example-dataset-2/20230616T200607Z/",
      "Description": "TRAIN Dataset created by Flywheel creation.",
      "Status": "COMPLETED",
      "NumberOfDocuments": 5572,
      "CreationTime": "2023-06-16T20:06:07.722000+00:00"
    }
  ]
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[Flywheel Overview](#)」を参照してください。

- API 詳細については、「コマンドリファレンス[ListDatasets](#)」の「」を参照してください。
AWS CLI

list-document-classification-jobs

次のコード例は、list-document-classification-jobs を使用する方法を示しています。

AWS CLI

すべてのドキュメント分類ジョブを一覧表示するには

次の `list-document-classification-jobs` の例では、すべてのドキュメント分類ジョブを一覧表示しています。

```
aws comprehend list-document-classification-jobs
```

出力:

```
{
  "DocumentClassificationJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "exampleclassificationjob",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
      "EndTime": "2023-06-14T17:15:58.582000+00:00",
      "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/1234567890101-CLN-e758dd56b824aa717ceab551f11749fb/output/output.tar.gz"
      },
      "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE2",
      "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE2",
      "JobName": "exampleclassificationjob2",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-14T17:22:39.829000+00:00",
      "EndTime": "2023-06-14T17:28:46.107000+00:00",
      "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/jobdata/",
        "InputFormat": "ONE_DOC_PER_LINE"
      }
    }
  ]
}
```

```
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/1234567890101-CLN-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}
```

詳細については、「Amazon Comprehend 開発者ガイド」の「[カスタム分類](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDocumentClassificationJobs](#)」の「」を参照してください。

list-document-classifier-summaries

次の例は、list-document-classifier-summaries を使用する方法を説明しています。

AWS CLI

作成されたすべてのドキュメント分類子の概要を一覧表示するには

次のlist-document-classifier-summaries例では、作成されたすべてのドキュメント分類子の概要を一覧表示します。

```
aws comprehend list-document-classifier-summaries
```

出力:

```
{
  "DocumentClassifierSummariesList": [
    {
      "DocumentClassifierName": "example-classifier-1",
      "NumberOfVersions": 1,
      "LatestVersionCreatedAt": "2023-06-13T22:07:59.825000+00:00",
      "LatestVersionName": "1",
      "LatestVersionStatus": "TRAINED"
    },
    {
      "DocumentClassifierName": "example-classifier-2",
```



```
        "NumberOfVersions": 2,
        "LatestVersionCreatedAt": "2023-06-13T21:54:59.589000+00:00",
        "LatestVersionName": "2",
        "LatestVersionStatus": "TRAINED"
    }
]
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[カスタムモデルの作成と管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDocumentClassifierSummaries](#)」の「」を参照してください。

list-document-classifiers

次のコード例は、list-document-classifiers を使用する方法を示しています。

AWS CLI

すべてのドキュメント分類子を一覧表示するには

次の list-document-classifiers の例は、トレーニング済みおよびトレーニング中のすべてのドキュメント分類子モデルを一覧表示します。

```
aws comprehend list-document-classifiers
```

出力:

```
{
  "DocumentClassifierPropertiesList": [
    {
      "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier1",
      "LanguageCode": "en",
      "Status": "TRAINED",
      "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
      "EndTime": "2023-06-13T19:42:31.752000+00:00",
      "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
      "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
      "InputDataConfig": {
        "DataFormat": "COMPREHEND_CSV",
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
      }
    }
  ]
}
```

```

    },
    "OutputDataConfig": {},
    "ClassifierMetadata": {
      "NumberOfLabels": 3,
      "NumberOfTrainedDocuments": 5016,
      "NumberOfTestDocuments": 557,
      "EvaluationMetrics": {
        "Accuracy": 0.9856,
        "Precision": 0.9919,
        "Recall": 0.9459,
        "F1Score": 0.9673,
        "MicroPrecision": 0.9856,
        "MicroRecall": 0.9856,
        "MicroF1Score": 0.9856,
        "HammingLoss": 0.0144
      }
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-testorle",
    "Mode": "MULTI_CLASS"
  },
  {
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier2",
    "LanguageCode": "en",
    "Status": "TRAINING",
    "SubmitTime": "2023-06-13T21:20:28.690000+00:00",
    "InputDataConfig": {
      "DataFormat": "COMPREHEND_CSV",
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata"
    },
    "OutputDataConfig": {},
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-testorle",
    "Mode": "MULTI_CLASS"
  }
]
}

```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[カスタムモデルの作成と管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDocumentClassifiers](#)」の「」を参照してください。

list-dominant-language-detection-jobs

次のコード例は、list-dominant-language-detection-jobs を使用する方法を示しています。

AWS CLI

主要な言語検出ジョブをすべて一覧表示するには

次のlist-dominant-language-detection-jobs例では、進行中のすべての非同期優勢言語検出ジョブと完了した非同期優勢言語検出ジョブを一覧表示します。

```
aws comprehend list-dominant-language-detection-jobs
```

出力:

```
{
  "DominantLanguageDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "languageanalysis1",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-09T18:10:38.037000+00:00",
      "EndTime": "2023-06-09T18:18:45.498000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
      },
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
```

```
"JobName": "languageanalysis2",
"JobStatus": "STOPPED",
"SubmitTime": "2023-06-09T18:16:33.690000+00:00",
"EndTime": "2023-06-09T18:24:40.608000+00:00",
"InputDataConfig": {
  "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
  "InputFormat": "ONE_DOC_PER_LINE"
},
"OutputDataConfig": {
  "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-LANGUAGE-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
},
"DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
}
]
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListDominantLanguageDetectionJobs](#)」の「」を参照してください。

list-endpoints

次のコード例は、list-endpoints を使用する方法を示しています。

AWS CLI

すべてのエンドポイントを一覧表示するには

次のlist-endpoints例では、アクティブなモデル固有のエンドポイントをすべて一覧表示します。

```
aws comprehend list-endpoints
```

出力:

```
{
```

```
"EndpointPropertiesList": [  
  {  
    "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier-endpoint/ExampleClassifierEndpoint",  
    "Status": "IN_SERVICE",  
    "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/exampleclassifier1",  
    "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/exampleclassifier1",  
    "DesiredInferenceUnits": 1,  
    "CurrentInferenceUnits": 1,  
    "CreationTime": "2023-06-13T20:32:54.526000+00:00",  
    "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"  
  },  
  {  
    "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier-endpoint/ExampleClassifierEndpoint2",  
    "Status": "IN_SERVICE",  
    "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/exampleclassifier2",  
    "DesiredModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/exampleclassifier2",  
    "DesiredInferenceUnits": 1,  
    "CurrentInferenceUnits": 1,  
    "CreationTime": "2023-06-13T20:32:54.526000+00:00",  
    "LastModifiedTime": "2023-06-13T20:32:54.526000+00:00"  
  }  
]  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEndpoints](#)」の「」を参照してください。

list-entities-detection-jobs

次のコード例は、list-entities-detection-jobs を使用する方法を示しています。

AWS CLI

すべてのエンティティ検出ジョブを一覧表示するには

次のlist-entities-detection-jobs例では、すべての非同期エンティティ検出ジョブを一覧表示します。

```
aws comprehend list-entities-detection-jobs
```

出力:

```
{
  "EntitiesDetectionJobPropertiesList": [
    {
      "JobId": "468af39c28ab45b83eb0c4ab9EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-job/468af39c28ab45b83eb0c4ab9EXAMPLE",
      "JobName": "example-entities-detection",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-08T20:57:46.476000+00:00",
      "EndTime": "2023-06-08T21:05:53.718000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/thefolder/111122223333-NER-468af39c28ab45b83eb0c4ab9EXAMPLE/output/output.tar.gz"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "809691caeaab0e71406f80a28EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-job/809691caeaab0e71406f80a28EXAMPLE",
      "JobName": "example-entities-detection-2",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-08T21:30:15.323000+00:00",
      "EndTime": "2023-06-08T21:40:23.509000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
```

```
    "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-NER-809691caeaab0e71406f80a28EXAMPLE/output/output.tar.gz"
  },
  "LanguageCode": "en",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
},
{
  "JobId": "e00597c36b448b91d70dea165EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/e00597c36b448b91d70dea165EXAMPLE",
  "JobName": "example-entities-detection-3",
  "JobStatus": "STOPPED",
  "SubmitTime": "2023-06-08T22:19:28.528000+00:00",
  "EndTime": "2023-06-08T22:27:33.991000+00:00",
  "InputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
    "InputFormat": "ONE_DOC_PER_LINE"
  },
  "OutputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-NER-e00597c36b448b91d70dea165EXAMPLE/output/output.tar.gz"
  },
  "LanguageCode": "en",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
}
]
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[\[Entities\] \(エンティティ\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEntitiesDetectionJobs](#)」の「」を参照してください。

list-entity-recognizer-summaries

次のコード例は、list-entity-recognizer-summaries を使用する方法を示しています。

AWS CLI

作成されたすべてのエンティティレコグナイザーの概要を一覧表示するには

次のlist-entity-recognizer-summaries例では、すべてのエンティティレコグナイザーの概要を一覧表示します。

```
aws comprehend list-entity-recognizer-summaries
```

出力:

```
{
  "EntityRecognizerSummariesList": [
    {
      "RecognizerName": "entity-recognizer-3",
      "NumberOfVersions": 2,
      "LatestVersionCreatedAt": "2023-06-15T23:15:07.621000+00:00",
      "LatestVersionName": "2",
      "LatestVersionStatus": "STOP_REQUESTED"
    },
    {
      "RecognizerName": "entity-recognizer-2",
      "NumberOfVersions": 1,
      "LatestVersionCreatedAt": "2023-06-14T22:55:27.805000+00:00",
      "LatestVersionName": "2"
      "LatestVersionStatus": "TRAINED"
    },
    {
      "RecognizerName": "entity-recognizer-1",
      "NumberOfVersions": 1,
      "LatestVersionCreatedAt": "2023-06-14T20:44:59.631000+00:00",
      "LatestVersionName": "1",
      "LatestVersionStatus": "TRAINED"
    }
  ]
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[カスタムエンティティ認識](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEntityRecognizerSummaries](#)」の「」を参照してください。

list-entity-recognizers

次のコード例は、list-entity-recognizers を使用する方法を示しています。

AWS CLI

すべてのカスタムエンティティレコグナイザーを一覧表示するには

次のlist-entity-recognizers例では、作成されたすべてのカスタムエンティティレコグナイザーを一覧表示します。

```
aws comprehend list-entity-recognizers
```

出力:

```
{
  "EntityRecognizerPropertiesList": [
    {
      "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/EntityRecognizer/version/1",
      "LanguageCode": "en",
      "Status": "TRAINED",
      "SubmitTime": "2023-06-14T20:44:59.631000+00:00",
      "EndTime": "2023-06-14T20:59:19.532000+00:00",
      "TrainingStartTime": "2023-06-14T20:48:52.811000+00:00",
      "TrainingEndTime": "2023-06-14T20:58:11.473000+00:00",
      "InputDataConfig": {
        "DataFormat": "COMPREHEND_CSV",
        "EntityTypes": [
          {
            "Type": "BUSINESS"
          }
        ],
        "Documents": {
          "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/dataset/",
          "InputFormat": "ONE_DOC_PER_LINE"
        },
        "EntityList": {
          "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/entity.csv"
        }
      },
      "RecognizerMetadata": {
        "NumberOfTrainedDocuments": 1814,
        "NumberOfTestDocuments": 486,
        "EvaluationMetrics": {
          "Precision": 100.0,
          "Recall": 100.0,

```

```
        "F1Score": 100.0
      },
      "EntityTypes": [
        {
          "Type": "BUSINESS",
          "EvaluationMetrics": {
            "Precision": 100.0,
            "Recall": 100.0,
            "F1Score": 100.0
          },
          "NumberOfTrainMentions": 1520
        }
      ]
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-servicerole",
    "VersionName": "1"
  },
  {
    "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/entityrecognizer3",
    "LanguageCode": "en",
    "Status": "TRAINED",
    "SubmitTime": "2023-06-14T22:57:51.056000+00:00",
    "EndTime": "2023-06-14T23:14:13.894000+00:00",
    "TrainingStartTime": "2023-06-14T23:01:33.984000+00:00",
    "TrainingEndTime": "2023-06-14T23:13:02.984000+00:00",
    "InputDataConfig": {
      "DataFormat": "COMPREHEND_CSV",
      "EntityTypes": [
        {
          "Type": "DEVICE"
        }
      ]
    },
    "Documents": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/raw_txt.csv",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "EntityList": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/trainingdata/entity_list.csv"
    }
  },
  "RecognizerMetadata": {
    "NumberOfTrainedDocuments": 4616,
```

```
    "NumberOfTestDocuments": 3489,
    "EvaluationMetrics": {
      "Precision": 98.54227405247813,
      "Recall": 100.0,
      "F1Score": 99.26578560939794
    },
    "EntityTypes": [
      {
        "Type": "DEVICE",
        "EvaluationMetrics": {
          "Precision": 98.54227405247813,
          "Recall": 100.0,
          "F1Score": 99.26578560939794
        },
        "NumberOfTrainMentions": 2764
      }
    ]
  },
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-servicerole"
}
]
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[カスタムエンティティ認識](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEntityRecognizers](#)」の「」を参照してください。

list-events-detection-jobs

次のコード例は、list-events-detection-jobs を使用する方法を示しています。

AWS CLI

すべてのイベント検出ジョブを一覧表示するには

次のlist-events-detection-jobs例では、すべての非同期イベント検出ジョブを一覧表示します。

```
aws comprehend list-events-detection-jobs
```

出力:

```
{
  "EventsDetectionJobPropertiesList": [
    {
      "JobId": "aa9593f9203e84f3ef032ce18EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/aa9593f9203e84f3ef032ce18EXAMPLE",
      "JobName": "events_job_1",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-12T19:14:57.751000+00:00",
      "EndTime": "2023-06-12T19:21:04.962000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/EventsData/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/1111222233333-EVENTS-aa9593f9203e84f3ef032ce18EXAMPLE/output/"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
      "TargetEventTypes": [
        "BANKRUPTCY",
        "EMPLOYMENT",
        "CORPORATE_ACQUISITION",
        "CORPORATE_MERGER",
        "INVESTMENT_GENERAL"
      ]
    },
    {
      "JobId": "4a990a2f7e82adfca6e171135EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/4a990a2f7e82adfca6e171135EXAMPLE",
      "JobName": "events_job_2",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-12T19:55:43.702000+00:00",
      "EndTime": "2023-06-12T20:03:49.893000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/EventsData/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
```

```

        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/1111222233333-EVENTS-4a990a2f7e82adfca6e171135EXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "TargetEventTypes": [
        "BANKRUPTCY",
        "EMPLOYMENT",
        "CORPORATE_ACQUISITION",
        "CORPORATE_MERGER",
        "INVESTMENT_GENERAL"
    ]
}
]
}
}

```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEventsDetectionJobs](#)」の「」を参照してください。

list-flywheel-iteration-history

次のコード例は、list-flywheel-iteration-history を使用する方法を示しています。

AWS CLI

すべてのフライホイールイテレーション履歴を一覧表示するには

次のlist-flywheel-iteration-history例では、フライホイールのすべての反復を一覧表示します。

```

aws comprehend list-flywheel-iteration-history
  --flywheel-arn arn:aws:comprehend:us-west-2:1111222233333:flywheel/example-flywheel

```

出力:

```

{
  "FlywheelIterationPropertiesList": [
    {

```

```
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel",
    "FlywheelIterationId": "20230619EXAMPLE",
    "CreationTime": "2023-06-19T04:00:32.594000+00:00",
    "EndTime": "2023-06-19T04:00:49.248000+00:00",
    "Status": "COMPLETED",
    "Message": "FULL_ITERATION: Flywheel iteration performed all functions
successfully.",
    "EvaluatedModelArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier/version/1",
    "EvaluatedModelMetrics": {
        "AverageF1Score": 0.7742663922375772,
        "AverageF1Score": 0.9876464664646313,
        "AveragePrecision": 0.9800000253081214,
        "AverageRecall": 0.9445600253081214,
        "AverageAccuracy": 0.9997281665190434
    },
    "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-BUCKET/example-flywheel/
schemaVersion=1/20230619EXAMPLE/evaluation/20230619EXAMPLE/"
  },
  {
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/
example-flywheel-2",
    "FlywheelIterationId": "20230616EXAMPLE",
    "CreationTime": "2023-06-16T21:10:26.385000+00:00",
    "EndTime": "2023-06-16T23:33:16.827000+00:00",
    "Status": "COMPLETED",
    "Message": "FULL_ITERATION: Flywheel iteration performed all functions
successfully.",
    "EvaluatedModelArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/spamvshamclassify/version/1",
    "EvaluatedModelMetrics": {
        "AverageF1Score": 0.7742663922375772,
        "AverageF1Score": 0.9767700253081214,
        "AveragePrecision": 0.9767700253081214,
        "AverageRecall": 0.9767700253081214,
        "AverageAccuracy": 0.9858281665190434
    },
    "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-BUCKET/example-
flywheel-2/schemaVersion=1/20230616EXAMPLE/evaluation/20230616EXAMPLE/"
  }
]
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Flywheel 概要](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFlywheelIterationHistory](#)」の「」を参照してください。

list-flywheels

次の例は、list-flywheels を使用する方法を説明しています。

AWS CLI

すべてのフライホイールを一覧表示するには

次のlist-flywheels例では、作成されたすべてのフライホイールを一覧表示します。

```
aws comprehend list-flywheels
```

出力:

```
{
  "FlywheelSummaryList": [
    {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1",
      "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier/version/1",
      "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/example-flywheel-1/schemaVersion=1/20230616T200543Z/",
      "Status": "ACTIVE",
      "ModelType": "DOCUMENT_CLASSIFIER",
      "CreationTime": "2023-06-16T20:05:43.242000+00:00",
      "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",
      "LatestFlywheelIteration": "20230619T040032Z"
    },
    {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-2",
      "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/exampleclassifier2/version/1",
      "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/example-flywheel-2/schemaVersion=1/20220616T200543Z/",
      "Status": "ACTIVE",
```

```
    "ModelType": "DOCUMENT_CLASSIFIER",
    "CreationTime": "2022-06-16T20:05:43.242000+00:00",
    "LastModifiedTime": "2022-06-19T04:00:43.027000+00:00",
    "LatestFlywheelIteration": "20220619T040032Z"
  }
]
}
```

詳細については、Amazon Comprehend デベロッパーガイドの「[Flywheel 概要](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFlywheels](#)」の「」を参照してください。

list-key-phrases-detection-jobs

次のコード例は、list-key-phrases-detection-jobs を使用する方法を示しています。

AWS CLI

すべてのキーフレーズ検出ジョブを一覧表示するには

次のlist-key-phrases-detection-jobs例では、進行中のすべてのキーフレーズ検出ジョブと完了した非同期キーフレーズ検出ジョブを一覧表示します。

```
aws comprehend list-key-phrases-detection-jobs
```

出力:

```
{
  "KeyPhrasesDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "keyphrasesanalysis1",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-08T22:31:43.767000+00:00",
      "EndTime": "2023-06-08T22:39:52.565000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/AsyncBatchJobs/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
    },
  ],
}
```



```
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-KP-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a33EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a33EXAMPLE",
    "JobName": "keyphrasesanalysis2",
    "JobStatus": "STOPPED",
    "SubmitTime": "2023-06-08T22:57:52.154000+00:00",
    "EndTime": "2023-06-08T23:05:48.385000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-KP-123456abcdeb0e11022f22a33EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a44EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-
detection-job/123456abcdeb0e11022f22a44EXAMPLE",
    "JobName": "keyphrasesanalysis3",
    "JobStatus": "FAILED",
    "Message": "NO_READ_ACCESS_TO_INPUT: The provided data access role does
not have proper access to the input data.",
    "SubmitTime": "2023-06-09T16:47:04.029000+00:00",
    "EndTime": "2023-06-09T16:47:18.413000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
```

```
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-KP-123456abcdeb0e11022f22a44EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
]
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListKeyPhrasesDetectionJobs](#)」の「」を参照してください。

list-pii-entities-detection-jobs

次のコード例は、list-pii-entities-detection-jobs を使用する方法を示しています。

AWS CLI

すべての pii エンティティ検出ジョブを一覧表示するには

次のlist-pii-entities-detection-jobs例では、進行中の非同期 pii 検出ジョブと完了した非同期 pii 検出ジョブをすべて一覧表示します。

```
aws comprehend list-pii-entities-detection-jobs
```

出力:

```
{
  "PiiEntitiesDetectionJobPropertiesList": [
    {
      "JobId": "6f9db0c42d0c810e814670ee4EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/6f9db0c42d0c810e814670ee4EXAMPLE",
      "JobName": "example-pii-detection-job",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-09T21:02:46.241000+00:00",
      "EndTime": "2023-06-09T21:12:52.602000+00:00",
    }
  ]
}
```

```

    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/111122223333-
PII-6f9db0c42d0c810e814670ee4EXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "Mode": "ONLY_OFFSETS"
  },
  {
    "JobId": "d927562638cfa739331a99b3cEXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/d927562638cfa739331a99b3cEXAMPLE",
    "JobName": "example-pii-detection-job-2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-09T21:20:58.211000+00:00",
    "EndTime": "2023-06-09T21:31:06.027000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/AsyncBatchJobs/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-PII-d927562638cfa739331a99b3cEXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "Mode": "ONLY_OFFSETS"
  }
]
}

```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPiiEntitiesDetectionJobs](#)」の「」を参照してください。

list-sentiment-detection-jobs

次のコード例は、list-sentiment-detection-jobs を使用する方法を示しています。

AWS CLI

すべての感情検出ジョブを一覧表示するには

次のlist-sentiment-detection-jobs例では、進行中のすべての非同期感情検出ジョブと完了した非同期感情検出ジョブを一覧表示します。

```
aws comprehend list-sentiment-detection-jobs
```

出力:

```
{
  "SentimentDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "example-sentiment-detection-job",
      "JobStatus": "IN_PROGRESS",
      "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
      "EndTime": "2023-06-09T22:52:27.416000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE2",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE2",
      "JobName": "example-sentiment-detection-job-2",
      "JobStatus": "COMPLETED",
    }
  ]
}
```

```

    "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
    "EndTime": "2023-06-09T23:26:00.168000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData2",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSentimentDetectionJobs](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、Amazon Comprehend リソースのタグを一覧表示します。

```

aws comprehend list-tags-for-resource \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1

```

出力:

```
{
```

```
"ResourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1",
  "Tags": [
    {
      "Key": "Department",
      "Value": "Finance"
    },
    {
      "Key": "location",
      "Value": "Seattle"
    }
  ]
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-targeted-sentiment-detection-jobs

次の例は、list-targeted-sentiment-detection-jobs を使用する方法を説明しています。

AWS CLI

すべてのターゲット感情検出ジョブを一覧表示するには

次のlist-targeted-sentiment-detection-jobs例では、進行中のすべてのターゲット感情検出ジョブと完了した非同期ターゲット感情検出ジョブを一覧表示します。

```
aws comprehend list-targeted-sentiment-detection-jobs
```

出力:

```
{
  "TargetedSentimentDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
```

```

    "JobName": "example-targeted-sentiment-detection-job",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
    "EndTime": "2023-06-09T22:52:27.416000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-I0role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
    "JobName": "example-targeted-sentiment-detection-job-2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
    "EndTime": "2023-06-09T23:26:00.168000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData2",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListTargetedSentimentDetectionJobs](#)」の「」を参照してください。

list-topics-detection-jobs

次の例は、list-topics-detection-jobs を使用する方法を説明しています。

AWS CLI

トピック検出ジョブをすべて一覧表示するには

次の list-topics-detection-jobs の例では、進行中および完了した非同期トピック検出ジョブをすべて一覧表示します。

```
aws comprehend list-topics-detection-jobs
```

出力:

```
{
  "TopicsDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "topic-analysis-1",
      "JobStatus": "IN_PROGRESS",
      "SubmitTime": "2023-06-09T18:40:35.384000+00:00",
      "EndTime": "2023-06-09T18:46:41.936000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
      },
      "NumberOfTopics": 10,
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    },
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE2",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-job/123456abcdeb0e11022f22a11EXAMPLE2",
      "JobName": "topic-analysis-2",
      "JobStatus": "COMPLETED",
    }
  ]
}
```



```

    "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
    "EndTime": "2023-06-09T18:50:50.872000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a1EXAMPLE3",
    "JobName": "topic-analysis-2",
    "JobStatus": "IN_PROGRESS",
    "SubmitTime": "2023-06-09T18:50:56.737000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE3/output/output.tar.gz"
    },
    "NumberOfTopics": 10,
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTopicsDetectionJobs](#)」の「」を参照してください。

put-resource-policy

次のコード例は、put-resource-policy を使用する方法を示しています。

AWS CLI

リソースベースのポリシーをアタッチするには

次のput-resource-policy例では、リソースベースのポリシーをモデルにアタッチして、を別の AWS アカウントでインポートできるようにします。ポリシーはアカウントのモデルにアタッチ111122223333され、アカウントのモデル444455556666インポートを許可します。

```
aws comprehend put-resource-policy \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1 \  
  --resource-policy '{"Version":"2012-10-17","Statement":  
[{"Effect":"Allow","Action":"comprehend:ImportModel","Resource":"*","Principal":  
{"AWS":["arn:aws:iam::444455556666:root"]}]}'
```

出力：

```
{  
  "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"  
}
```

詳細については、[「Amazon Comprehend デベロッパーガイド」](#)の AWS 「[アカウント間のカスタムモデルのコピー](#)」を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「[コマンドリファレンスPutResourcePolicy](#)」の「」を参照してください。

start-document-classification-job

次の例は、start-document-classification-job を使用する方法を説明しています。

AWS CLI

ドキュメント分類ジョブを開始するには

次の start-document-classification-job の例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルに対して、カスタムモデルを

使用してドキュメント分類ジョブを開始します。この例では、入力 S3 バケットには、SampleSMStext1.txt、SampleSMStext2.txt、SampleSMStext3.txt が含まれています。このモデルは、スパムおよびスパム以外の SMS メッセージのドキュメント分類、または「ham」について以前にトレーニングされています。ジョブが完了すると、output.tar.gz は --output-data-config タグで指定された場所に配置されます。output.tar.gz には各ドキュメントの分類を一覧表示する predictions.jsonl が含まれています。Json の出力は、1 ファイルに 1 行で出力されますが、ここでは読みやすい形式で表示されています。

```
aws comprehend start-document-classification-job \  
  --job-name exampleclassificationjob \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET-INPUT/jobdata/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/mymodel/version/12
```

SampleSMStext1.txt の内容:

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

SampleSMStext2.txt の内容:

```
"Hi, when do you want me to pick you up from practice?"
```

SampleSMStext3.txt の内容:

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

出力:

```
{  
  "JobId": "e758dd56b824aa717ceab551fEXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-  
job/e758dd56b824aa717ceab551fEXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

predictions.jsonl の内容:

```
{"File": "SampleSMSText1.txt", "Line": "0", "Classes": [{"Name": "spam", "Score": 0.9999}, {"Name": "ham", "Score": 0.0001}]}
{"File": "SampleSMSText2.txt", "Line": "0", "Classes": [{"Name": "ham", "Score": 0.9994}, {"Name": "spam", "Score": 0.0006}]}
{"File": "SampleSMSText3.txt", "Line": "0", "Classes": [{"Name": "spam", "Score": 0.9999}, {"Name": "ham", "Score": 0.0001}]}
```

詳細については、「Amazon Comprehend 開発者ガイド」の「[カスタム分類](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartDocumentClassificationJob](#)」の「」を参照してください。

start-dominant-language-detection-job

次のコード例は、start-dominant-language-detection-job を使用する方法を示しています。

AWS CLI

非同期言語検出ジョブを開始するには

次のstart-dominant-language-detection-job例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期言語検出ジョブを開始します。この例では、S3 バケットにが含まれていますSampletext1.txt。ジョブが完了すると、フォルダ output は --output-data-config タグで指定された場所に配置されます。フォルダには、各テキストファイルの主要な言語と、各予測のトレーニング済みモデルの信頼スコアoutput.txtが含まれています。

```
aws comprehend start-dominant-language-detection-job \
  --job-name example_language_analysis_job \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role \
  --language-code en
```

Sampletext1.txt の内容 :

```
"Physics is the natural science that involves the study of matter and its motion and behavior through space and time, along with related concepts such as energy and force."
```

出力:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

output.txt の内容:

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score": 0.9913753867149353}], "Line": 0}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartDominantLanguageDetectionJob](#)」の「」を参照してください。

start-entities-detection-job

次の例は、start-entities-detection-job を使用する方法を説明しています。

AWS CLI

例 1: 事前トレーニング済みモデルを使用して標準エンティティ検出ジョブを開始するには

次のstart-entities-detection-job例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期エンティティ検出ジョブを開始します。この例の S3 バケットにはSampletext1.txt、Sampletext2.txt、およびが含まれていますSampletext3.txt。ジョブが完了すると、フォルダ outputは --output-data-config タグで指定された場所に配置されます。フォルダには、各テキストファイル内で検出されたすべての名前付きエンティティと、各予測の事前トレーニング済みモデルの信頼スコアが一覧表示output.txtされます。Json 出力は入力ファイルごとに 1 行に出力されますが、読みやすいようにここでフォーマットされています。

```
aws comprehend start-entities-detection-job \  
  --job-name entitiestest \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --language-code en
```

Sampletext1.txt の内容:

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st."
```

Sampletext2.txt の内容:

```
"Dear Max, based on your autopay settings for your account example1.org account, we  
will withdraw your payment on the due date from your bank account number XXXXXX1111  
with the routing number XXXXX0000. "
```

Sampletext3.txt の内容:

```
"Jane, please submit any customer feedback from this weekend to AnySpa, 123 Main St,  
Anywhere and send comments to Alice at AnySpa@example.com."
```

出力:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

読みやすいように線のインデントoutput.txtを含む の内容 :

```
{  
  "Entities": [  
    {  
      "BeginOffset": 6,
```

```
"EndOffset": 15,
"Score": 0.9994006636420306,
"Text": "Zhang Wei",
"Type": "PERSON"
},
{
"BeginOffset": 22,
"EndOffset": 26,
"Score": 0.9976647915128143,
"Text": "John",
"Type": "PERSON"
},
{
"BeginOffset": 33,
"EndOffset": 67,
"Score": 0.9984608700836206,
"Text": "AnyCompany Financial Services, LLC",
"Type": "ORGANIZATION"
},
{
"BeginOffset": 88,
"EndOffset": 107,
"Score": 0.9868521019555556,
"Text": "1111-XXXX-1111-XXXX",
"Type": "OTHER"
},
{
"BeginOffset": 133,
"EndOffset": 139,
"Score": 0.998242565709204,
"Text": "$24.53",
"Type": "QUANTITY"
},
{
"BeginOffset": 155,
"EndOffset": 164,
"Score": 0.9993039263159287,
"Text": "July 31st",
"Type": "DATE"
}
],
"File": "SampleText1.txt",
"Line": 0
}
```

```
{
  "Entities": [
    {
      "BeginOffset": 5,
      "EndOffset": 8,
      "Score": 0.9866232147545232,
      "Text": "Max",
      "Type": "PERSON"
    },
    {
      "BeginOffset": 156,
      "EndOffset": 166,
      "Score": 0.9797723450933329,
      "Text": "XXXXXX1111",
      "Type": "OTHER"
    },
    {
      "BeginOffset": 191,
      "EndOffset": 200,
      "Score": 0.9247838572396843,
      "Text": "XXXXX0000",
      "Type": "OTHER"
    }
  ],
  "File": "SampleText2.txt",
  "Line": 0
}
{
  "Entities": [
    {
      "Score": 0.9990532994270325,
      "Type": "PERSON",
      "Text": "Jane",
      "BeginOffset": 0,
      "EndOffset": 4
    },
    {
      "Score": 0.9519651532173157,
      "Type": "DATE",
      "Text": "this weekend",
      "BeginOffset": 47,
      "EndOffset": 59
    },
    {
```



```
"Score": 0.5566426515579224,
>Type": "ORGANIZATION",
>Text": "AnySpa",
>BeginOffset": 63,
>EndOffset": 69
},
{
>Score": 0.8059805631637573,
>Type": "LOCATION",
>Text": "123 Main St, Anywhere",
>BeginOffset": 71,
>EndOffset": 92
},
{
>Score": 0.998830258846283,
>Type": "PERSON",
>Text": "Alice",
>BeginOffset": 114,
>EndOffset": 119
},
{
>Score": 0.997818112373352,
>Type": "OTHER",
>Text": "AnySpa@example.com",
>BeginOffset": 123,
>EndOffset": 138
}
],
>File": "SampleText3.txt",
>Line": 0
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

例 2: カスタムエンティティ検出ジョブを開始するには

次のstart-entities-detection-job例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期カスタムエンティティ検出ジョブを開始します。この例では、この例の S3 バケットに SampleFeedback1.txt、SampleFeedback2.txt、が含まれています SampleFeedback3.txt。エンティティレコグナイザーモデルは、デバイス名を認識するためのカスタマーサポートフィードバックについてトレーニングされました。ジョブが完了すると、--output-data-config タグで指定された場所にフォルダ output が配置されま

す。フォルダにはが含まれておりoutput.txt、各テキストファイル内で検出されたすべての名前付きエンティティと、各予測の事前トレーニング済みモデルの信頼スコアが一覧表示されます。Json の出力は、1 ファイルに 1 行で出力されますが、ここでは読みやすい形式で表示されています。

```
aws comprehend start-entities-detection-job \  
  --job-name customentitiestest \  
  --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/entityrecognizer" \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/jobdata/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-I0role"
```

SampleFeedback1.txt の内容:

```
"I've been on the AnyPhone app have had issues for 24 hours when trying to pay bill. Cannot make payment. Sigh. | Oh man! Lets get that app up and running. DM me, and we can get to work!"
```

SampleFeedback2.txt の内容:

```
"Hi, I have a discrepancy with my new bill. Could we get it sorted out? A rep added stuff I didnt sign up for when I did my AnyPhone 10 upgrade. | We can absolutely get this sorted!"
```

SampleFeedback3.txt の内容:

```
"Is the by 1 get 1 free AnySmartPhone promo still going on? | Hi Christian! It ended yesterday, send us a DM if you have any questions and we can take a look at your options!"
```

出力:

```
{  
  "JobId": "019ea9edac758806850fa8a79ff83021",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-job/019ea9edac758806850fa8a79ff83021",  
  "JobStatus": "SUBMITTED"
```

```
}
```

読みやすいように線のインデントoutput.txtを含むの内容 :

```
{
  "Entities": [
    {
      "BeginOffset": 17,
      "EndOffset": 25,
      "Score": 0.9999728210205924,
      "Text": "AnyPhone",
      "Type": "DEVICE"
    }
  ],
  "File": "SampleFeedback1.txt",
  "Line": 0
}
{
  "Entities": [
    {
      "BeginOffset": 123,
      "EndOffset": 133,
      "Score": 0.9999892116761524,
      "Text": "AnyPhone 10",
      "Type": "DEVICE"
    }
  ],
  "File": "SampleFeedback2.txt",
  "Line": 0
}
{
  "Entities": [
    {
      "BeginOffset": 23,
      "EndOffset": 35,
      "Score": 0.9999971389852362,
      "Text": "AnySmartPhone",
      "Type": "DEVICE"
    }
  ],
  "File": "SampleFeedback3.txt",
  "Line": 0
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[カスタムエンティティ認識](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartEntitiesDetectionJob](#)」の「」を参照してください。

start-events-detection-job

次のコード例は、start-events-detection-job を使用する方法を示しています。

AWS CLI

非同期イベント検出ジョブを開始するには

次のstart-events-detection-job例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期イベント検出ジョブを開始します。考えられるターゲットイベントタイプには、BANKRUPTCY、EMPLOYMENT、CORPORATE_ACQUISITIONINVESTMENT_GENERAL、CORPORATE_MERGER、およびSTOCK_SPLITが含まれます。この例の S3 バケットにはSampleText1.txt、SampleText2.txt、およびSampleText3.txtが含まれています。ジョブが完了すると、フォルダ outputは--output-data-config タグで指定された場所に配置されます。フォルダにはSampleText1.txt.out、SampleText2.txt.out、およびSampleText3.txt.outが含まれます。JSON 出力はファイルごとに 1 行に出力されますが、読みやすくするためにここでフォーマットされます。

```
aws comprehend start-events-detection-job \
  --job-name events-detection-1 \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/EventsData" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-servicerole \
  --language-code en \
  --target-event-types "BANKRUPTCY" "EMPLOYMENT" "CORPORATE_ACQUISITION" "CORPORATE_MERGER" "INVESTMENT_GENERAL"
```

SampleText1.txt の内容:

```
"Company AnyCompany grew by increasing sales and through acquisitions. After purchasing competing firms in 2020, AnyBusiness, a part of the AnyBusinessGroup, gave Jane Does firm a going rate of one cent a gallon or forty-two cents a barrel."
```

SampleText2.txt の内容:

```
"In 2021, AnyCompany officially purchased AnyBusiness for 100 billion dollars,
surprising and exciting the shareholders."
```

SampleText3.txt の内容:

```
"In 2022, AnyCompany stock crashed 50. Eventually later that year they filed for
bankruptcy."
```

出力:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

読みやすいように線のインデントSampleText1.txt.outを含む の内容 :

```
{
  "Entities": [
    {
      "Mentions": [
        {
          "BeginOffset": 8,
          "EndOffset": 18,
          "Score": 0.99977,
          "Text": "AnyCompany",
          "Type": "ORGANIZATION",
          "GroupScore": 1
        },
        {
          "BeginOffset": 112,
          "EndOffset": 123,
          "Score": 0.999747,
          "Text": "AnyBusiness",
          "Type": "ORGANIZATION",
          "GroupScore": 0.979826
        },
        {
```

```
        "BeginOffset": 171,
        "EndOffset": 175,
        "Score": 0.999615,
        "Text": "firm",
        "Type": "ORGANIZATION",
        "GroupScore": 0.871647
    }
],
},
{
  "Mentions": [
    {
      "BeginOffset": 97,
      "EndOffset": 102,
      "Score": 0.987687,
      "Text": "firms",
      "Type": "ORGANIZATION",
      "GroupScore": 1
    }
  ]
},
{
  "Mentions": [
    {
      "BeginOffset": 103,
      "EndOffset": 110,
      "Score": 0.999458,
      "Text": "in 2020",
      "Type": "DATE",
      "GroupScore": 1
    }
  ]
},
{
  "Mentions": [
    {
      "BeginOffset": 160,
      "EndOffset": 168,
      "Score": 0.999649,
      "Text": "John Doe",
      "Type": "PERSON",
      "GroupScore": 1
    }
  ]
}
```

```
    }
  ],
  "Events": [
    {
      "Type": "CORPORATE_ACQUISITION",
      "Arguments": [
        {
          "EntityIndex": 0,
          "Role": "INVESTOR",
          "Score": 0.99977
        }
      ],
      "Triggers": [
        {
          "BeginOffset": 56,
          "EndOffset": 68,
          "Score": 0.999967,
          "Text": "acquisitions",
          "Type": "CORPORATE_ACQUISITION",
          "GroupScore": 1
        }
      ]
    },
    {
      "Type": "CORPORATE_ACQUISITION",
      "Arguments": [
        {
          "EntityIndex": 1,
          "Role": "INVESTEES",
          "Score": 0.987687
        },
        {
          "EntityIndex": 2,
          "Role": "DATE",
          "Score": 0.999458
        },
        {
          "EntityIndex": 3,
          "Role": "INVESTOR",
          "Score": 0.999649
        }
      ],
      "Triggers": [
        {
```

```
        "BeginOffset": 76,  
        "EndOffset": 86,  
        "Score": 0.999973,  
        "Text": "purchasing",  
        "Type": "CORPORATE_ACQUISITION",  
        "GroupScore": 1  
      }  
    ]  
  }  
],  
"File": "SampleText1.txt",  
"Line": 0  
}
```

SampleText2.txt.out の内容:

```
{  
  "Entities": [  
    {  
      "Mentions": [  
        {  
          "BeginOffset": 0,  
          "EndOffset": 7,  
          "Score": 0.999473,  
          "Text": "In 2021",  
          "Type": "DATE",  
          "GroupScore": 1  
        }  
      ]  
    },  
    {  
      "Mentions": [  
        {  
          "BeginOffset": 9,  
          "EndOffset": 19,  
          "Score": 0.999636,  
          "Text": "AnyCompany",  
          "Type": "ORGANIZATION",  
          "GroupScore": 1  
        }  
      ]  
    },  
  ]  
}
```



```
"Mentions": [
  {
    "BeginOffset": 45,
    "EndOffset": 56,
    "Score": 0.999712,
    "Text": "AnyBusiness",
    "Type": "ORGANIZATION",
    "GroupScore": 1
  }
],
{
  "Mentions": [
    {
      "BeginOffset": 61,
      "EndOffset": 80,
      "Score": 0.998886,
      "Text": "100 billion dollars",
      "Type": "MONETARY_VALUE",
      "GroupScore": 1
    }
  ]
},
{
  "Events": [
    {
      "Type": "CORPORATE_ACQUISITION",
      "Arguments": [
        {
          "EntityIndex": 3,
          "Role": "AMOUNT",
          "Score": 0.998886
        },
        {
          "EntityIndex": 2,
          "Role": "INVESTEE",
          "Score": 0.999712
        },
        {
          "EntityIndex": 0,
          "Role": "DATE",
          "Score": 0.999473
        }
      ]
    }
  ]
}
```

```
        "EntityIndex": 1,
        "Role": "INVESTOR",
        "Score": 0.999636
      }
    ],
    "Triggers": [
      {
        "BeginOffset": 31,
        "EndOffset": 40,
        "Score": 0.99995,
        "Text": "purchased",
        "Type": "CORPORATE_ACQUISITION",
        "GroupScore": 1
      }
    ]
  }
],
"File": "SampleText2.txt",
"Line": 0
}
```

SampleText3.txt.out の内容:

```
{
  "Entities": [
    {
      "Mentions": [
        {
          "BeginOffset": 9,
          "EndOffset": 19,
          "Score": 0.999774,
          "Text": "AnyCompany",
          "Type": "ORGANIZATION",
          "GroupScore": 1
        },
        {
          "BeginOffset": 66,
          "EndOffset": 70,
          "Score": 0.995717,
          "Text": "they",
          "Type": "ORGANIZATION",
          "GroupScore": 0.997626
        }
      ]
    }
  ]
}
```

```
]
},
{
  "Mentions": [
    {
      "BeginOffset": 50,
      "EndOffset": 65,
      "Score": 0.999656,
      "Text": "later that year",
      "Type": "DATE",
      "GroupScore": 1
    }
  ]
},
],
"Events": [
  {
    "Type": "BANKRUPTCY",
    "Arguments": [
      {
        "EntityIndex": 1,
        "Role": "DATE",
        "Score": 0.999656
      },
      {
        "EntityIndex": 0,
        "Role": "FILER",
        "Score": 0.995717
      }
    ]
  },
],
"Triggers": [
  {
    "BeginOffset": 81,
    "EndOffset": 91,
    "Score": 0.999936,
    "Text": "bankruptcy",
    "Type": "BANKRUPTCY",
    "GroupScore": 1
  }
]
}
],
"File": "SampleText3.txt",
"Line": 0
```

```
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartEventsDetectionJob](#)」の「」を参照してください。

start-flywheel-iteration

次の例は、start-flywheel-iteration を使用する方法を説明しています。

AWS CLI

フライホイールの反復を開始するには

次のstart-flywheel-iteration例では、フライホイールの反復を開始します。このオペレーションでは、フライホイール内の新しいデータセットを使用して、新しいモデルバージョンをトレーニングします。

```
aws comprehend start-flywheel-iteration \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

出力:

```
{  
  "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",  
  "FlywheelIterationId": "12345123TEXAMPLE"  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Flywheel 概要](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartFlywheelIteration](#)」の「」を参照してください。

start-key-phrases-detection-job

次の例は、start-key-phrases-detection-job を使用する方法を説明しています。

AWS CLI

キーフレーズ検出ジョブを開始するには

次のstart-key-phrases-detection-job例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期キーフレーズ検出ジョブを開始します。この例の S3 バケットにはSampletext1.txt、Sampletext2.txt、およびが含まれていませんSampletext3.txt。ジョブが完了すると、フォルダ outputは --output-data-config タグで指定された場所に配置されます。フォルダには、各テキストファイル内で検出されたすべてのキーフレーズと、各予測の事前トレーニング済みモデルの信頼スコアoutput.txtを含むファイルが含まれます。Json の出力は、1 ファイルに 1 行で出力されますが、ここでは読みやすい形式で表示されています。

```
aws comprehend start-key-phrases-detection-job \  
  --job-name keyphrasesanalytest1 \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role" \  
  --language-code en
```

Sampletext1.txt の内容:

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st."
```

Sampletext2.txt の内容:

```
"Dear Max, based on your autopay settings for your account Internet.org account, we  
will withdraw your payment on the due date from your bank account number XXXXXX1111  
with the routing number XXXXX0000. "
```

Sampletext3.txt の内容:

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123  
Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

出力:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

読みやすいように線のインデントoutput.txtを含むの内容 :

```
{
  "File": "SampleText1.txt",
  "KeyPhrases": [
    {
      "BeginOffset": 6,
      "EndOffset": 15,
      "Score": 0.9748965572679326,
      "Text": "Zhang Wei"
    },
    {
      "BeginOffset": 22,
      "EndOffset": 26,
      "Score": 0.9997344722354619,
      "Text": "John"
    },
    {
      "BeginOffset": 28,
      "EndOffset": 62,
      "Score": 0.9843791074032948,
      "Text": "Your AnyCompany Financial Services"
    },
    {
      "BeginOffset": 64,
      "EndOffset": 107,
      "Score": 0.8976122401721824,
      "Text": "LLC credit card account 1111-XXXX-1111-XXXX"
    },
    {
      "BeginOffset": 112,
      "EndOffset": 129,
      "Score": 0.9999612982629748,
      "Text": "a minimum payment"
    },
    {
```

```
    "BeginOffset": 133,
    "EndOffset": 139,
    "Score": 0.99975728947036,
    "Text": "$24.53"
  },
  {
    "BeginOffset": 155,
    "EndOffset": 164,
    "Score": 0.9940866241449973,
    "Text": "July 31st"
  }
],
"Line": 0
}
{
  "File": "SampleText2.txt",
  "KeyPhrases": [
    {
      "BeginOffset": 0,
      "EndOffset": 8,
      "Score": 0.9974021100118472,
      "Text": "Dear Max"
    },
    {
      "BeginOffset": 19,
      "EndOffset": 40,
      "Score": 0.9961120519515884,
      "Text": "your autopay settings"
    },
    {
      "BeginOffset": 45,
      "EndOffset": 78,
      "Score": 0.9980620070116009,
      "Text": "your account Internet.org account"
    },
    {
      "BeginOffset": 97,
      "EndOffset": 109,
      "Score": 0.999919660140754,
      "Text": "your payment"
    },
    {
      "BeginOffset": 113,
      "EndOffset": 125,
```

```
    "Score": 0.9998370719754205,
    "Text": "the due date"
  },
  {
    "BeginOffset": 131,
    "EndOffset": 166,
    "Score": 0.9955068678502509,
    "Text": "your bank account number XXXXXX1111"
  },
  {
    "BeginOffset": 172,
    "EndOffset": 200,
    "Score": 0.8653433315829526,
    "Text": "the routing number XXXXX0000"
  }
],
"Line": 0
}
{
  "File": "SampleText3.txt",
  "KeyPhrases": [
    {
      "BeginOffset": 0,
      "EndOffset": 4,
      "Score": 0.9142947833681668,
      "Text": "Jane"
    },
    {
      "BeginOffset": 20,
      "EndOffset": 41,
      "Score": 0.9984325676596763,
      "Text": "any customer feedback"
    },
    {
      "BeginOffset": 47,
      "EndOffset": 59,
      "Score": 0.9998782448150636,
      "Text": "this weekend"
    },
    {
      "BeginOffset": 63,
      "EndOffset": 75,
      "Score": 0.99866741830757,
      "Text": "Sunshine Spa"
    }
  ]
}
```



```
    },
    {
      "BeginOffset": 77,
      "EndOffset": 88,
      "Score": 0.9695803485466054,
      "Text": "123 Main St"
    },
    {
      "BeginOffset": 108,
      "EndOffset": 116,
      "Score": 0.9997065928550928,
      "Text": "comments"
    },
    {
      "BeginOffset": 120,
      "EndOffset": 125,
      "Score": 0.9993466833825161,
      "Text": "Alice"
    },
    {
      "BeginOffset": 129,
      "EndOffset": 144,
      "Score": 0.9654563612885667,
      "Text": "AnySpa@example.com"
    }
  ],
  "Line": 0
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartKeyPhrasesDetectionJob](#)」の「」を参照してください。

start-pii-entities-detection-job

次のコード例は、start-pii-entities-detection-job を使用する方法を示しています。

AWS CLI

非同期PII検出ジョブを開始するには

次のstart-pii-entities-detection-job例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期個人識別情報 (PII) エンティティ検出ジョブを開始します。この例の S3 バケットにはSampletext1.txt、Sampletext2.txt、およびが含まれていますSampletext3.txt。ジョブが完了すると、フォルダ outputは --output-data-config タグで指定された場所に配置されます。フォルダにはSampleText1.txt.out、各テキストファイル内の名前付きエンティティを一覧表示SampleText3.txt.outする、SampleText2.txt.out、およびが含まれます。Json の出力は、1 ファイルに 1 行で出力されますが、ここでは読みやすい形式で表示されています。

```
aws comprehend start-pii-entities-detection-job \  
  --job-name entities_test \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role \  
  --language-code en \  
  --mode ONLY_OFFSETS
```

Sampletext1.txt の内容:

```
"Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card  
account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by July  
31st."
```

Sampletext2.txt の内容:

```
"Dear Max, based on your autopay settings for your account Internet.org account, we  
will withdraw your payment on the due date from your bank account number XXXXXX1111  
with the routing number XXXXX0000. "
```

Sampletext3.txt の内容:

```
"Jane, please submit any customer feedback from this weekend to Sunshine Spa, 123  
Main St, Anywhere and send comments to Alice at AnySpa@example.com."
```

出力:

```
{
```

```
"JobId": "123456abcdeb0e11022f22a11EXAMPLE",
"JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
"JobStatus": "SUBMITTED"
}
```

読みやすいように線のインデントSampleText1.txt.outを含むの内容 :

```
{
  "Entities": [
    {
      "BeginOffset": 6,
      "EndOffset": 15,
      "Type": "NAME",
      "Score": 0.9998490510222595
    },
    {
      "BeginOffset": 22,
      "EndOffset": 26,
      "Type": "NAME",
      "Score": 0.9998937958019426
    },
    {
      "BeginOffset": 88,
      "EndOffset": 107,
      "Type": "CREDIT_DEBIT_NUMBER",
      "Score": 0.9554297245278491
    },
    {
      "BeginOffset": 155,
      "EndOffset": 164,
      "Type": "DATE_TIME",
      "Score": 0.9999720462925257
    }
  ],
  "File": "SampleText1.txt",
  "Line": 0
}
```

読みやすいように行インデントSampleText2.txt.outを含むの内容 :

```
{
  "Entities": [
```

```
{
  "BeginOffset": 5,
  "EndOffset": 8,
  "Type": "NAME",
  "Score": 0.9994390774924007
},
{
  "BeginOffset": 58,
  "EndOffset": 70,
  "Type": "URL",
  "Score": 0.9999958276922101
},
{
  "BeginOffset": 156,
  "EndOffset": 166,
  "Type": "BANK_ACCOUNT_NUMBER",
  "Score": 0.9999721058045592
},
{
  "BeginOffset": 191,
  "EndOffset": 200,
  "Type": "BANK_ROUTING",
  "Score": 0.9998968945989909
}
],
"File": "SampleText2.txt",
"Line": 0
}
```

読みやすいように線のインデントSampleText3.txt.outを含むの内容：

```
{
  "Entities": [
    {
      "BeginOffset": 0,
      "EndOffset": 4,
      "Type": "NAME",
      "Score": 0.999949934606805
    },
    {
      "BeginOffset": 77,
      "EndOffset": 88,
      "Type": "ADDRESS",
```

```
    "Score": 0.9999035300466904
  },
  {
    "BeginOffset": 120,
    "EndOffset": 125,
    "Type": "NAME",
    "Score": 0.9998203838716296
  },
  {
    "BeginOffset": 129,
    "EndOffset": 144,
    "Type": "EMAIL",
    "Score": 0.9998313473105228
  }
],
"File": "SampleText3.txt",
"Line": 0
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartPiiEntitiesDetectionJob](#)」の「」を参照してください。

start-sentiment-detection-job

次の例は、start-sentiment-detection-job を使用する方法を説明しています。

AWS CLI

非同期感情分析ジョブを開始するには

次のstart-sentiment-detection-job例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期感情分析検出ジョブを開始します。この例の S3 バケットフォルダにはSampleMovieReview1.txt、SampleMovieReview2.txt、およびが含まれていますSampleMovieReview3.txt。ジョブが完了すると、フォルダ outputは --output-data-config タグで指定された場所に配置されます。フォルダには ファイルが含まれておりoutput.txt、各テキストファイルの一般的な感情と、各予測の事前トレーニング済みモデルの信頼スコアが含まれています。Json の出力は、1 ファイルに 1 行で出力されますが、ここでは読みやすい形式で表示されています。

```
aws comprehend start-sentiment-detection-job \  
  --job-name example-sentiment-detection-job \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/MovieData" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role
```

SampleMovieReview1.txt の内容:

```
"The film, AnyMovie2, is fairly predictable and just okay."
```

SampleMovieReview2.txt の内容:

```
"AnyMovie2 is the essential sci-fi film that I grew up watching when I was a kid. I  
highly recommend this movie."
```

SampleMovieReview3.txt の内容:

```
"Don't get fooled by the 'awards' for AnyMovie2. All parts of the film were poorly  
stolen from other modern directors."
```

出力:

```
{  
  "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-  
job/0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobStatus": "SUBMITTED"  
}
```

読みやすいようにインデントの行output.txtを含むの内容:

```
{  
  "File": "SampleMovieReview1.txt",  
  "Line": 0,  
  "Sentiment": "MIXED",  
  "SentimentScore": {  
    "Mixed": 0.6591159105300903,  

```

```
        "Negative": 0.26492202281951904,
        "Neutral": 0.035430654883384705,
        "Positive": 0.04053137078881264
    }
}
{
  "File": "SampleMovieReview2.txt",
  "Line": 0,
  "Sentiment": "POSITIVE",
  "SentimentScore": {
    "Mixed": 0.000008718466233403888,
    "Negative": 0.00006134175055194646,
    "Neutral": 0.0002941041602753103,
    "Positive": 0.9996358156204224
  }
}
{
  "File": "SampleMovieReview3.txt",
  "Line": 0,
  "Sentiment": "NEGATIVE",
  "SentimentScore": {
    "Mixed": 0.004146667663007975,
    "Negative": 0.9645107984542847,
    "Neutral": 0.016559595242142677,
    "Positive": 0.014782938174903393
  }
}
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartSentimentDetectionJob](#)」の「」を参照してください。

start-targeted-sentiment-detection-job

次のコード例は、start-targeted-sentiment-detection-job を使用する方法を示しています。

AWS CLI

非同期ターゲット感情分析ジョブを開始するには

次の `start-targeted-sentiment-detection-job` 例では、`--input-data-config` タグで指定されたアドレスにあるすべてのファイルの非同期ターゲット感情分析検出ジョブを開始します。この例の S3 バケットフォルダには `SampleMovieReview1.txt`、`SampleMovieReview2.txt`、および `SampleMovieReview3.txt` が含まれています。ジョブが完了すると、`output.tar.gz` は `--output-data-config` タグで指定された場所に配置されます。`output.tar.gz` には、ファイル `SampleMovieReview1.txt.out`、`SampleMovieReview2.txt.out`、および `SampleMovieReview3.txt.out`、それぞれに 1 つの入力テキストファイルのすべての名前付きエンティティと関連する感情が含まれます。

```
aws comprehend start-targeted-sentiment-detection-job \  
  --job-name targeted_movie_review_analysis1 \  
  --language-code en \  
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/MovieData" \  
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \  
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/  
AmazonComprehendServiceRole-example-role
```

`SampleMovieReview1.txt` の内容:

```
"The film, AnyMovie, is fairly predictable and just okay."
```

`SampleMovieReview2.txt` の内容:

```
"AnyMovie is the essential sci-fi film that I grew up watching when I was a kid. I  
highly recommend this movie."
```

`SampleMovieReview3.txt` の内容:

```
"Don't get fooled by the 'awards' for AnyMovie. All parts of the film were poorly  
stolen from other modern directors."
```

出力:

```
{  
  "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-  
detection-job/0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobStatus": "SUBMITTED"
```



```
}
```

読みやすいように行インデントSampleMovieReview1.txt.outを含むの内容:

```
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 4,
          "EndOffset": 8,
          "Score": 0.994972,
          "GroupScore": 1,
          "Text": "film",
          "Type": "MOVIE",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0,
              "Negative": 0,
              "Neutral": 1,
              "Positive": 0
            }
          }
        }
      ]
    },
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 10,
          "EndOffset": 18,
          "Score": 0.631368,
          "GroupScore": 1,
          "Text": "AnyMovie",
          "Type": "ORGANIZATION",
          "MentionSentiment": {
```

```
        "Sentiment": "POSITIVE",
        "SentimentScore": {
          "Mixed": 0.001729,
          "Negative": 0.000001,
          "Neutral": 0.000318,
          "Positive": 0.997952
        }
      }
    ]
  },
  "File": "SampleMovieReview1.txt",
  "Line": 0
}
```

読みやすさのためのSampleMovieReview2.txt.out行インデントの内容 :

```
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 0,
          "EndOffset": 8,
          "Score": 0.854024,
          "GroupScore": 1,
          "Text": "AnyMovie",
          "Type": "MOVIE",
          "MentionSentiment": {
            "Sentiment": "POSITIVE",
            "SentimentScore": {
              "Mixed": 0,
              "Negative": 0,
              "Neutral": 0.000007,
              "Positive": 0.999993
            }
          }
        }
      ],
    },
  ]
}
```

```
    "BeginOffset": 104,
    "EndOffset": 109,
    "Score": 0.999129,
    "GroupScore": 0.502937,
    "Text": "movie",
    "Type": "MOVIE",
    "MentionSentiment": {
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Mixed": 0,
        "Negative": 0,
        "Neutral": 0,
        "Positive": 1
      }
    }
  },
  {
    "BeginOffset": 33,
    "EndOffset": 37,
    "Score": 0.999823,
    "GroupScore": 0.999252,
    "Text": "film",
    "Type": "MOVIE",
    "MentionSentiment": {
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Mixed": 0,
        "Negative": 0,
        "Neutral": 0.000001,
        "Positive": 0.999999
      }
    }
  }
],
{
  "DescriptiveMentionIndex": [
    0,
    1,
    2
  ],
  "Mentions": [
    {
      "BeginOffset": 43,
```

```
"EndOffset": 44,
"Score": 0.999997,
"GroupScore": 1,
"Text": "I",
"Type": "PERSON",
"MentionSentiment": {
  "Sentiment": "NEUTRAL",
  "SentimentScore": {
    "Mixed": 0,
    "Negative": 0,
    "Neutral": 1,
    "Positive": 0
  }
}
},
{
  "BeginOffset": 80,
  "EndOffset": 81,
  "Score": 0.999996,
  "GroupScore": 0.52523,
  "Text": "I",
  "Type": "PERSON",
  "MentionSentiment": {
    "Sentiment": "NEUTRAL",
    "SentimentScore": {
      "Mixed": 0,
      "Negative": 0,
      "Neutral": 1,
      "Positive": 0
    }
  }
},
{
  "BeginOffset": 67,
  "EndOffset": 68,
  "Score": 0.999994,
  "GroupScore": 0.999499,
  "Text": "I",
  "Type": "PERSON",
  "MentionSentiment": {
    "Sentiment": "NEUTRAL",
    "SentimentScore": {
      "Mixed": 0,
      "Negative": 0,
```

```
        "Neutral": 1,
        "Positive": 0
      }
    }
  ],
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 75,
      "EndOffset": 78,
      "Score": 0.999978,
      "GroupScore": 1,
      "Text": "kid",
      "Type": "PERSON",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0,
          "Neutral": 1,
          "Positive": 0
        }
      }
    }
  ]
}
],
"File": "SampleMovieReview2.txt",
"Line": 0
}
```

読みやすいように線のインデントSampleMovieReview3.txt.outを含むの内容:

```
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        1
      ]
    }
  ]
}
```

```
],
  "Mentions": [
    {
      "BeginOffset": 64,
      "EndOffset": 68,
      "Score": 0.992953,
      "GroupScore": 0.999814,
      "Text": "film",
      "Type": "MOVIE",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0.000004,
          "Negative": 0.010425,
          "Neutral": 0.989543,
          "Positive": 0.000027
        }
      }
    }
  ],
  {
    "BeginOffset": 37,
    "EndOffset": 45,
    "Score": 0.999782,
    "GroupScore": 1,
    "Text": "AnyMovie",
    "Type": "ORGANIZATION",
    "MentionSentiment": {
      "Sentiment": "POSITIVE",
      "SentimentScore": {
        "Mixed": 0.000095,
        "Negative": 0.039847,
        "Neutral": 0.000673,
        "Positive": 0.959384
      }
    }
  }
]
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
```

```
    "BeginOffset": 47,
    "EndOffset": 50,
    "Score": 0.999991,
    "GroupScore": 1,
    "Text": "All",
    "Type": "QUANTITY",
    "MentionSentiment": {
      "Sentiment": "NEUTRAL",
      "SentimentScore": {
        "Mixed": 0.000001,
        "Negative": 0.000001,
        "Neutral": 0.999998,
        "Positive": 0
      }
    }
  }
},
{
  "DescriptiveMentionIndex": [
    0
  ],
  "Mentions": [
    {
      "BeginOffset": 106,
      "EndOffset": 115,
      "Score": 0.542083,
      "GroupScore": 1,
      "Text": "directors",
      "Type": "PERSON",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0,
          "Neutral": 1,
          "Positive": 0
        }
      }
    }
  ]
}
],
"File": "SampleMovieReview3.txt",
```

```
"Line": 0
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartTargetedSentimentDetectionJob](#)」の「」を参照してください。

start-topics-detection-job

次の例は、start-topics-detection-job を使用する方法を説明しています。

AWS CLI

トピック検出分析ジョブを開始するには

次の start-topics-detection-job の例では、--input-data-config タグで指定されたアドレスにあるすべてのファイルの非同期トピック検出ジョブを開始します。ジョブが完了すると、フォルダ、output は --output-data-config タグで指定された場所に配置されます。output には topic-terms.csv と doc-topics.csv が含まれています。最初の出力ファイル topic-terms.csv は、コレクション内のトピックのリストです。デフォルトでは、リストには、各トピックの上位の言葉が重みに応じてトピック別に含まれています。2 つ目のファイル doc-topics.csv には、トピックに関連するドキュメントと、そのトピックに関連するドキュメントの割合が一覧表示されます。

```
aws comprehend start-topics-detection-job \
  --job-name example_topics_detection_job \
  --language-code en \
  --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
  --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
  --data-access-role-arn arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role \
  --language-code en
```

出力:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
```



```
"JobStatus": "SUBMITTED"
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[トピックのモデリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartTopicsDetectionJob](#)」の「」を参照してください。

stop-dominant-language-detection-job

次の例は、stop-dominant-language-detection-job を使用する方法を説明しています。

AWS CLI

非同期優勢言語検出ジョブを停止するには

次のstop-dominant-language-detection-job例では、進行中の非同期優勢言語検出ジョブを停止しています。現在のジョブ状態が IN_PROGRESSの場合、ジョブは終了としてマークされ、STOP_REQUESTED状態になります。ジョブを停止する前にジョブが完了した場合、そのジョブの状態は COMPLETED になります。

```
aws comprehend stop-dominant-language-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopDominantLanguageDetectionJob](#)」の「」を参照してください。

stop-entities-detection-job

次の例は、stop-entities-detection-job を使用する方法を説明しています。

AWS CLI

非同期エンティティ検出ジョブを停止するには

次のstop-entities-detection-job例では、進行中の非同期エンティティ検出ジョブを停止します。現在のジョブ状態が IN_PROGRESSの場合、ジョブは終了としてマークされ、STOP_REQUESTED状態になります。ジョブを停止する前にジョブが完了した場合、そのジョブの状態は COMPLETED になります。

```
aws comprehend stop-entities-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopEntitiesDetectionJob](#)」の「」を参照してください。

stop-events-detection-job

次の例は、stop-events-detection-job を使用する方法を説明しています。

AWS CLI

非同期イベント検出ジョブを停止するには

次のstop-events-detection-job例では、進行中の非同期イベント検出ジョブを停止します。現在のジョブ状態が IN_PROGRESSの場合、ジョブは終了としてマークされ、STOP_REQUESTED状態になります。ジョブを停止する前にジョブが完了した場合、そのジョブの状態は COMPLETED になります。

```
aws comprehend stop-events-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopEventsDetectionJob](#)」の「」を参照してください。

stop-key-phrases-detection-job

次のコード例は、stop-key-phrases-detection-job を使用する方法を示しています。

AWS CLI

非同期キーフレーズ検出ジョブを停止するには

次のstop-key-phrases-detection-job例では、進行中の非同期キーフレーズ検出ジョブを停止します。現在のジョブ状態が IN_PROGRESSの場合、ジョブは終了としてマークされ、STOP_REQUESTED状態になります。ジョブを停止する前にジョブが完了した場合、そのジョブの状態は COMPLETED になります。

```
aws comprehend stop-key-phrases-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopKeyPhrasesDetectionJob](#)」の「」を参照してください。

stop-pii-entities-detection-job

次の例は、stop-pii-entities-detection-job を使用する方法を説明しています。

AWS CLI

非同期 pii エンティティ検出ジョブを停止するには

次のstop-pii-entities-detection-job例では、進行中の非同期 pii エンティティ検出ジョブを停止します。現在のジョブ状態が IN_PROGRESSの場合、ジョブは終了としてマークされ、STOP_REQUESTED状態になります。ジョブを停止する前にジョブが完了した場合、そのジョブの状態は COMPLETED になります。

```
aws comprehend stop-pii-entities-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopPiiEntitiesDetectionJob](#)」の「」を参照してください。

stop-sentiment-detection-job

次の例は、stop-sentiment-detection-job を使用する方法を説明しています。

AWS CLI

非同期感情検出ジョブを停止するには

次のstop-sentiment-detection-job例では、進行中の非同期感情検出ジョブを停止します。現在のジョブ状態が IN_PROGRESSの場合、ジョブは終了としてマークされ、STOP_REQUESTED状態になります。ジョブを停止する前にジョブが完了した場合、そのジョブの状態は COMPLETED になります。

```
aws comprehend stop-sentiment-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopSentimentDetectionJob](#)」の「」を参照してください。

stop-targeted-sentiment-detection-job

次のコード例は、stop-targeted-sentiment-detection-job を使用する方法を示しています。

AWS CLI

非同期ターゲット感情検出ジョブを停止するには

次のstop-targeted-sentiment-detection-job例では、進行中の非同期ターゲット感情検出ジョブを停止します。現在のジョブ状態が IN_PROGRESSの場合、ジョブは終了としてマークされ、STOP_REQUESTED状態になります。ジョブを停止する前にジョブが完了した場合、そのジョブの状態は COMPLETED になります。

```
aws comprehend stop-targeted-sentiment-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

出力:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のインサイトのための非同期分析](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StopTargetedSentimentDetectionJob](#)」の「」を参照してください。

stop-training-document-classifier

次の例は、stop-training-document-classifier を使用する方法を説明しています。

AWS CLI

ドキュメント分類子モデルのトレーニングを停止するには

次のstop-training-document-classifier例では、進行中のドキュメント分類子モデルのトレーニングを停止します。

```
aws comprehend stop-training-document-classifier
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[カスタムモデルの作成と管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StopTrainingDocumentClassifier](#)」の「」を参照してください。

stop-training-entity-recognizer

次のコード例は、stop-training-entity-recognizer を使用する方法を示しています。

AWS CLI

エンティティレコグナイザーモデルのトレーニングを停止するには

次のstop-training-entity-recognizer例では、進行中のエンティティレコグナイザーモデルのトレーニングを停止します。

```
aws comprehend stop-training-entity-recognizer
```

```
--entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/examplerrecognizer1"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[カスタムモデルの作成と管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopTrainingEntityRecognizer](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

例 1: リソースにタグを付けるには

次のtag-resource例では、Amazon Comprehend リソースに 1 つのタグを追加します。

```
aws comprehend tag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
  example-classifier/version/1 \  
  --tags Key=Location,Value=Seattle
```

このコマンドには出力がありません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[リソースのタグ付け](#)」を参照してください。

例 2: リソースに複数のタグを追加するには

次のtag-resource例では、Amazon Comprehend リソースに複数のタグを追加します。

```
aws comprehend tag-resource \  
  --resource-arn "arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
  example-classifier/version/1" \  
  --tags Key=Location,Value=Seattle Key=Department,Value=Finance
```

このコマンドには出力がありません。

詳細については、Amazon Comprehend デベロッパーガイド」の「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

例 1: リソースから単一のタグを削除するには

次のuntag-resource例では、Amazon Comprehend リソースから単一のタグを削除します。

```
aws comprehend untag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1  
  --tag-keys Location
```

このコマンドでは何も出力されません。

詳細については、Amazon Comprehend デベロッパーガイド」の「[リソースのタグ付け](#)」を参照してください。

例 2: リソースから複数のタグを削除するには

次のuntag-resource例では、Amazon Comprehend リソースから複数のタグを削除します。

```
aws comprehend untag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1  
  --tag-keys Location Department
```

このコマンドでは何も出力されません。

詳細については、Amazon Comprehend デベロッパーガイド」の「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-endpoint

次のコード例は、update-endpoint を使用する方法を示しています。

AWS CLI

例 1: エンドポイントの推論単位を更新するには

次のupdate-endpoint例では、エンドポイントに関する情報を更新します。この例では、推論ユニットの数が増えます。

```
aws comprehend update-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint \  
  --desired-inference-units 2
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

例 2: エンドポイントのアクティブモデルを更新するには

次のupdate-endpoint例では、エンドポイントに関する情報を更新します。この例では、アクティブなモデルが変更されます。

```
aws comprehend update-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint \  
  --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier-new
```

このコマンドでは何も出力されません。

詳細については、「Amazon Comprehend デベロッパーガイド」の「[Amazon Comprehend のエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateEndpoint](#)」の「」を参照してください。

update-flywheel

次のコード例は、update-flywheel を使用する方法を示しています。

AWS CLI

フライホイール設定を更新するには

次のupdate-flywheel例では、フライホイール設定を更新します。この例では、フライホイールのアクティブなモデルが更新されます。

```
aws comprehend update-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1 \  
  --active-model-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/new-example-classifier-model
```

出力:

```
{  
  "FlywheelProperties": {  
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity",  
    "ActiveModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/new-example-classifier-model",  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/AmazonComprehendServiceRole-example-role",  
    "TaskConfig": {  
      "LanguageCode": "en",  
      "DocumentClassificationConfig": {  
        "Mode": "MULTI_CLASS"  
      }  
    },  
    "DataLakeS3Uri": "s3://DOC-EXAMPLE-BUCKET/flywheel-entity/schemaVersion=1/20230616T200543Z/",  
    "DataSecurityConfig": {},  
    "Status": "ACTIVE",  
    "ModelType": "DOCUMENT_CLASSIFIER",  
    "CreationTime": "2023-06-16T20:05:43.242000+00:00",  
    "LastModifiedTime": "2023-06-19T04:00:43.027000+00:00",  
    "LatestFlywheelIteration": "20230619T040032Z"  
  }  
}
```

詳細については、Amazon Comprehend デベロッパーガイド」の「[Flywheel 概要](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFlywheel](#)」の「」を参照してください。

を使用した Amazon Comprehend Medical の例 AWS CLI

次のコード例は、Amazon Comprehend Medical AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-entities-detection-v2-job

次の例は、describe-entities-detection-v2-job を使用する方法を説明しています。

AWS CLI

エンティティ検出ジョブを記述するには

次のdescribe-entities-detection-v2-job例は、非同期エンティティ検出ジョブに関連付けられたプロパティを示しています。

```
aws comprehendmedical describe-entities-detection-v2-job \  
  --job-id "ab9887877365fe70299089371c043b96"
```

出力:

```
{  
  "ComprehendMedicalAsyncJobProperties": {  
    "JobId": "ab9887877365fe70299089371c043b96",  
    "JobStatus": "COMPLETED",
```

```

    "SubmitTime": "2020-03-18T21:20:15.614000+00:00",
    "EndTime": "2020-03-18T21:27:07.350000+00:00",
    "ExpirationTime": "2020-07-16T21:20:15+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": ""
    },
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
      "S3Key": "867139942017-EntitiesDetection-
ab9887877365fe70299089371c043b96/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "DetectEntitiesModelV20190930"
  }
}

```

詳細については、Amazon Comprehend Medical デベロッパーガイド」の「[バッチAPIs](#)」を参照してください。

- API 詳細については、コマンドリファレンスの[DescribeEntitiesDetectionV2Job](#)」を参照してください。AWS CLI

describe-icd10-cm-inference-job

次のコード例は、describe-icd10-cm-inference-job を使用する方法を示しています。

AWS CLI

ICD-10-CM 推論ジョブを記述するには

次のdescribe-icd10-cm-inference-job例では、指定された job-id を持つリクエストされた推論ジョブのプロパティについて説明します。

```

aws comprehendmedical describe-icd10-cm-inference-job \
  --job-id "5780034166536cdb52ffa3295a1b00a7"

```

出力:

```
{
```

```
"ComprehendMedicalAsyncJobProperties": {
  "JobId": "5780034166536cdb52ffa3295a1b00a7",
  "JobStatus": "COMPLETED",
  "SubmitTime": "2020-05-18T21:20:15.614000+00:00",
  "EndTime": "2020-05-18T21:27:07.350000+00:00",
  "ExpirationTime": "2020-09-16T21:20:15+00:00",
  "InputDataConfig": {
    "S3Bucket": "comp-med-input",
    "S3Key": "AKIAIOSFODNN7EXAMPLE"
  },
  "OutputDataConfig": {
    "S3Bucket": "comp-med-output",
    "S3Key": "AKIAIOSFODNN7EXAMPLE"
  },
  "LanguageCode": "en",
  "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
  "ModelVersion": "0.1.0"
}
}
```

詳細については、[Amazon Comprehend Medical デベロッパーガイド](#)の「[オントロジーによるバッチ分析のリンク](#)」を参照してください。

- API 詳細については、「[コマンドリファレンス](#)」の[DescribeJob](#)「10CmlInferenceJob」を参照してください。AWS CLI

describe-phi-detection-job

次の例は、describe-phi-detection-job を使用する方法を説明しています。

AWS CLI

PHI検出ジョブを記述するには

次のdescribe-phi-detection-job例は、非同期保護ヘルス情報 (PHI) 検出ジョブに関連付けられたプロパティを示しています。

```
aws comprehendmedical describe-phi-detection-job \
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

出力:

```
{
  "ComprehendMedicalAsyncJobProperties": {
    "JobId": "4750034166536cdb52ffa3295a1b00a3",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
    "EndTime": "2020-03-19T20:45:07.894000+00:00",
    "ExpirationTime": "2020-07-17T20:38:37+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": ""
    },
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
      "S3Key": "867139942017-PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "PHIModelV20190903"
  }
}
```

詳細については、Amazon Comprehend Medical デベロッパーガイド」の「[バッチAPIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePhiDetectionJob](#)」の「」を参照してください。

describe-rx-norm-inference-job

次のコード例は、describe-rx-norm-inference-job を使用する方法を示しています。

AWS CLI

RxNorm 推論ジョブを記述するには

次のdescribe-rx-norm-inference-job例では、指定された job-id を持つリクエストされた推論ジョブのプロパティについて説明します。

```
aws comprehendmedical describe-rx-norm-inference-job \
  --job-id "eg8199877365fc70299089371c043b96"
```

出力:

```
{
  "ComprehendMedicalAsyncJobProperties": {
    "JobId": "g8199877365fc70299089371c043b96",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2020-05-18T21:20:15.614000+00:00",
    "EndTime": "2020-05-18T21:27:07.350000+00:00",
    "ExpirationTime": "2020-09-16T21:20:15+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
      "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "0.0.0"
  }
}
```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーによるバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRxNormInferenceJob](#)」の「」を参照してください。

describe-snomedct-inference-job

次の例は、describe-snomedct-inference-job を使用する方法を説明しています。

AWS CLI

SNOMED CT 推論ジョブを記述するには

次のdescribe-snomedct-inference-job例では、指定された job-id を持つリクエストされた推論ジョブのプロパティについて説明します。

```
aws comprehendmedical describe-snomedct-inference-job \
```

```
--job-id "2630034166536cdb52ffa3295a1b00a7"
```

出力:

```
{
  "ComprehendMedicalAsyncJobProperties": {
    "JobId": "2630034166536cdb52ffa3295a1b00a7",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2021-12-18T21:20:15.614000+00:00",
    "EndTime": "2021-12-18T21:27:07.350000+00:00",
    "ExpirationTime": "2022-05-16T21:20:15+00:00",
    "InputDataConfig": {
      "S3Bucket": "comp-med-input",
      "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "OutputDataConfig": {
      "S3Bucket": "comp-med-output",
      "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "0.1.0"
  }
}
```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーのバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSnomedctInferenceJob](#)」の「」を参照してください。

detect-entities-v2

次のコード例は、detect-entities-v2 を使用する方法を示しています。

AWS CLI

例 1: テキストからエンティティを直接検出するには

次のdetect-entities-v2例は、検出されたエンティティを示し、入力テキストから直接タイプに従ってラベル付けします。


```
aws comprehendmedical detect-entities-v2 \  
  --text "Sleeping trouble on present dosage of Clonidine. Severe rash on face and  
  leg, slightly itchy."
```

出力:

```
{  
  "Id": 0,  
  "BeginOffset": 38,  
  "EndOffset": 47,  
  "Score": 0.9942955374717712,  
  "Text": "Clonidine",  
  "Category": "MEDICATION",  
  "Type": "GENERIC_NAME",  
  "Traits": []  
}
```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[Detect Entities Version 2](#)」を参照してください。

例 2: ファイルパスからエンティティを検出するには

次のdetect-entities-v2例は、検出されたエンティティを示し、ファイルパスのタイプに従ってラベル付けします。

```
aws comprehendmedical detect-entities-v2 \  
  --text file://medical_entities.txt
```

medical_entities.txt の内容:

```
{  
  "Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg,  
  slightly itchy."  
}
```

出力:

```
{  
  "Id": 0,  
  "BeginOffset": 38,
```

```
"EndOffset": 47,  
"Score": 0.9942955374717712,  
"Text": "Clonidine",  
"Category": "MEDICATION",  
"Type": "GENERIC_NAME",  
"Traits": []  
}
```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[Detect Entities Version 2](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の[DetectEntitiesV2](#)を参照してください。
AWS CLI

detect-phi

次のコード例は、detect-phi を使用する方法を示しています。

AWS CLI

例 1: 保護されたヘルス情報 (PHI) をテキストから直接検出するには

次のdetect-phi例では、検出された保護されたヘルス情報 (PHI) エンティティを入力テキストから直接表示します。

```
aws comprehendmedical detect-phi \  
  --text "Patient Carlos Salazar presented with rash on his upper extremities and  
  dry cough. He lives at 100 Main Street, Anytown, USA where he works from his home  
  as a carpenter."
```

出力:

```
{  
  "Entities": [  
    {  
      "Id": 0,  
      "BeginOffset": 8,  
      "EndOffset": 21,  
      "Score": 0.9914507269859314,  
      "Text": "Carlos Salazar",  
      "Category": "PROTECTED_HEALTH_INFORMATION",  
      "Type": "NAME",
```

```
    "Traits": []
  },
  {
    "Id": 1,
    "BeginOffset": 94,
    "EndOffset": 109,
    "Score": 0.871849775314331,
    "Text": "100 Main Street, Anytown, USA",
    "Category": "PROTECTED_HEALTH_INFORMATION",
    "Type": "ADDRESS",
    "Traits": []
  },
  {
    "Id": 2,
    "BeginOffset": 145,
    "EndOffset": 154,
    "Score": 0.8302185535430908,
    "Text": "carpenter",
    "Category": "PROTECTED_HEALTH_INFORMATION",
    "Type": "PROFESSION",
    "Traits": []
  }
],
"ModelVersion": "0.0.0"
}
```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[DetectPHI](#)」を参照してください。

例 2: ファイルパスから直接ヘルス情報 (PHI) を保護する方法を検出するには

次のdetect-phi例は、ファイルパスから検出された保護されたヘルス情報 (PHI) エンティティを示しています。

```
aws comprehendmedical detect-phi \  
  --text file://phi.txt
```

phi.txt の内容:

```
"Patient Carlos Salazar presented with a rash on his upper extremities and a dry cough. He lives at 100 Main Street, Anytown, USA, where he works from his home as a carpenter."
```

出力:

```
{
  "Entities": [
    {
      "Id": 0,
      "BeginOffset": 8,
      "EndOffset": 21,
      "Score": 0.9914507269859314,
      "Text": "Carlos Salazar",
      "Category": "PROTECTED_HEALTH_INFORMATION",
      "Type": "NAME",
      "Traits": []
    },
    {
      "Id": 1,
      "BeginOffset": 94,
      "EndOffset": 109,
      "Score": 0.871849775314331,
      "Text": "100 Main Street, Anytown, USA",
      "Category": "PROTECTED_HEALTH_INFORMATION",
      "Type": "ADDRESS",
      "Traits": []
    },
    {
      "Id": 2,
      "BeginOffset": 145,
      "EndOffset": 154,
      "Score": 0.8302185535430908,
      "Text": "carpenter",
      "Category": "PROTECTED_HEALTH_INFORMATION",
      "Type": "PROFESSION",
      "Traits": []
    }
  ],
  "ModelVersion": "0.0.0"
}
```

詳細については、Amazon Comprehend Medical デベロッパーガイド」の「[DetectPHI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectPhi](#)」の「」を参照してください。

infer-icd10-cm

次の例は、infer-icd10-cm を使用する方法を説明しています。

AWS CLI

例 1: 病状エンティティを検出し、テキストから直接 ICD-10-CM Ontology にリンクするには

次のinfer-icd10-cm例では、検出された病状エンティティにラベルを付け、それらのエンティティを 2019 年版の国際疾病分類臨床修飾 (ICD-10-CM) のコードにリンクします。

```
aws comprehendmedical infer-icd10-cm \  
  --text "The patient complains of abdominal pain, has a long-standing history of  
diabetes treated with Micronase daily."
```

出力:

```
{  
  "Entities": [  
    {  
      "Id": 0,  
      "Text": "abdominal pain",  
      "Category": "MEDICAL_CONDITION",  
      "Type": "DX_NAME",  
      "Score": 0.9475538730621338,  
      "BeginOffset": 28,  
      "EndOffset": 42,  
      "Attributes": [],  
      "Traits": [  
        {  
          "Name": "SYMPTOM",  
          "Score": 0.6724207401275635  
        }  
      ],  
      "ICD10CMConcepts": [  
        {  
          "Description": "Unspecified abdominal pain",  
          "Code": "R10.9",  
          "Score": 0.6904221177101135  
        },  
        {  
          "Description": "Epigastric pain",  
          "Code": "R10.13",
```

```
        "Score": 0.1364113688468933
      },
      {
        "Description": "Generalized abdominal pain",
        "Code": "R10.84",
        "Score": 0.12508003413677216
      },
      {
        "Description": "Left lower quadrant pain",
        "Code": "R10.32",
        "Score": 0.10063883662223816
      },
      {
        "Description": "Lower abdominal pain, unspecified",
        "Code": "R10.30",
        "Score": 0.09933677315711975
      }
    ]
  },
  {
    "Id": 1,
    "Text": "diabetes",
    "Category": "MEDICAL_CONDITION",
    "Type": "DX_NAME",
    "Score": 0.9899052977561951,
    "BeginOffset": 75,
    "EndOffset": 83,
    "Attributes": [],
    "Traits": [
      {
        "Name": "DIAGNOSIS",
        "Score": 0.9258432388305664
      }
    ],
    "ICD10CMConcepts": [
      {
        "Description": "Type 2 diabetes mellitus without complications",
        "Code": "E11.9",
        "Score": 0.7158446311950684
      },
      {
        "Description": "Family history of diabetes mellitus",
        "Code": "Z83.3",
        "Score": 0.5704703330993652
      }
    ]
  }
]
```

```

    },
    {
      "Description": "Family history of other endocrine, nutritional
and metabolic diseases",
      "Code": "Z83.49",
      "Score": 0.19856023788452148
    },
    {
      "Description": "Type 1 diabetes mellitus with ketoacidosis
without coma",
      "Code": "E10.10",
      "Score": 0.13285516202449799
    },
    {
      "Description": "Type 2 diabetes mellitus with hyperglycemia",
      "Code": "E11.65",
      "Score": 0.0993388369679451
    }
  ]
}
],
"ModelVersion": "0.1.0"
}

```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の[ICD10-CM を推測する](#)を参照してください。

例 2: 病状エンティティを検出し、ファイルパスウェイから ICD-10-CM Ontology にリンクするには

次のinfer-icd-10-cm例では、検出された病状エンティティにラベルを付け、それらのエンティティを 2019 年版の国際疾病分類臨床修飾 (ICD-10-CM) のコードにリンクします。

```
aws comprehendmedical infer-icd10-cm \
  --text file://icd10cm.txt
```

icd10cm.txt の内容:

```
{
  "The patient complains of abdominal pain, has a long-standing history of
diabetes treated with Micronase daily."
}
```

出力:

```
{
  "Entities": [
    {
      "Id": 0,
      "Text": "abdominal pain",
      "Category": "MEDICAL_CONDITION",
      "Type": "DX_NAME",
      "Score": 0.9475538730621338,
      "BeginOffset": 28,
      "EndOffset": 42,
      "Attributes": [],
      "Traits": [
        {
          "Name": "SYMPTOM",
          "Score": 0.6724207401275635
        }
      ],
      "ICD10CMConcepts": [
        {
          "Description": "Unspecified abdominal pain",
          "Code": "R10.9",
          "Score": 0.6904221177101135
        },
        {
          "Description": "Epigastric pain",
          "Code": "R10.13",
          "Score": 0.1364113688468933
        },
        {
          "Description": "Generalized abdominal pain",
          "Code": "R10.84",
          "Score": 0.12508003413677216
        },
        {
          "Description": "Left lower quadrant pain",
          "Code": "R10.32",
          "Score": 0.10063883662223816
        },
        {
          "Description": "Lower abdominal pain, unspecified",
          "Code": "R10.30",
          "Score": 0.09933677315711975
        }
      ]
    }
  ]
}
```



```

    }
  ]
},
{
  "Id": 1,
  "Text": "diabetes",
  "Category": "MEDICAL_CONDITION",
  "Type": "DX_NAME",
  "Score": 0.9899052977561951,
  "BeginOffset": 75,
  "EndOffset": 83,
  "Attributes": [],
  "Traits": [
    {
      "Name": "DIAGNOSIS",
      "Score": 0.9258432388305664
    }
  ],
  "ICD10CMConcepts": [
    {
      "Description": "Type 2 diabetes mellitus without complications",
      "Code": "E11.9",
      "Score": 0.7158446311950684
    },
    {
      "Description": "Family history of diabetes mellitus",
      "Code": "Z83.3",
      "Score": 0.5704703330993652
    },
    {
      "Description": "Family history of other endocrine, nutritional
and metabolic diseases",
      "Code": "Z83.49",
      "Score": 0.19856023788452148
    },
    {
      "Description": "Type 1 diabetes mellitus with ketoacidosis
without coma",
      "Code": "E10.10",
      "Score": 0.13285516202449799
    },
    {
      "Description": "Type 2 diabetes mellitus with hyperglycemia",
      "Code": "E11.65",

```

```

        "Score": 0.0993388369679451
      }
    ]
  },
  "ModelVersion": "0.1.0"
}

```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[Infer-ICD10-CM](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の[InferIcd10Cm](#)」を参照してください。

AWS CLI

infer-rx-norm

次の例は、infer-rx-norm を使用する方法を説明しています。

AWS CLI

例 1: 薬剤エンティティを検出し、テキスト RxNorm から直接にリンクするには

次のinfer-rx-norm例は、検出された薬剤エンティティを表示およびラベル付けし、それらのエンティティを National Library of Medicine RxNorm データベースの概念識別子 (Rx CUI) にリンクします。

```

aws comprehendmedical infer-rx-norm \
  --text "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies taking Synthroid."

```

出力:

```

{
  "Entities": [
    {
      "Id": 0,
      "Text": "Levothyroxine",
      "Category": "MEDICATION",
      "Type": "GENERIC_NAME",
      "Score": 0.9996285438537598,
      "BeginOffset": 23,
      "EndOffset": 36,
    }
  ]
}

```

```
"Attributes": [  
  {  
    "Type": "DOSAGE",  
    "Score": 0.9892290830612183,  
    "RelationshipScore": 0.9997978806495667,  
    "Id": 1,  
    "BeginOffset": 37,  
    "EndOffset": 51,  
    "Text": "125 micrograms",  
    "Traits": []  
  },  
  {  
    "Type": "ROUTE_OR_MODE",  
    "Score": 0.9988924860954285,  
    "RelationshipScore": 0.998291552066803,  
    "Id": 2,  
    "BeginOffset": 52,  
    "EndOffset": 56,  
    "Text": "p.o.",  
    "Traits": []  
  },  
  {  
    "Type": "FREQUENCY",  
    "Score": 0.9953463673591614,  
    "RelationshipScore": 0.9999889135360718,  
    "Id": 3,  
    "BeginOffset": 57,  
    "EndOffset": 67,  
    "Text": "once daily",  
    "Traits": []  
  }  
],  
"Traits": [],  
"RxNormConcepts": [  
  {  
    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",  
    "Code": "966224",  
    "Score": 0.9912070631980896  
  },  
  {  
    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",  
    "Code": "966405",  
    "Score": 0.8698278665542603  
  },  
]
```

```

        {
            "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
            "Code": "966191",
            "Score": 0.7448257803916931
        },
        {
            "Description": "levothyroxine",
            "Code": "10582",
            "Score": 0.7050482630729675
        },
        {
            "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxyl]",
            "Code": "966190",
            "Score": 0.6921631693840027
        }
    ]
},
{
    "Id": 4,
    "Text": "Synthroid",
    "Category": "MEDICATION",
    "Type": "BRAND_NAME",
    "Score": 0.9946461319923401,
    "BeginOffset": 86,
    "EndOffset": 95,
    "Attributes": [],
    "Traits": [
        {
            "Name": "NEGATION",
            "Score": 0.5167351961135864
        }
    ],
    "RxNormConcepts": [
        {
            "Description": "Synthroid",
            "Code": "224920",
            "Score": 0.9462039470672607
        },
        {
            "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
[Synthroid]",
            "Code": "966282",

```

```

        "Score": 0.8309829235076904
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
        "Code": "966191",
        "Score": 0.4945160448551178
      },
      {
        "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
[Synthroid]",
        "Code": "966247",
        "Score": 0.3674522042274475
      },
      {
        "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
[Synthroid]",
        "Code": "966158",
        "Score": 0.2588822841644287
      }
    ]
  }
],
  "ModelVersion": "0.0.0"
}

```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[推論 RxNorm](#)」を参照してください。

例 2: ファイルパス RxNorm から薬剤エンティティとリンクを検出するには。

次のinfer-rx-norm例は、検出された薬剤エンティティを表示およびラベル付けし、それらのエンティティを National Library of Medicine RxNorm データベースの概念識別子 (Rx CUI) にリンクします。

```
aws comprehendmedical infer-rx-norm \
  --text file://rxnorm.txt
```

rxnorm.txt の内容:

```
{
  "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies
taking Synthroid."
```

```
}
```

出力:

```
{
  "Entities": [
    {
      "Id": 0,
      "Text": "Levothyroxine",
      "Category": "MEDICATION",
      "Type": "GENERIC_NAME",
      "Score": 0.9996285438537598,
      "BeginOffset": 23,
      "EndOffset": 36,
      "Attributes": [
        {
          "Type": "DOSAGE",
          "Score": 0.9892290830612183,
          "RelationshipScore": 0.9997978806495667,
          "Id": 1,
          "BeginOffset": 37,
          "EndOffset": 51,
          "Text": "125 micrograms",
          "Traits": []
        },
        {
          "Type": "ROUTE_OR_MODE",
          "Score": 0.9988924860954285,
          "RelationshipScore": 0.998291552066803,
          "Id": 2,
          "BeginOffset": 52,
          "EndOffset": 56,
          "Text": "p.o.",
          "Traits": []
        },
        {
          "Type": "FREQUENCY",
          "Score": 0.9953463673591614,
          "RelationshipScore": 0.9999889135360718,
          "Id": 3,
          "BeginOffset": 57,
          "EndOffset": 67,
          "Text": "once daily",
```

```

        "Traits": []
      }
    ],
    "Traits": [],
    "RxNormConcepts": [
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
        "Code": "966224",
        "Score": 0.9912070631980896
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
        "Code": "966405",
        "Score": 0.8698278665542603
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
        "Code": "966191",
        "Score": 0.7448257803916931
      },
      {
        "Description": "levothyroxine",
        "Code": "10582",
        "Score": 0.7050482630729675
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxy1]",
        "Code": "966190",
        "Score": 0.6921631693840027
      }
    ]
  },
  {
    "Id": 4,
    "Text": "Synthroid",
    "Category": "MEDICATION",
    "Type": "BRAND_NAME",
    "Score": 0.9946461319923401,
    "BeginOffset": 86,
    "EndOffset": 95,
    "Attributes": [],
    "Traits": [

```

```
    {
      "Name": "NEGATION",
      "Score": 0.5167351961135864
    }
  ],
  "RxNormConcepts": [
    {
      "Description": "Synthroid",
      "Code": "224920",
      "Score": 0.9462039470672607
    },
    {
      "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
[Synthroid]",
      "Code": "966282",
      "Score": 0.8309829235076904
    },
    {
      "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
      "Code": "966191",
      "Score": 0.4945160448551178
    },
    {
      "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
[Synthroid]",
      "Code": "966247",
      "Score": 0.3674522042274475
    },
    {
      "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
[Synthroid]",
      "Code": "966158",
      "Score": 0.2588822841644287
    }
  ]
}
],
"ModelVersion": "0.0.0"
}
```

詳細については、Amazon Comprehend Medical デベロッパーガイド」の「[推論 RxNorm](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[InferRxNorm](#)」の「」を参照してください。

infer-snomedct

次の例は、infer-snomedct を使用する方法を説明しています。

AWS CLI

例: エンティティを検出し、テキストから直接 SNOMED CT Ontology にリンクするには

次のinfer-snomedct例は、医療エンティティを検出し、2021年3月バージョンの Systematized Nomenclature of Medicine, Clinical Terms (SNOMED CT) の概念にリンクする方法を示しています。

```
aws comprehendmedical infer-snomedct \  
  --text "The patient complains of abdominal pain, has a long-standing history of diabetes treated with Micronase daily."
```

出力:

```
{  
  "Entities": [  
    {  
      "Id": 3,  
      "BeginOffset": 26,  
      "EndOffset": 40,  
      "Score": 0.9598260521888733,  
      "Text": "abdominal pain",  
      "Category": "MEDICAL_CONDITION",  
      "Type": "DX_NAME",  
      "Traits": [  
        {  
          "Name": "SYMPTOM",  
          "Score": 0.6819021701812744  
        }  
      ]  
    },  
    {  
      "Id": 4,  
      "BeginOffset": 73,
```

```
    "EndOffset": 81,
    "Score": 0.9905840158462524,
    "Text": "diabetes",
    "Category": "MEDICAL_CONDITION",
    "Type": "DX_NAME",
    "Traits": [
      {
        "Name": "DIAGNOSIS",
        "Score": 0.9255214333534241
      }
    ]
  },
  {
    "Id": 1,
    "BeginOffset": 95,
    "EndOffset": 104,
    "Score": 0.6371926665306091,
    "Text": "Micronase",
    "Category": "MEDICATION",
    "Type": "BRAND_NAME",
    "Traits": [],
    "Attributes": [
      {
        "Type": "FREQUENCY",
        "Score": 0.9761165380477905,
        "RelationshipScore": 0.9984188079833984,
        "RelationshipType": "FREQUENCY",
        "Id": 2,
        "BeginOffset": 105,
        "EndOffset": 110,
        "Text": "daily",
        "Category": "MEDICATION",
        "Traits": []
      }
    ]
  }
],
"UnmappedAttributes": [],
"ModelVersion": "1.0.0"
}
```

詳細については、Amazon Comprehend Medical デベロッパーガイド」の「[推論SNOMEDCT](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[InferSnomedct](#)」の「」を参照してください。

list-entities-detection-v2-jobs

次の例は、list-entities-detection-v2-jobs を使用する方法を説明しています。

AWS CLI

エンティティ検出ジョブを一覧表示するには

次のlist-entities-detection-v2-jobs例では、現在の非同期検出ジョブを一覧表示します。

```
aws comprehendmedical list-entities-detection-v2-jobs
```

出力:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "ab9887877365fe70299089371c043b96",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
      "EndTime": "2020-03-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-07-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": ""
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "867139942017-EntitiesDetection-ab9887877365fe70299089371c043b96/"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
      "ModelVersion": "DetectEntitiesModelV20190930"
    }
  ]
}
```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[バッチAPIs](#)」を参照してください。

- API 詳細については、AWS CLI コマンドリファレンスの[ListEntitiesDetectionV2Jobs](#)を参照してください。

list-icd10-cm-inference-jobs

次の例は、list-icd10-cm-inference-jobs を使用する方法を説明しています。

AWS CLI

現在のすべての ICD-10-CM 推論ジョブを一覧表示するには

次の例は、list-icd10-cm-inference-jobs オペレーションが現在の非同期 ICD-10-CM バッチ推論ジョブのリストを返す方法を示しています。

```
aws comprehendmedical list-icd10-cm-inference-jobs
```

出力:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "5780034166536cdb52ffa3295a1b00a7",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
      "EndTime": "2020-05-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-09-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::867139942017:role/ComprehendMedicalBatchProcessingRole",
      "ModelVersion": "0.1.0"
    }
  ]
}
```

```
]
}
```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーのバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、「コマンドリファレンス」の[ListId 「10CmlInferenceJobs」](#)を参照してください。AWS CLI

list-phi-detection-jobs

次のコード例は、list-phi-detection-jobs を使用する方法を示しています。

AWS CLI

保護されたヘルス情報 (PHI) 検出ジョブを一覧表示するには

次のlist-phi-detection-jobs例では、現在の保護対象ヘルス情報 (PHI) 検出ジョブを一覧表示します。

```
aws comprehendmedical list-phi-detection-jobs
```

出力:

```
{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "4750034166536cdb52ffa3295a1b00a3",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-03-19T20:38:37.594000+00:00",
      "EndTime": "2020-03-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-07-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": ""
      },
      "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "867139942017-
PHIDetection-4750034166536cdb52ffa3295a1b00a3/"
      },
      "LanguageCode": "en",
```

```

        "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
        "ModelVersion": "PHIModelV20190903"
    }
]
}

```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[バッチAPIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPhiDetectionJobs](#)」の「」を参照してください。

list-rx-norm-inference-jobs

次のコード例は、list-rx-norm-inference-jobs を使用する方法を示しています。

AWS CLI

現在のすべての Rx-Norm 推論ジョブを一覧表示するには

次の例は、現在の非同期 Rx-Norm バッチ推論ジョブのリストを list-rx-norm-inference-jobs で返す方法を示しています。

```
aws comprehendmedical list-rx-norm-inference-jobs
```

出力:

```

{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "4980034166536cfb52gga3295a1b00a3",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
      "EndTime": "2020-05-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-09-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
      },
      "OutputDataConfig": {

```

```

        "S3Bucket": "comp-med-output",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "0.0.0"
}
]
}

```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーのバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRxNormInferenceJobs](#)」の「」を参照してください。

list-snomedct-inference-jobs

次のコード例は、list-snomedct-inference-jobs を使用する方法を示しています。

AWS CLI

すべての SNOMED CT 推論ジョブを一覧表示するには

次の例は、list-snomedct-inference-jobs オペレーションが現在の非同期 SNOMED CT バッチ推論ジョブのリストを返す方法を示しています。

```
aws comprehendmedical list-snomedct-inference-jobs
```

出力:

```

{
  "ComprehendMedicalAsyncJobPropertiesList": [
    {
      "JobId": "5780034166536cdb52ffa3295a1b00a7",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2020-05-19T20:38:37.594000+00:00",
      "EndTime": "2020-05-19T20:45:07.894000+00:00",
      "ExpirationTime": "2020-09-17T20:38:37+00:00",
      "InputDataConfig": {
        "S3Bucket": "comp-med-input",

```

```

        "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "OutputDataConfig": {
        "S3Bucket": "comp-med-output",
        "S3Key": "AKIAIOSFODNN7EXAMPLE"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole",
    "ModelVersion": "0.1.0"
}
]
}

```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーによるバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSnomedctInferenceJobs](#)」の「」を参照してください。

start-entities-detection-v2-job

次の例は、start-entities-detection-v2-job を使用する方法を説明しています。

AWS CLI

エンティティ検出ジョブを開始するには

次のstart-entities-detection-v2-job例では、非同期エンティティ検出ジョブを開始します。

```

aws comprehendmedical start-entities-detection-v2-job \
  --input-data-config "S3Bucket=comp-med-input" \
  --output-data-config "S3Bucket=comp-med-output" \
  --data-access-role-arn arn:aws:iam::867139942017:role/
ComprehendMedicalBatchProcessingRole \
  --language-code en

```

出力:

```

{
  "JobId": "ab9887877365fe70299089371c043b96"
}

```



```
}
```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[バッチAPIs](#)」を参照してください。

- API 詳細については、コマンドリファレンスの[StartEntitiesDetectionV2Job](#)」を参照してください。AWS CLI

start-icd10-cm-inference-job

次の例は、start-icd10-cm-inference-job を使用する方法を説明しています。

AWS CLI

ICD-10-CM 推論ジョブを開始するには

次のstart-icd10-cm-inference-job例では、ICD-10-CM 推論バッチ分析ジョブを開始します。

```
aws comprehendmedical start-icd10-cm-inference-job \  
  --input-data-config "S3Bucket=comp-med-input" \  
  --output-data-config "S3Bucket=comp-med-output" \  
  --data-access-role-arn arn:aws:iam::867139942017:role/  
  ComprehendMedicalBatchProcessingRole \  
  --language-code en
```

出力:

```
{  
  "JobId": "ef7289877365fc70299089371c043b96"  
}
```

詳細については、「[Amazon Comprehend Medical デベロッパーガイド](#)」の「[オントロジーのバッチ分析のリンク](#)」を参照してください。Amazon Comprehend

- API 詳細については、「[コマンドリファレンス](#)」の[StartIcd「10CmInferenceJob](#)」を参照してください。AWS CLI

start-phi-detection-job

次のコード例は、start-phi-detection-job を使用する方法を示しています。

AWS CLI

PHI検出ジョブを開始するには

次のstart-phi-detection-job例では、非同期PHIエンティティ検出ジョブを開始します。

```
aws comprehendmedical start-phi-detection-job \  
  --input-data-config "S3Bucket=comp-med-input" \  
  --output-data-config "S3Bucket=comp-med-output" \  
  --data-access-role-arn arn:aws:iam::867139942017:role/  
  ComprehendMedicalBatchProcessingRole \  
  --language-code en
```

出力:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

詳細については、Amazon Comprehend Medical デベロッパーガイド」の「[バッチAPIs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartPhiDetectionJob](#)」の「」を参照してください。

start-rx-norm-inference-job

次のコード例は、start-rx-norm-inference-job を使用する方法を示しています。

AWS CLI

RxNorm 推論ジョブを開始するには

次のstart-rx-norm-inference-job例では、RxNorm 推論バッチ分析ジョブを開始します。

```
aws comprehendmedical start-rx-norm-inference-job \  
  --input-data-config "S3Bucket=comp-med-input" \  
  --output-data-config "S3Bucket=comp-med-output" \  
  --data-access-role-arn arn:aws:iam::867139942017:role/  
  ComprehendMedicalBatchProcessingRole \  
  --language-code en
```

出力:

```
{
  "JobId": "eg8199877365fc70299089371c043b96"
}
```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーのバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[StartRxNormInferenceJob](#)」の「」を参照してください。

start-snomedct-inference-job

次のコード例は、start-snomedct-inference-job を使用する方法を示しています。

AWS CLI

SNOMED CT 推論ジョブを開始するには

次のstart-snomedct-inference-job例では、SNOMEDCT 推論バッチ分析ジョブを開始します。

```
aws comprehendmedical start-snomedct-inference-job \
  --input-data-config "S3Bucket=comp-med-input" \
  --output-data-config "S3Bucket=comp-med-output" \
  --data-access-role-arn arn:aws:iam::867139942017:role/
  ComprehendMedicalBatchProcessingRole \
  --language-code en
```

出力:

```
{
  "JobId": "dg7289877365fc70299089371c043b96"
}
```

詳細については、[Amazon Comprehend Medical デベロッパーガイド」の「オントロジーによるバッチ分析のリンク」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartSnomedctInferenceJob](#)」の「」を参照してください。

stop-entities-detection-v2-job

次の例は、stop-entities-detection-v2-job を使用する方法を説明しています。

AWS CLI

エンティティ検出ジョブを停止するには

次のstop-entities-detection-v2-job例では、非同期エンティティ検出ジョブを停止しています。

```
aws comprehendmedical stop-entities-detection-v2-job \  
  --job-id "ab9887877365fe70299089371c043b96"
```

出力:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

詳細については、「Amazon Comprehend Medical デベロッパーガイド」の「[バッチAPIs](#)」を参照してください。

- API 詳細については、コマンドリファレンスの[StopEntitiesDetectionV2Job](#)」を参照してください。AWS CLI

stop-icd10-cm-inference-job

次の例は、stop-icd10-cm-inference-job を使用する方法を説明しています。

AWS CLI

ICD-10-CM 推論ジョブを停止するには

次のstop-icd10-cm-inference-job例では、ICD-10-CM 推論バッチ分析ジョブを停止しています。

```
aws comprehendmedical stop-icd10-cm-inference-job \  
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

出力:

```
{
  "JobId": "ef7289877365fc70299089371c043b96",
}
```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーのバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、「コマンドリファレンス」の[StopIcd「10CmlInferenceJob」](#)を参照してください。AWS CLI

stop-phi-detection-job

次のコード例は、stop-phi-detection-job を使用する方法を示しています。

AWS CLI

保護されたヘルス情報 (PHI) 検出ジョブを停止するには

次のstop-phi-detection-job例では、非同期で保護されたヘルス情報 (PHI) 検出ジョブを停止しています。

```
aws comprehendmedical stop-phi-detection-job \
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

出力:

```
{
  "JobId": "ab9887877365fe70299089371c043b96"
}
```

詳細については、Amazon Comprehend Medical デベロッパーガイド」の[「バッチAPIs」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopPhiDetectionJob](#)」の「」を参照してください。

stop-rx-norm-inference-job

次のコード例は、stop-rx-norm-inference-job を使用する方法を示しています。

AWS CLI

RxNorm 推論ジョブを停止するには

次のstop-rx-norm-inference-job例では、ICD-10-CM 推論バッチ分析ジョブを停止しています。

```
aws comprehendmedical stop-rx-norm-inference-job \  
  --job-id "eg8199877365fc70299089371c043b96"
```

出力:

```
{  
  "JobId": "eg8199877365fc70299089371c043b96",  
}
```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーのバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[StopRxNormInferenceJob](#)」の「」を参照してください。

stop-snomedct-inference-job

次の例は、stop-snomedct-inference-job を使用する方法を説明しています。

AWS CLI

SNOMED CT 推論ジョブを停止するには

次のstop-snomedct-inference-job例では、SNOMEDCT 推論バッチ分析ジョブを停止しています。

```
aws comprehendmedical stop-snomedct-inference-job \  
  --job-id "8750034166436cdb52ffa3295a1b00a1"
```

出力:

```
{  
  "JobId": "8750034166436cdb52ffa3295a1b00a1",  
}
```

```
}
```

詳細については、[「Amazon Comprehend Medical デベロッパーガイド」の「オントロジーのバッチ分析のリンク」](#)を参照してください。Amazon Comprehend

- API 詳細については、AWS CLI 「コマンドリファレンス[StopSnomedctInferenceJob](#)」の「」を参照してください。

AWS Config を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Config。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-config-rule

次のコード例は、delete-config-rule を使用する方法を示しています。

AWS CLI

AWS Config ルールを削除するには

次のコマンドは、 という名前の AWS Config ルールを削除しますMyConfigRule。

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConfigRule](#)」の「」を参照してください。

delete-delivery-channel

次の例は、delete-delivery-channel を使用する方法を説明しています。

AWS CLI

配信チャンネルを削除するには

次のコマンドは、デフォルトの配信チャンネルを削除します。

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDeliveryChannel](#)」の「」を参照してください。

delete-evaluation-results

次の例は、delete-evaluation-results を使用する方法を説明しています。

AWS CLI

評価結果を手動で削除するには

次のコマンドは、AWS マネージドルール s3- の現在の評価結果を削除します bucket-versioning-enabled。

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-enabled
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEvaluationResults](#)」の「」を参照してください。

deliver-config-snapshot

次の例は、deliver-config-snapshot を使用する方法を説明しています。

AWS CLI

設定スナップショットを配信するには

次のコマンドは、デフォルトの配信チャンネルに属する Amazon S3 バケットに設定スナップショットを配信します。


```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

出力:

```
{
  "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeliverConfigSnapshot](#)」の「」を参照してください。

describe-compliance-by-config-rule

次の例は、describe-compliance-by-config-rule を使用する方法を説明しています。

AWS CLI

AWS Config ルールのコンプライアンス情報を取得するには

次のコマンドは、1 つ以上の AWS リソースによって違反された各 AWS Config ルールのコンプライアンス情報を返します。

```
aws configservice describe-compliance-by-config-rule --compliance-
types NON_COMPLIANT
```

出力では、各CappedCount属性の値は、関連するルールに準拠していないリソースの数を示します。例えば、次の出力は、3 つのリソースが という名前のルールに準拠していないことを示していますInstanceTypesAreT2micro。

出力:

```
{
  "ComplianceByConfigRules": [
    {
      "Compliance": {
        "ComplianceContributorCount": {
          "CappedCount": 3,
          "CapExceeded": false
        },
        "ComplianceType": "NON_COMPLIANT"
      }
    }
  ]
}
```

```

    },
    "ConfigRuleName": "InstanceTypesAreT2micro"
  },
  {
    "Compliance": {
      "ComplianceContributorCount": {
        "CappedCount": 10,
        "CapExceeded": false
      },
      "ComplianceType": "NON_COMPLIANT"
    },
    "ConfigRuleName": "RequiredTagsForVolumes"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeComplianceByConfigRule](#)」の「」を参照してください。

describe-compliance-by-resource

次の例は、describe-compliance-by-resource を使用する方法を説明しています。

AWS CLI

AWS リソースのコンプライアンス情報を取得するには

次のコマンドは、AWS Config によって記録され、1 つ以上のルールに違反する各 EC2 インスタンスのコンプライアンス情報を返します。

```
aws configservice describe-compliance-by-resource --resource-type AWS::EC2::Instance
--compliance-types NON_COMPLIANT
```

出力では、各 CappedCount 属性の値は、リソースが違反するルールの数を示します。例えば、次の出力は、インスタンスが 2 つのルール i-1a2b3c4d に違反していることを示しています。

出力:

```

{
  "ComplianceByResources": [
    {

```

```
    "ResourceType": "AWS::EC2::Instance",
    "ResourceId": "i-1a2b3c4d",
    "Compliance": {
      "ComplianceContributorCount": {
        "CappedCount": 2,
        "CapExceeded": false
      },
      "ComplianceType": "NON_COMPLIANT"
    }
  },
  {
    "ResourceType": "AWS::EC2::Instance",
    "ResourceId": "i-2a2b3c4d ",
    "Compliance": {
      "ComplianceContributorCount": {
        "CappedCount": 3,
        "CapExceeded": false
      },
      "ComplianceType": "NON_COMPLIANT"
    }
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeComplianceByResource](#)」の「」を参照してください。

describe-config-rule-evaluation-status

次の例は、describe-config-rule-evaluation-status を使用する方法を説明しています。

AWS CLI

AWS Config ルールのステータス情報を取得するには

次のコマンドは、 という名前の AWS Config ルールのステータス情報を返します MyConfigRule。

```
aws configservice describe-config-rule-evaluation-status --config-rule-names MyConfigRule
```

出力:

```
{
  "ConfigRulesEvaluationStatus": [
    {
      "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/
config-rule-abcdef",
      "FirstActivatedTime": 1450311703.844,
      "ConfigRuleId": "config-rule-abcdef",
      "LastSuccessfulInvocationTime": 1450314643.156,
      "ConfigRuleName": "MyConfigRule"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeConfigRuleEvaluationStatus](#)」の「」を参照してください。

describe-config-rules

次のコード例は、describe-config-rules を使用する方法を示しています。

AWS CLI

AWS Config ルールの詳細を取得するには

次のコマンドは、`InstanceTypesAreT2micro` という名前の AWS Config ルールの詳細を返します。

```
aws configservice describe-config-rules --config-rule-names InstanceTypesAreT2micro
```

出力:

```
{
  "ConfigRules": [
    {
      "ConfigRuleState": "ACTIVE",
      "Description": "Evaluates whether EC2 instances are the t2.micro type.",
      "ConfigRuleName": "InstanceTypesAreT2micro",
      "ConfigRuleArn": "arn:aws:config:us-east-1:123456789012:config-rule/
config-rule-abcdef",
      "Source": {
        "Owner": "CUSTOM_LAMBDA",
```

```
    "SourceIdentifier": "arn:aws:lambda:us-
east-1:123456789012:function:InstanceTypeCheck",
    "SourceDetails": [
      {
        "EventSource": "aws.config",
        "MessageType": "ConfigurationItemChangeNotification"
      }
    ],
    "InputParameters": "{\"desiredInstanceType\": \"t2.micro\"}",
    "Scope": {
      "ComplianceResourceTypes": [
        "AWS::EC2::Instance"
      ]
    },
    "ConfigRuleId": "config-rule-abcdef"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeConfigRules](#)」の「」を参照してください。

describe-configuration-recorder-status

次のコード例は、describe-configuration-recorder-status を使用する方法を示しています。

AWS CLI

設定レコーダーのステータス情報を取得するには

次のコマンドは、デフォルトの設定レコーダーのステータスを返します。

```
aws configservice describe-configuration-recorder-status
```

出力:

```
{
  "ConfigurationRecordersStatus": [
    {
```

```
        "name": "default",
        "lastStatus": "SUCCESS",
        "recording": true,
        "lastStatusChangeTime": 1452193834.344,
        "lastStartTime": 1441039997.819,
        "lastStopTime": 1441039992.835
      }
    ]
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeConfigurationRecorderStatus](#)」の「」を参照してください。

describe-configuration-recorders

次の例は、describe-configuration-recorders を使用する方法を説明しています。

AWS CLI

設定レコーダーの詳細を取得するには

次のコマンドは、デフォルトの設定レコーダーに関する詳細を返します。

```
aws configservice describe-configuration-recorders
```

出力:

```
{
  "ConfigurationRecorders": [
    {
      "recordingGroup": {
        "allSupported": true,
        "resourceTypes": [],
        "includeGlobalResourceTypes": true
      },
      "roleARN": "arn:aws:iam::123456789012:role/config-ConfigRole-A1B2C3D4E5F6",
      "name": "default"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeConfigurationRecorders](#)」の「」を参照してください。

describe-delivery-channel-status

次の例は、describe-delivery-channel-status を使用する方法を説明しています。

AWS CLI

配信チャネルのステータス情報を取得するには

次のコマンドは、配信チャネルのステータスを返します。

```
aws configservice describe-delivery-channel-status
```

出力:

```
{
  "DeliveryChannelsStatus": [
    {
      "configStreamDeliveryInfo": {
        "lastStatusChangeTime": 1452193834.381,
        "lastStatus": "SUCCESS"
      },
      "configHistoryDeliveryInfo": {
        "lastSuccessfulTime": 1450317838.412,
        "lastStatus": "SUCCESS",
        "lastAttemptTime": 1450317838.412
      },
      "configSnapshotDeliveryInfo": {
        "lastSuccessfulTime": 1452185597.094,
        "lastStatus": "SUCCESS",
        "lastAttemptTime": 1452185597.094
      },
      "name": "default"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDeliveryChannelStatus](#)」の「」を参照してください。

describe-delivery-channels

次の例は、describe-delivery-channels を使用する方法を説明しています。

AWS CLI

配信チャネルの詳細を取得するには

次のコマンドは、配信チャネルに関する詳細を返します。

```
aws configservice describe-delivery-channels
```

出力:

```
{
  "DeliveryChannels": [
    {
      "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
      "name": "default",
      "s3BucketName": "config-bucket-123456789012"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDeliveryChannels](#)」の「」を参照してください。

get-compliance-details-by-config-rule

次のコード例は、get-compliance-details-by-config-rule を使用する方法を示しています。

AWS CLI

AWS Config ルールの評価結果を取得するには

次のコマンドは、`InstanceTypesAreT2micro` という名前の AWS Config ルールに準拠していないすべてのリソースの評価結果を返します。

```
aws configservice get-compliance-details-by-config-rule --config-rule-name InstanceTypesAreT2micro --compliance-types NON_COMPLIANT
```


出力:

```
{
  "EvaluationResults": [
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-1a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314645.261,
      "ConfigRuleInvokedTime": 1450314642.948,
      "ComplianceType": "NON_COMPLIANT"
    },
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-2a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314645.18,
      "ConfigRuleInvokedTime": 1450314642.902,
      "ComplianceType": "NON_COMPLIANT"
    },
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-3a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314643.346,
      "ConfigRuleInvokedTime": 1450314643.124,
      "ComplianceType": "NON_COMPLIANT"
    }
  ]
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetComplianceDetailsByConfigRule](#)」の「」を参照してください。

get-compliance-details-by-resource

次の例は、`get-compliance-details-by-resource` を使用する方法を説明しています。

AWS CLI

AWS リソースの評価結果を取得するには

次のコマンドは、EC2インスタンスが準拠*i-1a2b3c4d*していない各ルールの評価結果を返します。

```
aws configservice get-compliance-details-by-resource --resource-type AWS::EC2::Instance --resource-id i-1a2b3c4d --compliance-types NON_COMPLIANT
```

出力:

```
{
  "EvaluationResults": [
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-1a2b3c4d",
          "ConfigRuleName": "InstanceTypesAreT2micro"
        }
      },
      "ResultRecordedTime": 1450314643.288,
      "ConfigRuleInvokedTime": 1450314643.034,
      "ComplianceType": "NON_COMPLIANT"
    },
    {
      "EvaluationResultIdentifier": {
        "OrderingTimestamp": 1450314635.065,
        "EvaluationResultQualifier": {
          "ResourceType": "AWS::EC2::Instance",
          "ResourceId": "i-1a2b3c4d",
```

```
        "ConfigRuleName": "RequiredTagForEC2Instances"
      }
    },
    "ResultRecordedTime": 1450314645.261,
    "ConfigRuleInvokedTime": 1450314642.948,
    "ComplianceType": "NON_COMPLIANT"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetComplianceDetailsByResource](#)」の「」を参照してください。

get-compliance-summary-by-config-rule

次のコード例は、get-compliance-summary-by-config-rule を使用する方法を示しています。

AWS CLI

AWS Config ルールのコンプライアンス概要を取得するには

次のコマンドは、準拠しているルールの数と非準拠のルール数を返します。

```
aws configservice get-compliance-summary-by-config-rule
```

出力では、各CappedCount属性の値は、準拠または非準拠のルール数を示します。

出力:

```
{
  "ComplianceSummary": {
    "NonCompliantResourceCount": {
      "CappedCount": 3,
      "CapExceeded": false
    },
    "ComplianceSummaryTimestamp": 1452204131.493,
    "CompliantResourceCount": {
      "CappedCount": 2,
      "CapExceeded": false
    }
  }
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetComplianceSummaryByConfigRule](#)」の「」を参照してください。

get-compliance-summary-by-resource-type

次の例は、get-compliance-summary-by-resource-type を使用する方法を説明しています。

AWS CLI

すべてのリソースタイプのコンプライアンス概要を取得するには

次のコマンドは、非準拠の AWS リソースの数と準拠している数を返します。

```
aws configservice get-compliance-summary-by-resource-type
```

出力では、各CappedCount属性の値は、準拠または非準拠のリソースの数を示します。

出力:

```
{
  "ComplianceSummariesByResourceType": [
    {
      "ComplianceSummary": {
        "NonCompliantResourceCount": {
          "CappedCount": 16,
          "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1453237464.543,
        "CompliantResourceCount": {
          "CappedCount": 10,
          "CapExceeded": false
        }
      }
    }
  ]
}
```

特定のリソースタイプのコンプライアンス概要を取得するには

次のコマンドは、非準拠のEC2インスタンスの数と準拠している数を返します。

```
aws configservice get-compliance-summary-by-resource-type --resource-  
types AWS::EC2::Instance
```

出力では、各CappedCount属性の値は、準拠または非準拠のリソースの数を示します。

出力:

```
{  
  "ComplianceSummariesByResourceType": [  
    {  
      "ResourceType": "AWS::EC2::Instance",  
      "ComplianceSummary": {  
        "NonCompliantResourceCount": {  
          "CappedCount": 3,  
          "CapExceeded": false  
        },  
        "ComplianceSummaryTimestamp": 1452204923.518,  
        "CompliantResourceCount": {  
          "CappedCount": 7,  
          "CapExceeded": false  
        }  
      }  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetComplianceSummaryByResourceType](#)」の「」を参照してください。

get-resource-config-history

次の例は、get-resource-config-history を使用する方法を説明しています。

AWS CLI

AWS リソースの設定履歴を取得するには

次のコマンドは、ID が のEC2インスタンスの設定項目のリストを返しますi-1a2b3c4d。

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --  
resource-id i-1a2b3c4d
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceConfigHistory](#)」の「」を参照してください。

get-status

次の例は、get-status を使用する方法を説明しています。

AWS CLI

AWS Config のステータスを取得するには

次のコマンドは、配信チャンネルと設定レコーダーのステータスを返します。

```
aws configservice get-status
```

出力:

```
Configuration Recorders:

name: default
recorder: ON
last status: SUCCESS

Delivery Channels:

name: default
last stream delivery status: SUCCESS
last history delivery status: SUCCESS
last snapshot delivery status: SUCCESS
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStatus](#)」の「」を参照してください。

list-discovered-resources

次のコード例は、list-discovered-resources を使用する方法を示しています。

AWS CLI

AWS Config が検出したリソースを一覧表示するには

次のコマンドは、AWS Config が検出したEC2インスタンスを一覧表示します。

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Instance
```

出力:

```
{
  "resourceIdentifiers": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-1a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-2a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-3a2b3c4d"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListDiscoveredResources](#)」の「」を参照してください。

put-config-rule

次のコード例は、put-config-rule を使用する方法を示しています。

AWS CLI

AWS マネージド Config ルールを追加するには

次のコマンドは、AWS マネージド Config ルールを追加するJSONコードを提供します。

```
aws configservice put-config-rule --config-rule file://  
RequiredTagsForEC2Instances.json
```

RequiredTagsForEC2Instances.json は、ルール設定を含むJSONファイルです。

```
{
```

```

    "ConfigRuleName": "RequiredTagsForEC2Instances",
    "Description": "Checks whether the CostCenter and Owner tags are applied to EC2
instances.",
    "Scope": {
      "ComplianceResourceTypes": [
        "AWS::EC2::Instance"
      ]
    },
    "Source": {
      "Owner": "AWS",
      "SourceIdentifier": "REQUIRED_TAGS"
    },
    "InputParameters": "{\"tag1Key\":\"CostCenter\",\"tag2Key\":\"Owner\"}"
  }
}

```

ComplianceResourceTypes 属性の場合、このJSONコードはスコープを AWS::EC2::Instance タイプのリソースに制限するため、AWS Config はルールに対して EC2 インスタンスのみを評価します。このルールはマネージドルールであるため、Owner 属性は AWS に設定され、SourceIdentifier 属性はルール識別子 REQUIRED_TAGS に設定されます。InputParameters 属性には、ルールに必要なタグキー、および CostCenter と Owner が指定されます。

コマンドが成功すると、AWS Config は出力を返しません。ルール設定を確認するには、コマンドを実行し describe-config-rules、ルール名を指定します。

カスタマーマネージド Config ルールを追加するには

次のコマンドは、カスタマーマネージド Config ルールを追加する JSON コードを提供します。

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

InstanceTypesAreT2micro.json は、ルール設定を含む JSON ファイルです。

```

{
  "ConfigRuleName": "InstanceTypesAreT2micro",
  "Description": "Evaluates whether EC2 instances are the t2.micro type.",
  "Scope": {
    "ComplianceResourceTypes": [
      "AWS::EC2::Instance"
    ]
  },
  "Source": {

```



```

    "Owner": "CUSTOM_LAMBDA",
    "SourceIdentifier": "arn:aws:lambda:us-
east-1:123456789012:function:InstanceTypeCheck",
    "SourceDetails": [
      {
        "EventSource": "aws.config",
        "MessageType": "ConfigurationItemChangeNotification"
      }
    ]
  },
  "InputParameters": "{\"desiredInstanceType\":\"t2.micro\"}"
}

```

ComplianceResourceTypes 属性の場合、このJSONコードはスコープを AWS::EC2::Instance タイプのリソースに制限するため、AWS Config はルールに対して EC2 インスタンスのみを評価します。このルールはカスタマー管理ルールであるため、Owner 属性はに設定され CUSTOM_LAMBDA、SourceIdentifier 属性は AWS Lambda 関数 ARN のに設定されます。SourceDetails オブジェクトは必須です。InputParameters 属性に指定されたパラメータは、AWS Config が呼び出してルールに対してリソースを評価するときに AWS Lambda 関数に渡されます。

コマンドが成功すると、AWS Config は出力を返しません。ルール設定を確認するには、コマンドを実行し describe-config-rules、ルール名を指定します。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutConfigRule](#)」の「」を参照してください。

put-configuration-recorder

次のコード例は、put-configuration-recorder を使用する方法を示しています。

AWS CLI

例 1: サポートされているすべてのリソースを記録するには

次のコマンドは、グローバルリソースタイプを含む、サポートされているすべてのリソースタイプに対する変更を追跡する設定レコーダーを作成します。

```

aws configservice put-configuration-recorder \
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/
  config-role \

```

```
--recording-group allSupported=true,includeGlobalResourceTypes=true
```

コマンドが成功すると、AWS Config は出力を返しません。設定レコーダーの設定を確認するには、コマンドを実行します `describe-configuration-recorders`。

例 2: 特定のタイプのリソースを記録するには

次のコマンドは、`--recording-group` オプションの JSON ファイルで指定されているリソースタイプのみに対する変更を追跡する設定レコーダーを作成します。

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
  config-role \  
  --recording-group file://recordingGroup.json
```

`recordingGroup.json` は、AWS Config が記録するリソースのタイプを指定する JSON ファイルです。

```
{  
  "allSupported": false,  
  "includeGlobalResourceTypes": false,  
  "resourceTypes": [  
    "AWS::EC2::EIP",  
    "AWS::EC2::Instance",  
    "AWS::EC2::NetworkAcl",  
    "AWS::EC2::SecurityGroup",  
    "AWS::CloudTrail::Trail",  
    "AWS::EC2::Volume",  
    "AWS::EC2::VPC",  
    "AWS::IAM::User",  
    "AWS::IAM::Policy"  
  ]  
}
```

`resourceTypes` キーにリソースタイプを指定する前に、`allSupported` および `includeGlobalResource` タイプオプションを `false` または省略するように設定する必要があります。

コマンドが成功すると、AWS Config は出力を返しません。設定レコーダーの設定を確認するには、コマンドを実行します `describe-configuration-recorders`。

例 3: 特定のタイプのリソースを除く、サポートされているすべてのリソースを選択するには

次のコマンドは、`--recording-group` オプションの JSON ファイルで指定されているリソースタイプを除く、現在および将来のサポートされているすべてのリソースタイプに対する変更を追跡する設定レコーダーを作成します。

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
config-role \  
  --recording-group file://recordingGroup.json
```

`recordingGroup.json` は、AWS Config が記録するリソースのタイプを指定する JSON ファイルです。

```
{  
  "allSupported": false,  
  "exclusionByResourceTypes": {  
    "resourceTypes": [  
      "AWS::Redshift::ClusterSnapshot",  
      "AWS::RDS::DBClusterSnapshot",  
      "AWS::CloudFront::StreamingDistribution"  
    ]  
  },  
  "includeGlobalResourceTypes": false,  
  "recordingStrategy": {  
    "useOnly": "EXCLUSION_BY_RESOURCE_TYPES"  
  },  
}
```

記録から除外するリソースタイプを指定する前に、1) `allSupported` および `includeGlobalResourceTypes` オプションを `false` または省略するように設定する必要があります。2) の `useOnly` フィールドを `EXCLUSION_BY_RESOURCE_TYPES` に設定する必要があります。

コマンドが成功すると、AWS Config は出力を返しません。設定レコーダーの設定を確認するには、コマンドを実行します `describe-configuration-recorders`。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutConfigurationRecorder](#)」の「」を参照してください。

put-delivery-channel

次のコード例は、`put-delivery-channel` を使用する方法を示しています。

AWS CLI

配信チャネルを作成するには

次のコマンドは、配信チャネルの設定をJSONコードとして提供します。

```
aws configservice put-delivery-channel --delivery-channel file://  
deliveryChannel.json
```

deliveryChannel.json ファイルは配信チャネル属性を指定します。

```
{  
  "name": "default",  
  "s3BucketName": "config-bucket-123456789012",  
  "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",  
  "configSnapshotDeliveryProperties": {  
    "deliveryFrequency": "Twelve_Hours"  
  }  
}
```

この例では以下の属性を設定します。

name - 配信チャネルの名前。デフォルトでは、AWS Config は新しい配信チャネルdefaultに名前を割り当てます。put-delivery-channel コマンドで配信チャネル名を更新することはできません。名前を変更する手順については、「配信チャネルの名前変更」を参照してください。**s3BucketName**- AWS Config が設定スナップショットと設定履歴ファイルを配信する Amazon S3 バケットの名前。別の AWS アカウントに属するバケットを指定する場合、そのバケットには AWS Config へのアクセス許可を付与するポリシーが必要です。詳細については、Amazon S3 バケットのアクセス許可を参照してください。

snsTopicARN - AWS Config が設定変更に関する通知を送信する Amazon SNSトピックの Amazon リソースネーム (ARN)。別のアカウントからトピックを選択した場合、トピックには AWS Config にアクセス許可を付与するポリシーが必要です。詳細については、「Amazon SNS トピックのアクセス許可」を参照してください。

configSnapshotDeliveryProperties - AWS Config が設定スナップショットを配信する頻度と、定期的な Config ルールの評価を呼び出す頻度を設定する **deliveryFrequency** 属性が含まれます。

コマンドが成功すると、AWS Config は出力を返しません。配信チャネルの設定を確認するには、コマンドを実行します describe-delivery-channels。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutDeliveryChannel](#)」の「」を参照してください。

start-config-rules-evaluation

次のコード例は、start-config-rules-evaluation を使用する方法を示しています。

AWS CLI

AWS Config ルールのオンデマンド評価を実行するには

次のコマンドは、2 つの AWS マネージドルールの評価を開始します。

```
aws configservice start-config-rules-evaluation --config-rule-names s3-bucket-versioning-enabled cloudtrail-enabled
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartConfigRulesEvaluation](#)」の「」を参照してください。

start-configuration-recorder

次のコード例は、start-configuration-recorder を使用する方法を示しています。

AWS CLI

設定レコーダーを起動するには

次のコマンドは、デフォルトの設定レコーダーを起動します。

```
aws configservice start-configuration-recorder --configuration-recorder-name default
```

コマンドが成功すると、AWS Config は出力を返しません。AWS Config がリソースを記録していることを確認するには、get-status コマンドを実行します。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartConfigurationRecorder](#)」の「」を参照してください。

stop-configuration-recorder

次の例は、stop-configuration-recorder を使用する方法を説明しています。

AWS CLI

設定レコーダーを停止するには

次のコマンドは、デフォルトの設定レコーダーを停止します。

```
aws configservice stop-configuration-recorder --configuration-recorder-name default
```

コマンドが成功すると、AWS Config は出力を返しません。AWS Config がリソースを記録していないことを確認するには、get-status コマンドを実行します。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopConfigurationRecorder](#)」の「」を参照してください。

subscribe

次のコード例は、subscribe を使用する方法を示しています。

AWS CLI

AWS Config をサブスクライブするには

次のコマンドは、デフォルトの配信チャンネルと設定レコーダーを作成します。コマンドは、AWS Config が設定情報を配信する Amazon S3 バケットと Amazon SNS トピックも指定します。

```
aws configservice subscribe --s3-bucket config-bucket-123456789012  
--sns-topic arn:aws:sns:us-east-1:123456789012:config-topic --iam-  
role arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6
```

出力:

```
Using existing S3 bucket: config-bucket-123456789012  
Using existing SNS topic: arn:aws:sns:us-east-1:123456789012:config-topic  
Subscribe succeeded:
```

```
Configuration Recorders: [  
  {  
    "recordingGroup": {  
      "allSupported": true,  
      "resourceTypes": [],  
      "includeGlobalResourceTypes": false
```

```
    },
    "roleARN": "arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6",
    "name": "default"
  }
]

Delivery Channels: [
  {
    "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
    "name": "default",
    "s3BucketName": "config-bucket-123456789012"
  }
]
```

- API 詳細については、「コマンドリファレンス」の[「サブスクライブ」](#)を参照してください。
AWS CLI

を使用した Amazon Connect の例 AWS CLI

次のコード例は、Amazon Connect AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-user

次の例は、create-user を使用する方法を説明しています。

AWS CLI

ユーザーを作成するには

次のcreate-user例では、指定された属性を持つユーザーを指定された Amazon Connect インスタンスに追加します。

```
aws connect create-user \  
  --username Mary \  
  --password Pass@Word1 \  
  --identity-info FirstName=Mary,LastName=Major \  
  --phone-  
config PhoneType=DESK_PHONE,AutoAccept=true,AfterContactWorkTimeLimit=60,DeskPhoneNumber=  
+15555551212 \  
  --security-profile-id 12345678-1111-2222-aaaa-a1b2c3d4f5g7 \  
  --routing-profile-id 87654321-9999-3434-abcd-x1y2z3a1b2c3 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "UserId": "87654321-2222-1234-1234-111234567891",  
  "UserArn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111/agent/87654321-2222-1234-1234-111234567891"  
}
```

詳細については、「Amazon Connect 管理者ガイド」の「[ユーザーの追加](#)」を参照してください。Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUser](#)」の「」を参照してください。

delete-user

次の例は、delete-user を使用する方法を説明しています。

AWS CLI

ユーザーを削除するには

次のdelete-user例では、指定された Amazon Connect インスタンスから指定されたユーザーを削除します。

```
aws connect delete-user \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --user-id 87654321-2222-1234-1234-111234567891
```



```
--user-id 87654321-2222-1234-1234-111234567891
```

このコマンドでは何も出力されません。

詳細については、「Amazon Connect 管理者ガイド」の「[ユーザーの管理](#)」を参照してください。 Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUser](#)」の「」を参照してください。

describe-user-hierarchy-group

次のコード例は、describe-user-hierarchy-group を使用する方法を示しています。

AWS CLI

階層グループの詳細を表示するには

次のdescribe-user-hierarchy-group例では、指定された Amazon Connect 階層グループの詳細を表示します。

```
aws connect describe-user-hierarchy-group \  
  --hierarchy-group-id 12345678-1111-2222-800e-aaabbb555gg \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "HierarchyGroup": {  
    "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",  
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group/12345678-1111-2222-800e-a2b3c4d5f6g7",  
    "Name": "Example Corporation",  
    "LevelId": "1",  
    "HierarchyPath": {  
      "LevelOne": {  
        "Id": "abcdefgh-3333-4444-8af3-201123456789",  
        "Arn": "arn:aws:connect:us-west-2:123456789012:instance/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group/abcdefgh-3333-4444-8af3-201123456789",  
        "Name": "Example Corporation"  
      }  
    }  
  }  
}
```

```
}  
}
```

詳細については、「Amazon Connect 管理者ガイド」の「[エージェント階層の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUserHierarchyGroup](#)」の「」を参照してください。

describe-user-hierarchy-structure

次のコード例は、describe-user-hierarchy-structure を使用する方法を示しています。

AWS CLI

階層構造の詳細を表示するには

次のdescribe-user-hierarchy-structure例では、指定された Amazon Connect インスタンスの階層構造の詳細を表示します。

```
aws connect describe-user-hierarchy-group \  
--instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "HierarchyStructure": {  
    "LevelOne": {  
      "Id": "12345678-1111-2222-800e-aaabbb555gg",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/1",  
      "Name": "Corporation"  
    },  
    "LevelTwo": {  
      "Id": "87654321-2222-3333-ac99-123456789102",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/2",  
      "Name": "Services Division"  
    },  
    "LevelThree": {  
      "Id": "abcdefgh-3333-4444-8af3-201123456789",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/agent-group-level/3",  
    }  
  }  
}
```

```
        "Name": "EU Site"
      }
    }
  }
```

詳細については、Amazon Connect 管理者ガイド」の「[エージェント階層の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUserHierarchyStructure](#)」の「」を参照してください。

describe-user

次のコード例は、describe-user を使用する方法を示しています。

AWS CLI

ユーザーの詳細を表示するには

次のdescribe-user例では、指定された Amazon Connect ユーザーの詳細を表示します。

```
aws connect describe-user \  
  --user-id 0c245dc0-0cf5-4e37-800e-2a7481cc8a60 \  
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

出力:

```
{  
  "User": {  
    "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",  
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-  
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",  
    "Username": "Jane",  
    "IdentityInfo": {  
      "FirstName": "Jane",  
      "LastName": "Doe",  
      "Email": "example.com"  
    },  
    "PhoneConfig": {  
      "PhoneType": "SOFT_PHONE",  
      "AutoAccept": false,  
      "AfterContactWorkTimeLimit": 0,  
    }  
  }  
}
```

```
        "DeskPhoneNumber": ""
    },
    "DirectoryUserId": "8b444cf6-b368-4f29-ba18-07af27405658",
    "SecurityProfileIds": [
        "b6f85a42-1dc5-443b-b621-de0abf70c9cf"
    ],
    "RoutingProfileId": "0be36ee9-2b5f-4ef4-bcf7-87738e5be0e5",
    "Tags": {}
}
}
```

詳細については、「Amazon Connect 管理者ガイド」の「[ユーザーの管理](#)」を参照してください。 Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUser](#)」の「」を参照してください。

get-contact-attributes

次のコード例は、get-contact-attributes を使用する方法を示しています。

AWS CLI

問い合わせの属性を取得するには

次のget-contact-attributes例では、指定された Amazon Connect 連絡先に設定された属性を取得します。

```
aws connect get-contact-attributes \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --initial-contact-id 12345678-1111-2222-800e-a2b3c4d5f6g7
```

出力:

```
{
  "Attributes": {
    "greetingPlayed": "true"
  }
}
```

詳細については、[Amazon Connect 管理者ガイド](#)の「[Amazon Connect 問い合わせ属性の使用](#)」を参照してください。 Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[GetContactAttributes](#)」の「」を参照してください。

list-contact-flows

次の例は、list-contact-flows を使用する方法を説明しています。

AWS CLI

インスタンスの問い合わせフローを一覧表示するには

次のlist-contact-flows例では、指定された Amazon Connect インスタンスの問い合わせフローを一覧表示します。

```
aws connect list-contact-flows \  
--instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "ContactFlowSummaryList": [  
    {  
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/12345678-1111-2222-800e-  
a2b3c4d5f6g7",  
      "Name": "Default queue transfer",  
      "ContactFlowType": "QUEUE_TRANSFER"  
    },  
    {  
      "Id": "87654321-2222-3333-ac99-123456789102",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/87654321-2222-3333-  
ac99-123456789102",  
      "Name": "Default agent hold",  
      "ContactFlowType": "AGENT_HOLD"  
    },  
    {  
      "Id": "abcdefgh-3333-4444-8af3-201123456789",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/contact-flow/  
abcdefgh-3333-4444-8af3-201123456789",  
    }  
  ]  
}
```

```
        "Name": "Default customer hold",
        "ContactFlowType": "CUSTOMER_HOLD"
    },
]
}
```

詳細については、[Amazon Connect 管理者ガイド](#)の「[Amazon Connect 問い合わせフローの作成](#)」を参照してください。Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[ListContactFlows](#)」の「」を参照してください。

list-hours-of-operations

次の例は、list-hours-of-operations を使用する方法を説明しています。

AWS CLI

インスタンスの営業時間を一覧表示するには

次のlist-hours-of-operations例は、指定された Amazon Connect インスタンスのオペレーション時間を示しています。

```
aws connect list-hours-of-operations \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

出力:

```
{
  "HoursOfOperationSummaryList": [
    {
      "Id": "d69f1f84-7457-4924-8fbe-e64875546259",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/operating-hours/d69f1f84-7457-4924-8fbe-e64875546259",
      "Name": "Basic Hours"
    }
  ]
}
```

詳細については、[Amazon Connect 管理者ガイド](#)の「[キューの営業時間の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListHoursOfOperations](#)」の「」を参照してください。

list-phone-numbers

次の例は、list-phone-numbers を使用する方法を説明しています。

AWS CLI

インスタンスの電話番号を一覧表示するには

次のlist-phone-numbers例では、指定された Amazon Connect インスタンスの電話番号を一覧表示します。

```
aws connect list-phone-numbers \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "PhoneNumberSummaryList": [  
    {  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/xyz80zxy-xyz1-80zx-  
zx80-11111EXAMPLE",  
      "PhoneNumber": "+17065551212",  
      "PhoneNumberType": "DID",  
      "PhoneNumberCountryCode": "US"  
    },  
    {  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/phone-number/c0c0c0-xyz1-80zx-  
zx80-22222EXAMPLE",  
      "PhoneNumber": "+18555551212",  
      "PhoneNumberType": "TOLL_FREE",  
      "PhoneNumberCountryCode": "US"  
    }  
  ]  
}
```

詳細については、Amazon Connect 管理者ガイド」の「[コンタクトセンターの電話番号の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPhoneNumbers](#)」の「」を参照してください。

list-queues

次の例は、list-queues を使用する方法を説明しています。

AWS CLI

インスタンス内のキューを一覧表示するには

次のlist-queues例では、指定された Amazon Connect インスタンスのキューを一覧表示します。

```
aws connect list-queues \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "QueueSummaryList": [  
    {  
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/12345678-1111-2222-800e-  
a2b3c4d5f6g7",  
      "QueueType": "AGENT"  
    },  
    {  
      "Id": "87654321-2222-3333-ac99-123456789102",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/87654321-2222-3333-  
ac99-123456789102",  
      "QueueType": "AGENT"  
    },  
    {  
      "Id": "abcdefgh-3333-4444-8af3-201123456789",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/agent/  
abcdefgh-3333-4444-8af3-201123456789",  
    }  
  ]  
}
```



```
    "QueueType": "AGENT"
  },
  {
    "Id": "hgfedcba-4444-5555-a31f-123456789102",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/queue/hgfedcba-4444-5555-a31f-123456789102",
    "Name": "BasicQueue",
    "QueueType": "STANDARD"
  },
]
}
```

詳細については、「Amazon Connect 管理者ガイド」の「[キューの作成](#)」を参照してください。
Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[ListQueues](#)」の「」を参照してください。

list-routing-profiles

次のコード例は、list-routing-profiles を使用する方法を示しています。

AWS CLI

インスタンスのルーティングプロファイルを一覧表示するには

次のlist-routing-profiles例では、指定された Amazon Connect インスタンスのルーティングプロファイルを一覧表示します。

```
aws connect list-routing-profiles \  
--instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "RoutingProfileSummaryList": [  
    {  
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/routing-profile/12345678-1111-2222-800e-  
a2b3c4d5f6g7",  
      "Name": "Basic Routing Profile"    }  
  ]  
}
```

```
    },  
  ]  
}
```

詳細については、Amazon Connect 管理者ガイド」の「[ルーティングプロファイルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRoutingProfiles](#)」の「」を参照してください。

list-security-profiles

次の例は、list-security-profiles を使用する方法を説明しています。

AWS CLI

インスタンスのセキュリティプロファイルを一覧表示するには

次のlist-security-profiles例では、指定された Amazon Connect インスタンスのセキュリティプロファイルを一覧表示します。

```
aws connect list-security-profiles \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "SecurityProfileSummaryList": [  
    {  
      "Id": "12345678-1111-2222-800e-a2b3c4d5f6g7",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-  
a2b3c4d5f6g7",  
      "Name": "CallCenterManager"  
    },  
    {  
      "Id": "87654321-2222-3333-ac99-123456789102",  
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/87654321-2222-3333-  
ac99-123456789102",  
      "Name": "QualityAnalyst"  
    },  
  ],  
}
```

```
{
  "Id": "abcdefgh-3333-4444-8af3-201123456789",
  "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/
abcdefgh-3333-4444-8af3-201123456789",
  "Name": "Agent"
},
{
  "Id": "12345678-1111-2222-800e-x2y3c4d5fzzzz",
  "Arn": "arn:aws:connect:us-west-2:123456789012:instance/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/security-profile/12345678-1111-2222-800e-
x2y3c4d5fzzzz",
  "Name": "Admin"
}
]
```

詳細については、Amazon Connect 管理者ガイド」の「[アクセス許可の割り当て: セキュリティプロファイル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSecurityProfiles](#)」の「」を参照してください。

list-user-hierarchy-groups

次のコード例は、list-user-hierarchy-groups を使用する方法を示しています。

AWS CLI

インスタンス内のユーザー階層グループを一覧表示するには

次のlist-user-hierarchy-groups例では、指定された Amazon Connect インスタンスのユーザー階層グループを一覧表示します。

```
aws connect list-user-hierarchy-groups \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

出力:

```
{
  "UserHierarchyGroupSummaryList": [
    {
```

```
    "Id": "0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent-group/0e2f6d1d-b3ca-494b-8dbc-ba81d9f8182a",
    "Name": "Example Corporation"
  },
]
}
```

詳細については、Amazon Connect 管理者ガイド」の「[エージェント階層の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUserHierarchyGroups](#)」の「」を参照してください。

list-users

次の例は、list-users を使用する方法を説明しています。

AWS CLI

インスタンス内のユーザー階層グループを一覧表示するには

次のlist-users例では、指定された Amazon Connect インスタンスのユーザーを一覧表示します。

```
aws connect list-users \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

出力:

```
{
  "UserSummaryList": [
    {
      "Id": "0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/0c245dc0-0cf5-4e37-800e-2a7481cc8a60",
      "Username": "Jane"
    },
    {
      "Id": "46f0c67c-3fc7-4806-ac99-403798788c14",
      "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/46f0c67c-3fc7-4806-ac99-403798788c14",
```

```
    "Username": "Paulo"
  },
  {
    "Id": "55a83578-95e1-4710-8af3-2b7afe310e48",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/55a83578-95e1-4710-8af3-2b7afe310e48",
    "Username": "JohnD"
  },
  {
    "Id": "703e27b5-c9f0-4f1f-a239-64ccbb160125",
    "Arn": "arn:aws:connect:us-west-2:123456789012:instance/40c83b68-
ea62-414c-97bb-d018e39e158e/agent/703e27b5-c9f0-4f1f-a239-64ccbb160125",
    "Username": "JohnS"
  }
]
}
```

詳細については、「Amazon Connect 管理者ガイド」の「[ユーザーの追加](#)」を参照してください。Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUsers](#)」の「」を参照してください。

update-contact-attributes

次の例は、update-contact-attributes を使用する方法を説明しています。

AWS CLI

連絡先の属性を更新するには

次のupdate-contact-attributes例では、指定された Amazon Connect ユーザーの greetingPlayed 属性を更新します。

```
aws connect update-contact-attributes \  
  --initial-contact-id 11111111-2222-3333-4444-12345678910 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --attributes greetingPlayed=false
```

このコマンドでは何も出力されません。

詳細については、[Amazon Connect 管理者ガイド](#)の「[Amazon Connect 問い合わせ属性の使用](#)」を参照してください。Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateContactAttributes](#)」の「」を参照してください。

update-user-hierarchy

次のコード例は、update-user-hierarchy を使用する方法を示しています。

AWS CLI

ユーザーの階層を更新するには

次のupdate-user-hierarchy例では、指定された Amazon Connect ユーザーのエージェント階層を更新します。

```
aws connect update-user-hierarchy \  
  --hierarchy-group-id 12345678-a1b2-c3d4-e5f6-123456789abc \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、「Amazon Connect 管理者ガイド」の「[エージェント設定の設定](#)」を参照してください。 Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserHierarchy](#)」の「」を参照してください。

update-user-identity-info

次のコード例は、update-user-identity-info を使用する方法を示しています。

AWS CLI

ユーザーの ID 情報を更新するには

次のupdate-user-identity-info例では、指定された Amazon Connect ユーザーの ID 情報を更新します。

```
aws connect update-user-identity-info \  
  --identity-info FirstName=Mary,LastName=Major,Email=marym@example.com \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、Amazon Connect 管理者ガイド」の「[エージェント設定の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserIdentityInfo](#)」の「」を参照してください。

update-user-phone-config

次のコード例は、update-user-phone-config を使用する方法を示しています。

AWS CLI

ユーザーの電話設定を更新するには

次のupdate-user-phone-config例では、指定されたユーザーの電話設定を更新します。

```
aws connect update-user-phone-config \  
  --phone-  
config PhoneType=SOFT_PHONE,AutoAccept=false,AfterContactWorkTimeLimit=60,DeskPhoneNumber=  
+18005551212 \  
  --user-id 12345678-4444-3333-2222-111122223333 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、Amazon Connect 管理者ガイド」の「[エージェント設定の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserPhoneConfig](#)」の「」を参照してください。

update-user-routing-profile

次のコード例は、update-user-routing-profile を使用する方法を示しています。

AWS CLI

ユーザーのルーティングプロファイルを更新するには

次のupdate-user-routing-profile例では、指定された Amazon Connect ユーザーのルーティングプロファイルを更新します。

```
aws connect update-user-routing-profile \  
  --routing-profile-id 12345678-1111-3333-2222-4444EXAMPLE \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、「Amazon Connect 管理者ガイド」の[「エージェント設定の設定」](#)を参照してください。Amazon Connect

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserRoutingProfile](#)」の「」を参照してください。

update-user-security-profiles

次のコード例は、update-user-security-profiles を使用する方法を示しています。

AWS CLI

ユーザーのセキュリティプロファイルを更新するには

次のupdate-user-security-profiles例では、指定された Amazon Connect ユーザーのセキュリティプロファイルを更新します。

```
aws connect update-user-security-profiles \  
  --security-profile-ids 12345678-1234-1234-1234-1234567892111 \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、「Amazon Connect 管理者ガイド」の[「アクセス許可の割り当て: セキュリティプロファイル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUserSecurityProfiles](#)」の「」を参照してください。

AWS Cost and Usage Report を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Cost and Usage Report。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-report-definition

次のコード例は、delete-report-definition を使用する方法を示しています。

AWS CLI

AWS コストと使用状況レポートを削除するには

この例では、AWS コストと使用状況レポートを削除します。

コマンド:

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteReportDefinition](#)」の「」を参照してください。

describe-report-definitions

次のコード例は、describe-report-definitions を使用する方法を示しています。

AWS CLI

AWS コストと使用状況レポートのリストを取得するには

この例では、アカウントが所有する AWS コストと使用状況レポートのリストについて説明します。

コマンド:

```
aws cur --region us-east-1 describe-report-definitions --max-items 5
```

出力:

```
{
  "ReportDefinitions": [
    {
      "ReportName": "ExampleReport",
      "Compression": "ZIP",
      "S3Region": "us-east-1",
      "Format": "textORcsv",
      "S3Prefix": "exampleprefix",
      "S3Bucket": "example-s3-bucket",
      "TimeUnit": "DAILY",
      "AdditionalArtifacts": [
        "REDSHIFT",
        "QUICKSIGHT"
      ],
      "AdditionalSchemaElements": [
        "RESOURCES"
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReportDefinitions](#)」の「」を参照してください。

put-report-definition

次の例は、put-report-definition を使用する方法を説明しています。

AWS CLI

AWS コストと使用状況レポートを作成するには

次のput-report-definition例では、Amazon Redshift または Amazon にアップロードできる毎日の AWS コストと使用状況レポートを作成します QuickSight。

```
aws cur put-report-definition --report-definition file://report-definition.json
```

report-definition.json の内容:

```
{
  "ReportName": "ExampleReport",
  "TimeUnit": "DAILY",
  "Format": "textORcsv",
  "Compression": "ZIP",
  "AdditionalSchemaElements": [
    "RESOURCES"
  ],
  "S3Bucket": "example-s3-bucket",
  "S3Prefix": "exampleprefix",
  "S3Region": "us-east-1",
  "AdditionalArtifacts": [
    "REDSHIFT",
    "QUICKSIGHT"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutReportDefinition](#)」の「」を参照してください。

を使用した Cost Explorer Service の例 AWS CLI

次のコード例は、Cost Explorer Service AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

get-cost-and-usage

次のコード例は、get-cost-and-usage を使用する方法を示しています。

AWS CLI

2017 年 9 月のアカウントの S3 使用量を取得するには

次のget-cost-and-usage例では、2017 年 9 月のアカウントの S3 使用量を取得します。

```
aws ce get-cost-and-usage \  
  --time-period Start=2017-09-01,End=2017-10-01 \  
  --granularity MONTHLY \  
  --metrics "BlendedCost" "UnblendedCost" "UsageQuantity" \  
  --group-by Type=DIMENSION,Key=SERVICE Type=TAG,Key=Environment \  
  --filter file://filters.json
```

filters.json の内容:

```
{  
  "Dimensions": {  
    "Key": "SERVICE",  
    "Values": [  
      "Amazon Simple Storage Service"  
    ]  
  }  
}
```

出力:

```
{  
  "GroupDefinitions": [  
    {  
      "Type": "DIMENSION",  
      "Key": "SERVICE"  
    },  
    {  
      "Type": "TAG",  
      "Key": "Environment"  
    }  
  ],  
}
```

```
"ResultsByTime": [
  {
    "Estimated": false,
    "TimePeriod": {
      "Start": "2017-09-01",
      "End": "2017-10-01"
    },
    "Total": {},
    "Groups": [
      {
        "Keys": [
          "Amazon Simple Storage Service",
          "Environment$"
        ],
        "Metrics": {
          "BlendedCost": {
            "Amount": "40.3527508453",
            "Unit": "USD"
          },
          "UnblendedCost": {
            "Amount": "40.3543773134",
            "Unit": "USD"
          },
          "UsageQuantity": {
            "Amount": "9312771.098461578",
            "Unit": "N/A"
          }
        }
      }
    ],
    {
      "Keys": [
        "Amazon Simple Storage Service",
        "Environment$Dev"
      ],
      "Metrics": {
        "BlendedCost": {
          "Amount": "0.2682364644",
          "Unit": "USD"
        },
        "UnblendedCost": {
          "Amount": "0.2682364644",
          "Unit": "USD"
        },
        "UsageQuantity": {
```

```
    "Amount": "22403.4395271182",  
    "Unit": "N/A"  
  }  
}  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCostAndUsage](#)」の「」を参照してください。

get-dimension-values

次のコード例は、get-dimension-values を使用する方法を示しています。

AWS CLI

値「ElasticSERVICE」でディメンションのタグを取得するには

この例ではSERVICE、2017年1月1日から2017年5月18日の間、ディメンションのタグを「Elastic」の値で取得します。

コマンド:

```
aws ce get-dimension-values --search-string Elastic --time-  
period Start=2017-01-01,End=2017-05-18 --dimension SERVICE
```

出力:

```
{  
  "TotalSize": 6,  
  "DimensionValues": [  
    {  
      "Attributes": {},  
      "Value": "Amazon ElastiCache"  
    },  
    {  
      "Attributes": {},  
      "Value": "EC2 - Other"  
    }  
  ]  
}
```

```
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Compute Cloud - Compute"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Load Balancing"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic MapReduce"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elasticsearch Service"
    }
  ],
  "ReturnSize": 6
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDimensionValues](#)」の「」を参照してください。

get-reservation-coverage

次のコード例は、get-reservation-coverage を使用する方法を示しています。

AWS CLI

us-east-1 リージョンの EC2 t2.nano インスタンスの予約カバレッジを取得するには

この例では、2017 年 7 月～9 月の us-east-1 リージョンの EC2 t2.nano インスタンスの予約カバレッジを取得します。

コマンド:

```
aws ce get-reservation-coverage --time-period Start=2017-07-01,End=2017-10-01 --  
group-by Type=Dimension,Key=REGION --filter file://filters.json
```

filters.json:

```
{
  "And": [
    {
      "Dimensions": {
        "Key": "INSTANCE_TYPE",
        "Values": [
          "t2.nano"
        ]
      },
      "Dimensions": {
        "Key": "REGION",
        "Values": [
          "us-east-1"
        ]
      }
    }
  ]
}
```

出力:

```
{
  "TotalSize": 6,
  "DimensionValues": [
    {
      "Attributes": {},
      "Value": "Amazon ElastiCache"
    },
    {
      "Attributes": {},
      "Value": "EC2 - Other"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Compute Cloud - Compute"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic Load Balancing"
    },
    {
      "Attributes": {},
      "Value": "Amazon Elastic MapReduce"
    }
  ]
}
```



```

    },
    {
      "Attributes": {},
      "Value": "Amazon Elasticsearch Service"
    }
  ],
  "ReturnSize": 6
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReservationCoverage](#)」の「」を参照してください。

get-reservation-purchase-recommendation

次の例は、get-reservation-purchase-recommendation を使用する方法を説明しています。

AWS CLI

3年間の期間EC2RIsを持つ部分前払いの予約レコメンデーションを取得するには

次のget-reservation-purchase-recommendation例では、過去 60 日間のEC2使用状況に基づいて、3年間の期間を持つ部分前払いEC2インスタンスのレコメンデーションを取得します。

```

aws ce get-reservation-purchase-recommendation \
  --service "Amazon Redshift" \
  --lookback-period-in-days SIXTY_DAYS \
  --term-in-years THREE_YEARS \
  --payment-option PARTIAL_UPFRONT

```

出力:

```

{
  "Recommendations": [],
  "Metadata": {
    "GenerationTimestamp": "2018-08-08T15:20:57Z",
    "RecommendationId": "00d59dde-a1ad-473f-8ff2-iexample3330b"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReservationPurchaseRecommendation](#)」の「」を参照してください。

get-reservation-utilization

次のコード例は、get-reservation-utilization を使用する方法を示しています。

AWS CLI

アカウントの予約使用率を取得するには

次のget-reservation-utilization例では、2018-08-01アカウントの から 2018-03-01 までのすべての t2.nano インスタンスタイプの RI 使用率を取得します。

```
aws ce get-reservation-utilization \  
  --time-period Start=2018-03-01,End=2018-08-01 \  
  --filter file://filters.json
```

filters.json の内容:

```
{  
  "Dimensions": {  
    "Key": "INSTANCE_TYPE",  
    "Values": [  
      "t2.nano"  
    ]  
  }  
}
```

出力:

```
{  
  "Total": {  
    "TotalAmortizedFee": "0",  
    "UtilizationPercentage": "0",  
    "PurchasedHours": "0",  
    "NetRISavings": "0",  
    "TotalActualHours": "0",  
    "AmortizedRecurringFee": "0",  
    "UnusedHours": "0",  
    "TotalPotentialRISavings": "0",  
    "OnDemandCostOfRIHoursUsed": "0",  
    "AmortizedUpfrontFee": "0"  
  },  
  "UtilizationsByTime": []  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReservationUtilization](#)」の「」を参照してください。

get-tags

次の例は、get-tags を使用する方法を説明しています。

AWS CLI

コスト配分タグのキーと値を取得するには

この例では、「プロジェクト」のキーと「」を含む値を持つすべてのコスト配分タグを取得しますsecretProject。

コマンド:

```
aws ce get-tags --search-string secretProject --time-period Start=2017-01-01,End=2017-05-18 --tag-key Project
```

出力:

```
{
  "ReturnSize": 2,
  "Tags": [
    "secretProject1",
    "secretProject2"
  ],
  "TotalSize": 2
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTags](#)」の「」を参照してください。

を使用した Firehose の例 AWS CLI

次のコード例は、Firehose AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

list-delivery-streams

次のコード例は、list-delivery-streams を使用する方法を示しています。

AWS CLI

使用可能な配信ストリームを一覧表示するには

次のlist-delivery-streams例では、AWS アカウントで使用可能な配信ストリームを一覧表示します。

```
aws firehose list-delivery-streams
```

出力:

```
{
  "DeliveryStreamNames": [
    "my-stream"
  ],
  "HasMoreDeliveryStreams": false
}
```

詳細については、Amazon Kinesis Data Firehose デベロッパーガイドの「[Amazon Kinesis Data Firehose 配信ストリームの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeliveryStreams](#)」の「」を参照してください。

put-record-batch

次の例は、put-record-batch を使用する方法を説明しています。

AWS CLI

複数のレコードをストリームに書き込むには

この put-record-batch の例は、3 つのレコードをストリームに書き込みます。データは Base64 形式でエンコードされます。

```
aws firehose put-record-batch \  
  --delivery-stream-name my-stream \  
  --records file://records.json
```

myfile.json の内容:

```
[  
  {"Data": "Rmlyc3QgdGhpbmc="},  
  {"Data": "U2Vjb25kIHRoaW5n"},  
  {"Data": "VGhpcmQgdGhpbmc="}  
]
```

出力:

```
{  
  "FailedPutCount": 0,  
  "Encrypted": false,  
  "RequestResponses": [  
    {  
      "RecordId": "9D20J6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhY0aWKPSmZy/  
CGlRVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn  
+3boDnURDEaD42V7Giixp0yLJkYZcae1i7HzlCEoy9LJhMr8EjDSi40m/9Vc2uhwwuAtGE0XKpxJ2WD7ZRWtAnY1KAnv  
    },  
    {  
      "RecordId": "jFirejqxCLlK5xjH/UNm1MVcjktEN76I7916X9PaZ  
+PVa0SXDFu1WG0qEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/GsuF37Uhg67GkmR5z9016XKJ  
+/+pDloFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCUkBphR2QVzhP90iHLbzGwy8/  
DfH8sqWEUYASNJKS8GXP5s"  
    },  
    {
```

```

    "RecordId":
      "oy0amQ40o5Y2YV4vxzufdcM00w6n3EPr3tpPJGoYVVKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3ie
DTBt3qBlmTj7Xq8SKVb01S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/s0kQLe+qh0sYKXW1"
    }
  ]
}

```

詳細については、「Amazon Kinesis Data Firehose デイベロッパーガイド」の「[Amazon Kinesis Data Firehose 配信ストリームへのデータの送信](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRecordBatch](#)」の「」を参照してください。

put-record

次の例は、put-record を使用する方法を説明しています。

AWS CLI

ストリームにレコードを書き込むには

次のput-record例では、ストリームにデータを書き込みます。データは Base64 形式でエンコードされます。

```

aws firehose put-record \
  --delivery-stream-name my-stream \
  --record '{"Data": "SGVsbG8gd29ybGQ="}'

```

出力:

```

{
  "RecordId": "RjB5K/nnoGFHqwTsZ1Nd/
TTqvjE8V5dsyXZTQn2JXrdpMT0wssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/
j1Xujv31iDhnPdaLw4B0yM9Amv7PcCuB2079RuM0NhoakbyUymlwY8yt20G8X2420wu1j1Fafhci4erAt7QhDEvpwuK8
  "Encrypted": false
}

```

詳細については、「Amazon Kinesis Data Firehose デイベロッパーガイド」の「[Amazon Kinesis Data Firehose 配信ストリームへのデータの送信](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRecord](#)」の「」を参照してください。

を使用した Amazon Data Lifecycle Manager の例 AWS CLI

次のコード例は、Amazon Data Lifecycle Manager AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-default-role

次のコード例は、create-default-role を使用する方法を示しています。

AWS CLI

Amazon に必要なIAMロールを作成するには DLM

次のdlm create-default-role例では、スナップショットを管理するための AWS DataLifecycleManagerDefaultRole デフォルトのロールを作成します。

```
aws dlm create-default-role \  
  --resource-type snapshot
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Elastic Compute Cloud ユーザーガイド」](#)の「[Amazon Data Lifecycle Manager のデフォルトのサービスロール](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateDefaultRole](#)」の「」を参照してください。

create-lifecycle-policy

次のコード例は、create-lifecycle-policy を使用する方法を示しています。

AWS CLI

ライフサイクルポリシーを作成するには

次のcreate-lifecycle-policy例では、指定した時間にボリュームのスナップショットを毎日作成するライフサイクルポリシーを作成します。指定されたタグはスナップショットに追加され、タグはボリュームからコピーされ、スナップショットに追加されます。新しいスナップショットの作成が指定された最大数を超える場合、最も古いスナップショットは削除されます。

```
aws dlm create-lifecycle-policy \  
  --description "My first policy" \  
  --state ENABLED \  
  --execution-role-arn arn:aws:iam::12345678910:role/  
AWSDataLifecycleManagerDefaultRole \  
  --policy-details file://policyDetails.json
```

policyDetails.json の内容:

```
{  
  "ResourceTypes": [  
    "VOLUME"  
  ],  
  "TargetTags": [  
    {  
      "Key": "costCenter",  
      "Value": "115"  
    }  
  ],  
  "Schedules": [  
    {  
      "Name": "DailySnapshots",  
      "CopyTags": true,  
      "TagsToAdd": [  
        {  
          "Key": "type",  
          "Value": "myDailySnapshot"  
        }  
      ],  
      "CreateRule": {
```



```
        "Interval": 24,  
        "IntervalUnit": "HOURS",  
        "Times": [  
            "03:00"  
        ]  
    },  
    "RetainRule": {  
        "Count": 5  
    }  
}  
]  
}
```

出力:

```
{  
  "PolicyId": "policy-0123456789abcdef0"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLifecyclePolicy](#)」の「」を参照してください。

delete-lifecycle-policy

次のコード例は、delete-lifecycle-policy を使用する方法を示しています。

AWS CLI

ライフサイクルポリシーを削除するには

次の例では、指定されたライフサイクルポリシーを削除します。

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLifecyclePolicy](#)」の「」を参照してください。

get-lifecycle-policies

次のコード例は、get-lifecycle-policies を使用する方法を示しています。

AWS CLI

ライフサイクルポリシーの概要を取得するには

次のget-lifecycle-policies例では、すべてのライフサイクルポリシーを一覧表示します。

```
aws dlm get-lifecycle-policies
```

出力:

```
{
  "Policies": [
    {
      "PolicyId": "policy-0123456789abcdef0",
      "Description": "My first policy",
      "State": "ENABLED"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLifecyclePolicies](#)」の「」を参照してください。

get-lifecycle-policy

次のコード例は、get-lifecycle-policy を使用する方法を示しています。

AWS CLI

ライフサイクルポリシーを記述するには

次のget-lifecycle-policy例では、指定されたライフサイクルポリシーの詳細を表示します。

```
aws dlm get-lifecycle-policy \
  --policy-id policy-0123456789abcdef0
```

出力:

```
{
  "Policy": {
    "PolicyId": "policy-0123456789abcdef0",
```

```
    "Description": "My policy",
    "State": "ENABLED",
    "ExecutionRoleArn": "arn:aws:iam::123456789012:role/
AWSDataLifecycleManagerDefaultRole",
    "DateCreated": "2019-08-08T17:45:42Z",
    "DateModified": "2019-08-08T17:45:42Z",
    "PolicyDetails": {
      "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
      "ResourceTypes": [
        "VOLUME"
      ],
      "TargetTags": [
        {
          "Key": "costCenter",
          "Value": "115"
        }
      ],
      "Schedules": [
        {
          "Name": "DailySnapshots",
          "CopyTags": true,
          "TagsToAdd": [
            {
              "Key": "type",
              "Value": "myDailySnapshot"
            }
          ],
          "CreateRule": {
            "Interval": 24,
            "IntervalUnit": "HOURS",
            "Times": [
              "03:00"
            ]
          },
          "RetainRule": {
            "Count": 5
          }
        }
      ]
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLifecyclePolicy](#)」の「」を参照してください。

update-lifecycle-policy

次の例は、update-lifecycle-policy を使用する方法を説明しています。

AWS CLI

例 1: ライフサイクルポリシーを有効にするには

次のupdate-lifecycle-policy例では、指定されたライフサイクルポリシーを有効にします。

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --state ENABLED
```

例 2: ライフサイクルポリシーを無効にするには

次のupdate-lifecycle-policy例では、指定されたライフサイクルポリシーを無効にします。

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --state DISABLED
```

例 3: ライフサイクルポリシーの詳細を更新するには

次のupdate-lifecycle-policy例では、指定されたライフサイクルポリシーのターゲットタグを更新します。

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --policy-details file://policyDetails.json
```

policyDetails.json の内容。このファイルで参照されていないその他の詳細は、コマンドによって変更されません。

```
{
```

```
"TargetTags": [  
  {  
    "Key": "costCenter",  
    "Value": "120"  
  },  
  {  
    "Key": "project",  
    "Value": "lima"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLifecyclePolicy](#)」の「」を参照してください。

AWS Data Pipeline を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Data Pipeline。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

activate-pipeline

次のコード例は、activate-pipeline を使用する方法を示しています。

AWS CLI

パイプラインをアクティブ化するには

この例では、指定されたパイプラインをアクティブ化します。

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

特定の日時でパイプラインをアクティブ化するには、次のコマンドを使用します。

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --start-timestamp 2015-04-07T00:00:00Z
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ActivatePipeline](#)」の「」を参照してください。

add-tags

次のコード例は、add-tags を使用する方法を示しています。

AWS CLI

パイプラインにタグを追加するには

この例では、指定されたタグを指定されたパイプラインに追加します。

```
aws datapipeline add-tags --pipeline-id df-00627471S0VYZEXAMPLE --tags key=environment,value=production key=owner,value=sales
```

タグを表示するには、describe-pipelines コマンドを使用します。例えば、サンプルコマンドで追加されたタグは、describe-pipelines の出力に次のように表示されます。

```
{
  ...
  "tags": [
    {
      "value": "production",
      "key": "environment"
    },
    {
      "value": "sales",
      "key": "owner"
    }
  ]
  ...
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTags](#)」の「」を参照してください。

create-pipeline

次のコード例は、create-pipeline を使用する方法を示しています。

AWS CLI

パイプラインを作成するには

この例では、パイプラインを作成します。

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

出力例を次に示します。

```
{
  "pipelineId": "df-00627471S0VYZEXAMPLE"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePipeline](#)」の「」を参照してください。

deactivate-pipeline

次のコード例は、deactivate-pipeline を使用する方法を示しています。

AWS CLI

パイプラインを非アクティブ化するには

この例では、指定されたパイプラインを非アクティブ化します。

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

実行中のすべてのアクティビティが完了した後にのみパイプラインを非アクティブ化するには、次のコマンドを使用します。

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --no-cancel-active
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeactivatePipeline](#)」の「」を参照してください。

delete-pipeline

次のコード例は、delete-pipeline を使用する方法を示しています。

AWS CLI

パイプラインを削除するには

この例では、指定されたパイプラインを削除します。

```
aws datapipeline delete-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePipeline](#)」の「」を参照してください。

describe-pipelines

次の例は、describe-pipelines を使用する方法を説明しています。

AWS CLI

パイプラインを記述するには

この例では、指定されたパイプラインについて説明します。

```
aws datapipeline describe-pipelines --pipeline-ids df-00627471S0VYZEXAMPLE
```

出力例を次に示します。

```
{
  "pipelineDescriptionList": [
    {
      "fields": [
        {
          "stringValue": "PENDING",
```



```
        "key": "@pipelineState"
      },
      {
        "stringValue": "my-pipeline",
        "key": "name"
      },
      {
        "stringValue": "2015-04-07T16:05:58",
        "key": "@creationTime"
      },
      {
        "stringValue": "df-00627471S0VYZEXAMPLE",
        "key": "@id"
      },
      {
        "stringValue": "123456789012",
        "key": "pipelineCreator"
      },
      {
        "stringValue": "PIPELINE",
        "key": "@sphere"
      },
      {
        "stringValue": "123456789012",
        "key": "@userId"
      },
      {
        "stringValue": "123456789012",
        "key": "@accountId"
      },
      {
        "stringValue": "my-pipeline-token",
        "key": "uniqueId"
      }
    ],
    "pipelineId": "df-00627471S0VYZEXAMPLE",
    "name": "my-pipeline",
    "tags": []
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePipelines](#)」の「」を参照してください。

get-pipeline-definition

次のコード例は、get-pipeline-definition を使用する方法を示しています。

AWS CLI

パイプライン定義を取得するには

この例では、指定されたパイプラインのパイプライン定義を取得します。

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE
```

出力例を次に示します。

```
{
  "parameters": [
    {
      "type": "AWS::S3::ObjectKey",
      "id": "myS3OutputLoc",
      "description": "S3 output folder"
    },
    {
      "default": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/data",
      "type": "AWS::S3::ObjectKey",
      "id": "myS3InputLoc",
      "description": "S3 input folder"
    },
    {
      "default": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
${OUTPUT1_STAGING_DIR}/output.txt",
      "type": "String",
      "id": "myShellCmd",
      "description": "Shell command to run"
    }
  ],
  "objects": [
    {
      "type": "Ec2Resource",
      "terminateAfter": "20 Minutes",
      "instanceType": "t1.micro",
```

```

    "id": "EC2ResourceObj",
    "name": "EC2ResourceObj"
  },
  {
    "name": "Default",
    "failureAndRerunMode": "CASCADE",
    "resourceRole": "DataPipelineDefaultResourceRole",
    "schedule": {
      "ref": "DefaultSchedule"
    },
    "role": "DataPipelineDefaultRole",
    "scheduleType": "cron",
    "id": "Default"
  },
  {
    "directoryPath": "#{myS3OutputLoc}/#{format(@scheduledStartTime, 'YYYY-MM-dd-HH-mm-ss')}",
    "type": "S3DataNode",
    "id": "S3OutputLocation",
    "name": "S3OutputLocation"
  },
  {
    "directoryPath": "#{myS3InputLoc}",
    "type": "S3DataNode",
    "id": "S3InputLocation",
    "name": "S3InputLocation"
  },
  {
    "startAt": "FIRST_ACTIVATION_DATE_TIME",
    "name": "Every 15 minutes",
    "period": "15 minutes",
    "occurrences": "4",
    "type": "Schedule",
    "id": "DefaultSchedule"
  },
  {
    "name": "ShellCommandActivityObj",
    "command": "#{myShellCmd}",
    "output": {
      "ref": "S3OutputLocation"
    },
    "input": {
      "ref": "S3InputLocation"
    }
  },

```

```
        "stage": "true",
        "type": "ShellCommandActivity",
        "id": "ShellCommandActivityObj",
        "runsOn": {
            "ref": "EC2ResourceObj"
        }
    },
    ],
    "values": {
        "myS3OutputLoc": "s3://my-s3-bucket/",
        "myS3InputLoc": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/
data",
        "myShellCmd": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
${OUTPUT1_STAGING_DIR}/output.txt"
    }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPipelineDefinition](#)」の「」を参照してください。

list-pipelines

次のコード例は、list-pipelines を使用する方法を示しています。

AWS CLI

パイプラインを一覧表示するには

この例では、パイプラインを一覧表示します。

```
aws datapipeline list-pipelines
```

出力例を次に示します。

```
{
  "pipelineIdList": [
    {
      "id": "df-00627471S0VYZEXAMPLE",
      "name": "my-pipeline"
    },
    {
      "id": "df-09028963KNVMREXAMPLE",
```

```

    "name": "ImportDDB"
  },
  {
    "id": "df-0870198233ZYVEXAMPLE",
    "name": "CrossRegionDDB"
  },
  {
    "id": "df-00189603TB4MZEXAMPLE",
    "name": "CopyRedshift"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPipelines](#)」の「」を参照してください。

list-runs

次のコード例は、list-runs を使用する方法を示しています。

AWS CLI

例 1: パイプラインの実行を一覧表示するには

次のlist-runs例では、指定されたパイプラインの実行を一覧表示します。

```
aws datapipeline list-runs --pipeline-id df-00627471S0VYZEXAMPLE
```

出力:

Name	Scheduled Start	Status	ID
	Started	Ended	
1. EC2ResourceObj	2015-04-12T17:33:02	CREATING	
@EC2ResourceObj_2015-04-12T17:33:02		2015-04-12T17:33:10	
2. S3InputLocation	2015-04-12T17:33:02	FINISHED	
@S3InputLocation_2015-04-12T17:33:02		2015-04-12T17:33:09	
2015-04-12T17:33:09			
3. S3OutputLocation	2015-04-12T17:33:02	WAITING_ON_DEPENDENCIES	
@S3OutputLocation_2015-04-12T17:33:02		2015-04-12T17:33:09	

```
4. ShellCommandActivityObj    2015-04-12T17:33:02    WAITING_FOR_RUNNER
   @ShellCommandActivityObj_2015-04-12T17:33:02    2015-04-12T17:33:09
```

例 2: 指定された日付の間のパイプライン実行を一覧表示するには

次のlist-runs例では--start-interval、を使用して、出力に含める日付を指定します。

```
aws datapipeline list-runs --pipeline-id df-01434553B58A2SHZUK05 --start-
interval 2017-10-07T00:00:00,2017-10-08T00:00:00
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRuns](#)」の「」を参照してください。

put-pipeline-definition

次の例は、put-pipeline-definition を使用する方法を説明しています。

AWS CLI

パイプライン定義をアップロードするには

この例では、指定されたパイプライン定義を指定されたパイプラインにアップロードします。

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --
pipeline-definition file://my-pipeline-definition.json
```

出力例を次に示します。

```
{
  "validationErrors": [],
  "errored": false,
  "validationWarnings": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutPipelineDefinition](#)」の「」を参照してください。

remove-tags

次のコード例は、remove-tags を使用する方法を示しています。

AWS CLI

パイプラインからタグを削除するには

この例では、指定されたパイプラインから指定されたタグを削除します。

```
aws datapipeline remove-tags --pipeline-id df-00627471S0VYZEXAMPLE --tag-  
keys environment
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveTags](#)」の「」を参照してください。

DataSync を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています DataSync。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

update-location-azure-blob

次のコード例は、update-location-azure-blob を使用する方法を示しています。

AWS CLI

新しいエージェントを使用して転送場所を更新するには

次のupdate-location-object-storage例では、Microsoft Azure Blob Storage DataSync の場所を新しいエージェントで更新します。

```
aws datasync update-location-azure-blob \  

```

```
--location-arn arn:aws:datsync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
--agent-arns arn:aws:datsync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
--sas-configuration '{ \  
    "Token": "sas-token-for-azure-blob-storage-access" \  
'
```

このコマンドでは何も出力されません。

詳細については、AWS DataSync 「ユーザーガイド」の「[エージェントの置き換え](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLocationAzureBlob](#)」の「」を参照してください。

update-location-hdfs

次のコード例は、update-location-hdfs を使用する方法を示しています。

AWS CLI

新しいエージェントを使用して転送場所を更新するには

次のupdate-location-hdfs例では、DataSync HDFS新しいエージェントを使用してロケーションを更新します。HDFS クラスターが Kerberos 認証を使用している場合のみ、--kerberos-keytab と --kerberos-krb5-conf オプションが必要です。

```
aws datsync update-location-hdfs \  
--location-arn arn:aws:datsync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
--agent-arns arn:aws:datsync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
--kerberos-keytab file://hdfs.keytab \  
--kerberos-krb5-conf file://krb5.conf
```

hdfs.keytab の内容:

```
N/A. The content of this file is encrypted and not human readable.
```

krb5.conf の内容:


```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = true
    ticket_lifetime = 24h
    forwardable = true
    udp_preference_limit = 1000000
    default_tkt_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-sha1
    default_tgs_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-sha1
    permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-sha1

[realms]
    EXAMPLE.COM = {
        kdc = kdc1.example.com
        admin_server = krbadmin.example.com
        default_domain = example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM

[logging]
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kerberos/kadmin.log
    default = FILE:/var/log/krb5libs.log
```

このコマンドでは何も出力されません。

詳細については、AWS DataSync 「ユーザーガイド」の「[エージェントの置き換え](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLocationHdfs](#)」の「」を参照してください。

update-location-nfs

次の例は、update-location-nfs を使用する方法を説明しています。

AWS CLI

新しいエージェントを使用して転送場所を更新するには

次のupdate-location-nfs例では、DataSync NFS新しいエージェントを使用してロケーションを更新します。

```
aws datasync update-location-nfs \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
  --on-prem-config AgentArns=arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0
```

このコマンドでは何も出力されません。

詳細については、AWS DataSync 「ユーザーガイド」の [「エージェントの置き換え」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateLocationNfs](#)」の「」を参照してください。

update-location-object-storage

次の例は、update-location-object-storage を使用する方法を説明しています。

AWS CLI

新しいエージェントを使用して転送場所を更新するには

次のupdate-location-object-storage例では、DataSync オブジェクトストレージの場所を新しいエージェントで更新します。

```
aws datasync update-location-object-storage \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
  --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
  --secret-key secret-key-for-object-storage
```

このコマンドでは何も出力されません。

詳細については、AWS DataSync 「ユーザーガイド」の [「エージェントの置き換え」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLocationObjectStorage](#)」の「」を参照してください。

update-location-smb

次のコード例は、update-location-smb を使用する方法を示しています。

AWS CLI

新しいエージェントを使用して転送場所を更新するには

次のupdate-location-smb例では、DataSync SMB新しいエージェントを使用してロケーションを更新します。

```
aws datasync update-location-smb \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/Loc-  
abcdef01234567890 \  
  --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
  --password smb-file-server-password
```

このコマンドでは何も出力されません。

詳細については、AWS DataSync 「ユーザーガイド」の「[エージェントの置き換え](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLocationSmb](#)」の「」を参照してください。

DAX を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますDAX。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-cluster

次の例は、create-cluster を使用する方法を説明しています。

AWS CLI

DAX クラスターを作成するには

次の create-cluster 例では、指定された設定で DAX クラスターを作成します。

```
aws dax create-cluster \  
  --cluster-name daxcluster \  
  --node-type dax.r4.large \  
  --replication-factor 3 \  
  --iam-role-arn roleARN \  
  --sse-specification Enabled=true
```

出力:

```
{  
  "Cluster": {  
    "ClusterName": "daxcluster",  
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",  
    "TotalNodes": 3,  
    "ActiveNodes": 0,  
    "NodeType": "dax.r4.large",  
    "Status": "creating",  
    "ClusterDiscoveryEndpoint": {  
      "Port": 8111  
    },  
    "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",  
    "SubnetGroup": "default",  
    "SecurityGroups": [  
      {  
        "SecurityGroupIdentifier": "sg-1af6e36e",  
        "Status": "active"  
      }  
    ]  
  }  
}
```

```
    ],
    "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
    "ParameterGroup": {
      "ParameterGroupName": "default.dax1.0",
      "ParameterApplyStatus": "in-sync",
      "NodeIdsToReboot": []
    },
    "SSEDescription": {
      "Status": "ENABLED"
    }
  }
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[ステップ 3: DAX クラスターを作成する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCluster](#)」の「」を参照してください。

create-parameter-group

次のコード例は、create-parameter-group を使用する方法を示しています。

AWS CLI

パラメータグループを作成するには

次の create-parameter-group 「」例では、指定された設定でパラメータグループを作成します。

```
aws dax create-parameter-group \
  --parameter-group-name daxparametergroup \
  --description "A new parameter group"
```

出力:

```
{
  "ParameterGroup": {
    "ParameterGroupName": "daxparametergroup",
    "Description": "A new parameter group"
  }
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateParameterGroup](#)」の「」を参照してください。

create-subnet-group

次のコード例は、create-subnet-group を使用する方法を示しています。

AWS CLI

DAXサブネットグループを作成するには

次のcreate-subnet-group例では、指定された設定でサブネットグループを作成します。

```
aws dax create-subnet-group \  
  --subnet-group-name daxSubnetGroup \  
  --subnet-ids subnet-11111111 subnet-22222222
```

出力:

```
{  
  "SubnetGroup": {  
    "SubnetGroupName": "daxSubnetGroup",  
    "VpcId": "vpc-05a1fa8e00c325226",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-11111111",  
        "SubnetAvailabilityZone": "us-west-2b"  
      },  
      {  
        "SubnetIdentifier": "subnet-22222222",  
        "SubnetAvailabilityZone": "us-west-2c"  
      }  
    ]  
  }  
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[「ステップ 2: サブネットグループを作成する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubnetGroup](#)」の「」を参照してください。

decrease-replication-factor

次の例は、`decrease-replication-factor` を使用する方法を説明しています。

AWS CLI

クラスターから 1 つ以上のノードを削除するには

次の`decrease-replication-factor`例では、指定されたDAXクラスター内のノードの数を 1 つに減らします。

```
aws dax decrease-replication-factor \  
  --cluster-name daxcluster \  
  --new-replication-factor 1
```

出力:

```
{  
  "Cluster": {  
    "ClusterName": "daxcluster",  
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",  
    "TotalNodes": 3,  
    "ActiveNodes": 3,  
    "NodeType": "dax.r4.large",  
    "Status": "modifying",  
    "ClusterDiscoveryEndpoint": {  
      "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",  
      "Port": 8111  
    },  
    "Nodes": [  
      {  
        "NodeId": "daxcluster-a",  
        "Endpoint": {  
          "Address": "daxcluster-  
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",  
          "Port": 8111  
        },  
        "NodeCreateTime": 1576625059.509,  
        "AvailabilityZone": "us-west-2c",
```

```
    "NodeStatus": "available",
    "ParameterGroupStatus": "in-sync"
  },
  {
    "NodeId": "daxcluster-b",
    "Endpoint": {
      "Address": "daxcluster-
b.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
      "Port": 8111
    },
    "NodeCreateTime": 1576625059.509,
    "AvailabilityZone": "us-west-2a",
    "NodeStatus": "available",
    "ParameterGroupStatus": "in-sync"
  },
  {
    "NodeId": "daxcluster-c",
    "Endpoint": {
      "Address": "daxcluster-
c.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
      "Port": 8111
    },
    "NodeCreateTime": 1576625059.509,
    "AvailabilityZone": "us-west-2b",
    "NodeStatus": "available",
    "ParameterGroupStatus": "in-sync"
  }
],
"PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
"SubnetGroup": "default",
"SecurityGroups": [
  {
    "SecurityGroupIdentifier": "sg-1af6e36e",
    "Status": "active"
  }
],
"IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
"ParameterGroup": {
  "ParameterGroupName": "default.dax1.0",
  "ParameterApplyStatus": "in-sync",
  "NodeIdsToReboot": []
},
"SSEDescription": {
```



```
        "Status": "ENABLED"
      }
    }
  }
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DecreaseReplicationFactor](#)」の「」を参照してください。

delete-cluster

次の例は、delete-cluster を使用する方法を説明しています。

AWS CLI

DAXクラスターを削除するには

次のdelete-cluster例では、指定されたDAXクラスターを削除します。

```
aws dax delete-cluster \
  --cluster-name daxcluster
```

出力:

```
{
  "Cluster": {
    "ClusterName": "daxcluster",
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
    "TotalNodes": 3,
    "ActiveNodes": 0,
    "NodeType": "dax.r4.large",
    "Status": "deleting",
    "ClusterDiscoveryEndpoint": {
      "Address": "dd.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
      "Port": 8111
    },
    "PreferredMaintenanceWindow": "fri:06:00-fri:07:00",
    "SubnetGroup": "default",
    "SecurityGroups": [
      {
```

```
        "SecurityGroupIdentifier": "sg-1af6e36e",
        "Status": "active"
    }
],
"IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
"ParameterGroup": {
    "ParameterGroupName": "default.dax1.0",
    "ParameterApplyStatus": "in-sync",
    "NodeIdsToReboot": []
},
"SSEDescription": {
    "Status": "ENABLED"
}
}
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteCluster](#)」の「」を参照してください。

delete-parameter-group

次のコード例は、delete-parameter-group を使用する方法を示しています。

AWS CLI

パラメータグループを削除するには

次のdelete-parameter-group例では、指定されたDAXパラメータグループを削除します。

```
aws dax delete-parameter-group \
  --parameter-group-name daxparametergroup
```

出力:

```
{
  "DeletionMessage": "Parameter group daxparametergroup has been deleted."
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスタの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteParameterGroup](#)」の「」を参照してください。

delete-subnet-group

次の例は、delete-subnet-group を使用する方法を説明しています。

AWS CLI

サブネットグループを削除するには

次のdelete-subnet-group例では、指定されたDAXサブネットグループを削除します。

```
aws dax delete-subnet-group \  
  --subnet-group-name daxSubnetGroup
```

出力:

```
{  
  "DeletionMessage": "Subnet group daxSubnetGroup has been deleted."  
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスタの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteSubnetGroup](#)」の「」を参照してください。

describe-clusters

次のコード例は、describe-clusters を使用する方法を示しています。

AWS CLI

プロビジョニングされたすべてのDAXクラスタに関する情報を返すには

次のdescribe-clusters例では、プロビジョニングされたすべてのDAXクラスタの詳細を表示します。

aws dax describe-clusters

出力:

```
{
  "Clusters": [
    {
      "ClusterName": "daxcluster",
      "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",
      "TotalNodes": 1,
      "ActiveNodes": 1,
      "NodeType": "dax.r4.large",
      "Status": "available",
      "ClusterDiscoveryEndpoint": {
        "Address":
"daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",
        "Port": 8111
      },
      "Nodes": [
        {
          "NodeId": "daxcluster-a",
          "Endpoint": {
            "Address": "daxcluster-
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",
            "Port": 8111
          },
          "NodeCreateTime": 1576625059.509,
          "AvailabilityZone": "us-west-2c",
          "NodeStatus": "available",
          "ParameterGroupStatus": "in-sync"
        }
      ],
      "PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
      "SubnetGroup": "default",
      "SecurityGroups": [
        {
          "SecurityGroupIdentifier": "sg-1af6e36e",
          "Status": "active"
        }
      ],
      "IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
      "ParameterGroup": {
```

```

        "ParameterGroupName": "default.dax1.0",
        "ParameterApplyStatus": "in-sync",
        "NodeIdsToReboot": []
    },
    "SSEDescription": {
        "Status": "ENABLED"
    }
}
]
}

```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeClusters](#)」の「」を参照してください。

describe-default-parameters

次のコード例は、describe-default-parameters を使用する方法を示しています。

AWS CLI

のデフォルトのシステムパラメータ情報を返すには DAX

次のdescribe-default-parameters例では、のデフォルトのシステムパラメータ情報を表示しますDAX。

```
aws dax describe-default-parameters
```

出力:

```

{
  "Parameters": [
    {
      "ParameterName": "query-ttl-millis",
      "ParameterType": "DEFAULT",
      "ParameterValue": "300000",
      "NodeTypeSpecificValues": [],
      "Description": "Duration in milliseconds for queries to remain cached",
      "Source": "user",
      "DataType": "integer",

```

```
    "AllowedValues": "0-",
    "IsModifiable": "TRUE",
    "ChangeType": "IMMEDIATE"
  },
  {
    "ParameterName": "record-ttl-millis",
    "ParameterType": "DEFAULT",
    "ParameterValue": "300000",
    "NodeTypeSpecificValues": [],
    "Description": "Duration in milliseconds for records to remain valid in
cache (Default: 0 = infinite)",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": "TRUE",
    "ChangeType": "IMMEDIATE"
  }
]
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeDefaultParameters](#)」の「」を参照してください。

describe-events

次の例は、describe-events を使用する方法を説明しています。

AWS CLI

DAXクラスターとパラメータグループに関連するすべてのイベントを返すには

次のdescribe-events例では、DAXクラスターとパラメータグループに関連するイベントの詳細を表示します。

```
aws dax describe-events
```

出力:

```
{
```

```
"Events": [  
  {  
    "SourceName": "daxcluster",  
    "SourceType": "CLUSTER",  
    "Message": "Cluster deleted.",  
    "Date": 1576702736.706  
  },  
  {  
    "SourceName": "daxcluster",  
    "SourceType": "CLUSTER",  
    "Message": "Removed node daxcluster-b.",  
    "Date": 1576702691.738  
  },  
  {  
    "SourceName": "daxcluster",  
    "SourceType": "CLUSTER",  
    "Message": "Removed node daxcluster-a.",  
    "Date": 1576702633.498  
  },  
  {  
    "SourceName": "daxcluster",  
    "SourceType": "CLUSTER",  
    "Message": "Removed node daxcluster-c.",  
    "Date": 1576702631.329  
  },  
  {  
    "SourceName": "daxcluster",  
    "SourceType": "CLUSTER",  
    "Message": "Cluster created.",  
    "Date": 1576626560.057  
  }  
]  
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の [DAX「クラスタの管理」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DescribeEvents](#)」の「」を参照してください。

describe-parameter-groups

次のコード例は、describe-parameter-groups を使用する方法を示しています。

AWS CLI

で定義されているパラメータグループを記述するには DAX

次のdescribe-parameter-groups例では、で定義されているパラメータグループの詳細を取得しますDAX。

```
aws dax describe-parameter-groups
```

出力:

```
{
  "ParameterGroups": [
    {
      "ParameterGroupName": "default.dax1.0",
      "Description": "Default parameter group for dax1.0"
    }
  ]
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeParameterGroups](#)」の「」を参照してください。

describe-parameters

次のコード例は、describe-parameters を使用する方法を示しています。

AWS CLI

パラメータグループで定義されたDAXパラメータを記述するには

次のdescribe-parameters例では、指定されたパラメータグループで定義されているDAXパラメータの詳細を取得します。

```
aws dax describe-parameters \
  --parameter-group-name default.dax1.0
```

出力:


```
{
  "Parameters": [
    {
      "ParameterName": "query-ttl-millis",
      "ParameterType": "DEFAULT",
      "ParameterValue": "300000",
      "NodeTypeSpecificValues": [],
      "Description": "Duration in milliseconds for queries to remain cached",
      "Source": "user",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": "TRUE",
      "ChangeType": "IMMEDIATE"
    },
    {
      "ParameterName": "record-ttl-millis",
      "ParameterType": "DEFAULT",
      "ParameterValue": "300000",
      "NodeTypeSpecificValues": [],
      "Description": "Duration in milliseconds for records to remain valid in
cache (Default: 0 = infinite)",
      "Source": "user",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": "TRUE",
      "ChangeType": "IMMEDIATE"
    }
  ]
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeParameters](#)」の「」を参照してください。

describe-subnet-groups

次のコード例は、describe-subnet-groups を使用する方法を示しています。

AWS CLI

で定義されているサブネットグループを記述するには DAX

次のdescribe-subnet-groups例では、で定義されているサブネットグループの詳細を取得しますDAX。

```
aws dax describe-subnet-groups
```

出力:

```
{
  "SubnetGroups": [
    {
      "SubnetGroupName": "default",
      "Description": "Default CacheSubnetGroup",
      "VpcId": "vpc-ee70a196",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-874953af",
          "SubnetAvailabilityZone": "us-west-2d"
        },
        {
          "SubnetIdentifier": "subnet-bd3d1fc4",
          "SubnetAvailabilityZone": "us-west-2a"
        },
        {
          "SubnetIdentifier": "subnet-72c2ff28",
          "SubnetAvailabilityZone": "us-west-2c"
        },
        {
          "SubnetIdentifier": "subnet-09e6aa42",
          "SubnetAvailabilityZone": "us-west-2b"
        }
      ]
    }
  ]
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeSubnetGroups](#)」の「」を参照してください。

increase-replication-factor

次のコード例は、`increase-replication-factor` を使用する方法を示しています。

AWS CLI

DAX クラスターのレプリケーション係数を増やすには

次の `increase-replication-factor` 例では、指定された DAX クラスターのレプリケーション係数を 3 に引き上げます。

```
aws dax increase-replication-factor \  
  --cluster-name daxcluster \  
  --new-replication-factor 3
```

出力:

```
{  
  "Cluster": {  
    "ClusterName": "daxcluster",  
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",  
    "TotalNodes": 3,  
    "ActiveNodes": 1,  
    "NodeType": "dax.r4.large",  
    "Status": "modifying",  
    "ClusterDiscoveryEndpoint": {  
      "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",  
      "Port": 8111  
    },  
    "Nodes": [  
      {  
        "NodeId": "daxcluster-a",  
        "Endpoint": {  
          "Address": "daxcluster-  
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",  
          "Port": 8111  
        },  
        "NodeCreateTime": 1576625059.509,  
        "AvailabilityZone": "us-west-2c",  
        "NodeStatus": "available",  
        "ParameterGroupStatus": "in-sync"  
      },  
      {
```

```
        "NodeId": "daxcluster-b",
        "NodeStatus": "creating"
    },
    {
        "NodeId": "daxcluster-c",
        "NodeStatus": "creating"
    }
],
"PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
"SubnetGroup": "default",
"SecurityGroups": [
    {
        "SecurityGroupIdentifier": "sg-1af6e36e",
        "Status": "active"
    }
],
"IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
"ParameterGroup": {
    "ParameterGroupName": "default.dax1.0",
    "ParameterApplyStatus": "in-sync",
    "NodeIdsToReboot": []
},
"SSEDescription": {
    "Status": "ENABLED"
}
}
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の [DAX「クラスターの管理」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [IncreaseReplicationFactor](#)」の「」を参照してください。

list-tags

次の例は、list-tags を使用する方法を説明しています。

AWS CLI

DAXリソースのタグを一覧表示するには

次のlist-tags例では、指定されたDAXクラスターにアタッチされたタグキーと値を一覧表示します。

```
aws dax list-tags \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "ClusterUsage",  
      "Value": "prod"  
    }  
  ]  
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTags](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

DAXリソースにタグを付けるには

次のtag-resource例では、指定されたタグキー名と関連値を指定されたDAXクラスターにアタッチして、クラスターの使用状況を記述します。

```
aws dax tag-resource \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \  
  --tags="Key=ClusterUsage,Value=prod"
```

出力:

```
{
```

```
"Tags": [  
  {  
    "Key": "ClusterUsage",  
    "Value": "prod"  
  }  
]
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、`untag-resource` を使用する方法を示しています。

AWS CLI

DAXリソースからタグを削除するには

次の`untag-resource`例では、指定されたキー名のタグをDAXクラスターから削除します。

```
aws dax untag-resource \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \  
  --tag-keys="ClusterUsage"
```

出力:

```
{  
  "Tags": []  
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の[DAX「クラスターの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

を使用した検出例 AWS CLI

次のコード例は、Detective AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-invitation

次のコード例は、accept-invitation を使用する方法を示しています。

AWS CLI

動作グラフのメンバーアカウントになるための招待を受け入れるには

次のaccept-invitation例では、動作グラフ `arn:aws:detective:us-east-1:111122223333:graph:123412341234` のメンバーアカウントになる招待を受け入れています。

```
aws detective accept-invitation \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

このコマンドでは何も出力されません。

詳細については、「Amazon Detective 管理ガイド」の「[動作グラフの招待への対応](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptInvitation](#)」の「」を参照してください。

create-graph

次のコード例は、create-graph を使用する方法を示しています。

AWS CLI

Amazon Detective を有効にして新しい動作グラフを作成するには

次のcreate-graph例では、コマンドが実行されているリージョンでコマンドを実行する AWS アカウントに対して Detective を有効にします。そのアカウントを管理者アカウントとする新しい動作グラフが作成されます。このコマンドは、値 Finance を Department タグにも割り当てます。

```
aws detective create-graph \  
  --tags '{"Department": "Finance"}'
```

出力:

```
{  
  "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:027c7c4610ea4aacaf0b883093cab899"  
}
```

詳細については、[「Amazon Detective 管理ガイド」の「Amazon Detective の有効化」](#)を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateGraph](#)」の「」を参照してください。

create-members

次のコード例は、create-members を使用する方法を示しています。

AWS CLI

メンバーアカウントを動作グラフに招待するには

次のcreate-members例では、動作グラフ arn:aws:detective:us-east-1:111122223333:graph:123412341234 で 2 つの AWS アカウントをメンバーアカウントとして招待します。アカウントごとに、リクエストは AWS アカウント ID とアカウントのルートユーザーの E メールアドレスを提供します。リクエストには、招待メールに挿入するカスタムメッセージが含まれています。


```
aws detective create-members \  
  --  
  accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email  
  \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --message "This is Paul Santos. I need to add your account to the data we use  
  for security investigation in Amazon Detective. If you have any questions, contact  
  me at psantos@example.com."
```

出力:

```
{  
  "Members": [  
    {  
      "AccountId": "444455556666",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "mmajor@example.com",  
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "Status": "INVITED",  
      "UpdatedTime": 1579826107000  
    },  
    {  
      "AccountId": "123456789012",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "jstiles@example.com",  
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "Status": "VERIFICATION_IN_PROGRESS",  
      "UpdatedTime": 1579826107000  
    }  
  ],  
  "UnprocessedAccounts": [ ]  
}
```

詳細については、「Amazon Detective 管理ガイド」の「メンバーアカウントを動作グラフ <<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>> に招待する」を参照してください。

招待メールを送信せずにメンバーアカウントを招待するには

次のcreate-members例では、動作グラフ `arn:aws:detective:us-east-1:111122223333:graph:123412341234` で 2 つの AWS アカウントをメンバーアカウントとして招待します。アカウントごとに、リクエストは AWS アカウント ID とアカウントのルートユーザーの E メールアドレスを提供します。メンバーアカウントは招待メールを受信しません。

```
aws detective create-members \
  --
accounts AccountId=444455556666,EmailAddress=mmajor@example.com AccountId=123456789012,Email
\
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
  --disable-email-notification
```

出力:

```
{
  "Members": [
    {
      "AccountId": "444455556666",
      "AdministratorId": "111122223333",
      "EmailAddress": "mmajor@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "INVITED",
      "UpdatedTime": 1579826107000
    },
    {
      "AccountId": "123456789012",
      "AdministratorId": "111122223333",
      "EmailAddress": "jstiles@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "VERIFICATION_IN_PROGRESS",
      "UpdatedTime": 1579826107000
    }
  ],
  "UnprocessedAccounts": [ ]
}
```

詳細については、「Amazon Detective 管理ガイド」の「メンバーアカウントを動作グラフ <<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>> に招待する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateMembers](#)」の「」を参照してください。

delete-graph

次のコード例は、delete-graph を使用する方法を示しています。

AWS CLI

Detective を無効にして動作グラフを削除するには

次のdelete-graph例では、Detective を無効にし、指定された動作グラフを削除します。

```
aws detective delete-graph \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Detective 管理ガイド](#)」の「[Amazon Detective の無効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteGraph](#)」の「」を参照してください。

delete-members

次の例は、delete-members を使用する方法を説明しています。

AWS CLI

動作グラフからメンバーアカウントを削除するには

次のdelete-members例では、動作グラフ `arn:aws:detective:us-east-1:111122223333:graph:123412341234` から 2 つのメンバーアカウントを削除します。アカウントを識別するために、リクエストは AWS アカウント を提供します IDs。

```
aws detective delete-members \  
  --account-ids 444455556666 123456789012 \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

```
--graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

出力:

```
{
  "AccountIds": [ "444455556666", "123456789012" ],
  "UnprocessedAccounts": [ ]
}
```

詳細については、「Amazon Detective 管理ガイド」の「動作グラフ <<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-remove-member-accounts.html>> からメンバーアカウントを削除する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteMembers](#)」の「」を参照してください。

disassociate-membership

次の例は、disassociate-membership を使用する方法を説明しています。

AWS CLI

動作グラフからメンバーシップを辞退するには

次の関連付け解除メンバーの例は、動作グラフ arn:aws:detective:us-east-1:111122223333:graph:123412341234 からコマンドを実行する AWS アカウントを削除します。

```
aws detective disassociate-membership \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

詳細については、「Amazon Detective 管理ガイド」の「動作グラフ <<https://docs.aws.amazon.com/detective/latest/adminguide/member-remove-self-from-graph.html>> からアカウントを削除する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateMembership](#)」の「」を参照してください。

get-members

次のコード例は、get-members を使用する方法を示しています。

AWS CLI

選択した動作グラフメンバーアカウントに関する情報を取得するには

次の`get-members`例では、動作グラフ `arn:aws:detective:us-east-1:111122223333:graph:123412341234` の 2 つのメンバーアカウントに関する情報を取得します。2 つのアカウントの場合、リクエストは AWS アカウント を提供します IDs。

```
aws detective get-members \  
  --account-ids 444455556666 123456789012 \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

出力:

```
{  
  "MemberDetails": [  
    {  
      "AccountId": "444455556666",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "mmajor@example.com",  
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "Status": "INVITED",  
      "UpdatedTime": 1579826107000  
    }  
    {  
      "AccountId": "123456789012",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "jstiles@example.com",  
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "Status": "INVITED",  
      "UpdatedTime": 1579826107000  
    }  
  ],  
  "UnprocessedAccounts": [ ]  
}
```

詳細については、「Amazon Detective 管理ガイド」の「動作グラフ <<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html>> でアカウントのリストを表示する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMembers](#)」の「」を参照してください。

list-graphs

次のコード例は、list-graphs を使用する方法を示しています。

AWS CLI

アカウントが管理者である動作グラフのリストを表示するには

次のlist-graphs例では、呼び出し元のアカウントが現在のリージョン内の管理者である動作グラフを取得します。

```
aws detective list-graphs
```

出力:

```
{
  "GraphList": [
    {
      "Arn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "CreatedTime": 1579736111000
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGraphs](#)」の「」を参照してください。

list-invitations

次の例は、list-invitations を使用する方法を説明しています。

AWS CLI

アカウントがメンバーであるか、招待されている動作グラフのリストを表示するには

次のlist-invitations例では、呼び出し元のアカウントが招待された動作グラフを取得します。結果には、開いている招待と受け入れられた招待のみが含まれます。拒否された招待や削除されたメンバーシップは含まれません。

```
aws detective list-invitations
```

出力:

```
{
  "Invitations": [
    {
      "AccountId": "444455556666",
      "AdministratorId": "111122223333",
      "EmailAddress": "mmajor@example.com",
      "GraphArn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "InvitedTime": 1579826107000,
      "MasterId": "111122223333",
      "Status": "INVITED",
      "UpdatedTime": 1579826107000
    }
  ]
}
```

詳細については、「Amazon Detective 管理ガイド」の「動作グラフの招待のリストの表示」 <<https://docs.aws.amazon.com/detective/latest/adminguide/member-view-graph-invitations.html>> を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInvitations](#)」の「」を参照してください。

list-members

次の例は、list-members を使用する方法を説明しています。

AWS CLI

動作グラフにメンバーアカウントを一覧表示するには

次のlist-members例では、動作グラフ の招待された有効なメンバーアカウントを取得しますarn:aws:detective:us-east-1:111122223333:graph:123412341234。結果には、削除されたメンバーアカウントは含まれません。

```
aws detective list-members \  
--graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

出力:

```
{  
  "MemberDetails": [  
    {  
      "AccountId": "444455556666",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "mmajor@example.com",  
      "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "Status": "INVITED",  
      "UpdatedTime": 1579826107000  
    },  
    {  
      "AccountId": "123456789012",  
      "AdministratorId": "111122223333",  
      "EmailAddress": "jstiles@example.com",  
      "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:123412341234",  
      "InvitedTime": 1579826107000,  
      "MasterId": "111122223333",  
      "PercentOfGraphUtilization": 2,  
      "PercentOfGraphUtilizationUpdatedTime": 1586287843,  
      "Status": "ENABLED",  
      "UpdatedTime": 1579973711000,  
      "VolumeUsageInBytes": 200,  
      "VolumeUsageUpdatedTime": 1586287843  
    }  
  ]  
}
```

詳細については、[「Amazon Detective 管理ガイド」の「動作グラフでのアカウントのリストの表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMembers](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

動作グラフに割り当てられたタグを取得するには

次のlist-tags-for-resource例では、指定された動作グラフに割り当てられたタグを返します。

```
aws detective list-tags-for-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

出力:

```
{  
  "Tags": {  
    "Department" : "Finance"  
  }  
}
```

詳細については、「Amazon Detective 管理ガイド」の「[動作グラフのタグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

reject-invitation

次の例は、reject-invitation を使用する方法を説明しています。

AWS CLI

動作グラフのメンバーアカウントへの招待を拒否するには

次のreject-invitation例では、動作グラフ arn:aws:detective:us-east-1:111122223333:graph:123412341234 のメンバーアカウントへの招待を拒否しています。

```
aws detective reject-invitation \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

このコマンドでは何も出力されません。

詳細については、「Amazon Detective 管理ガイド」の「動作グラフの招待 < <https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html> > への応答」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RejectInvitation](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを割り当てるには

次のtag-resource例では、指定された動作グラフに Department タグの値を割り当てます。

```
aws detective tag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tags '{"Department": "Finance"}'
```

このコマンドでは何も出力されません。

詳細については、「Amazon Detective 管理ガイド」の「[動作グラフのタグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグ値を削除するには

次のuntag-resource例では、指定された動作グラフから Department タグを削除します。

```
aws detective untag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tag-key Department
```

```
--tag-keys "Department"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Detective 管理ガイド」の「[動作グラフのタグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

を使用した Device Farm の例 AWS CLI

次のコード例は、Device Farm AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-device-pool

次の例は、create-device-pool を使用する方法を説明しています。

AWS CLI

デバイスプールを作成するには

次のコマンドは、プロジェクトの Android デバイスプールを作成します。

```
aws devicefarm create-device-pool --name pool1 --rules file://  
device-pool-rules.json --project-arn "arn:aws:devicefarm:us-  
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506"
```

プロジェクトは、`create-project`または の出力ARNから取得できます`list-projects`。
ファイルは、デバイスプラットフォームを指定する現在のフォルダ内のJSONドキュメント`device-pool-rules.json`です。

```
[
  {
    "attribute": "PLATFORM",
    "operator": "EQUALS",
    "value": "\"ANDROID\""
  }
]
```

出力:

```
{
  "devicePool": {
    "rules": [
      {
        "operator": "EQUALS",
        "attribute": "PLATFORM",
        "value": "\"ANDROID\""
      }
    ],
    "type": "PRIVATE",
    "name": "pool1",
    "arn": "arn:aws:devicefarm:us-west-2:123456789012:devicepool:070fc3ca-7ec1-4741-9c1f-d3e044efc506/2aa8d2a9-5e73-47ca-b929-659cb34b7dcd"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDevicePool](#)」の「」を参照してください。

create-project

次の例は、`create-project` を使用する方法を説明しています。

AWS CLI

プロジェクトを作成するには

次のコマンドは、 という名前の新しいプロジェクトを作成しますmy-project。

```
aws devicefarm create-project --name my-project
```

出力:

```
{
  "project": {
    "name": "myproject",
    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
    "created": 1503612890.057
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProject](#)」の「」を参照してください。

create-upload

次の例は、create-upload を使用する方法を説明しています。

AWS CLI

アップロードを作成するには

次のコマンドは、Android アプリのアップロードを作成します。

```
aws devicefarm create-upload --project-arn "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506" --name app.apk --
type ANDROID_APP
```

create-project または list-project の出力ARNからプロジェクトを取得できます。

出力:

```
{
  "upload": {
    "status": "INITIALIZED",
    "name": "app.apk",
    "created": 1503614408.769,
```

```
    "url": "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/
arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-
c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-
west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-
Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f",
    "type": "ANDROID_APP",
    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
  }
}
```

出力URLで署名された を使用して、ファイルを Device Farm にアップロードします。

```
curl -T app.apk "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/
arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-
c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-
west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-
Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateUpload](#)」の「」を参照してください。

get-upload

次のコード例は、get-upload を使用する方法を示しています。

AWS CLI

アップロードを表示するには

次のコマンドは、アップロードに関する情報を取得します。

```
aws devicefarm get-upload --arn "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
```

アップロードは、 の出力ARNから取得できますcreate-upload。

出力:

```
{
  "upload": {
    "status": "SUCCEEDED",
    "name": "app.apk",
    "created": 1505262773.186,
    "type": "ANDROID_APP",
    "arn": "arn:aws:devicefarm:us-west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-ae9e-4087-09e6-f4cea3599514",
    "metadata": "{\"device_admin\":false,\"activity_name\": \"com.example.client.LauncherActivity\", \"version_name\": \"1.0.2.94\", \"screens\": [\"small\", \"normal\", \"large\", \"xlarge\"], \"error_type\": null, \"sdk_version\": \"16\", \"package_name\": \"com.example.client\", \"version_code\": \"20994\", \"native_code\": [\"armeabi-v7a\"], \"target_sdk_version\": \"25\"}"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUpload](#)」の「」を参照してください。

list-projects

次の例は、list-projects を使用する方法を説明しています。

AWS CLI

プロジェクトを一覧表示するには

以下は、プロジェクトのリストを取得します。

```
aws devicefarm list-projects
```

出力:

```
{
  "projects": [
    {
      "name": "myproject",
```

```
    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
    "created": 1503612890.057
  },
  {
    "name": "otherproject",
    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:a5f5b752-8098-49d1-86bf-5f7682c1c77e",
    "created": 1505257519.337
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProjects](#)」の「」を参照してください。

AWS Direct Connect を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Direct Connect。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-direct-connect-gateway-association-proposal

次の例は、accept-direct-connect-gateway-association-proposal を使用する方法を説明しています。

AWS CLI

ゲートウェイ関連付け提案を受け入れるには

以下は、指定された提案 `accept-direct-connect-gateway-association-proposal` を受け入れます。

```
aws directconnect accept-direct-connect-gateway-association-proposal \
  --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \
  --proposal-id cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE \
  --associated-gateway-owner-account 111122223333

{
  "directConnectGatewayAssociation": {
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "directConnectGatewayOwnerAccount": "111122223333",
    "associationState": "associating",
    "associatedGateway": {
      "id": "tgw-02f776b1a7EXAMPLE",
      "type": "transitGateway",
      "ownerAccount": "111122223333",
      "region": "us-east-1"
    },
    "associationId": "6441f8bf-5917-4279-ade1-9708bEXAMPLE",
    "allowedPrefixesToDirectConnectGateway": [
      {
        "cidr": "192.168.1.0/30"
      }
    ]
  }
}
```

詳細については、AWS Direct Connect ユーザーガイドの「[トランジットゲートウェイの関連付け提案の承諾または拒否](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AcceptDirectConnectGatewayAssociationProposal](#)」の「」を参照してください。

allocate-connection-on-interconnect

次のコード例は、`allocate-connection-on-interconnect` を使用する方法を示しています。

AWS CLI

相互接続でホスト接続を作成するには

次のallocate-connection-on-interconnectコマンドは、相互接続にホスト接続を作成します。

```
aws directconnect allocate-connection-on-interconnect --bandwidth 500Mbps --
connection-name mydcinterconnect --owner-account 123456789012 --interconnect-
id dxcon-fgktov66 --vlan 101
```

出力:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffzc51m1",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AllocateConnectionOnInterconnect](#)」の「」を参照してください。

allocate-hosted-connection

次の例は、allocate-hosted-connection を使用する方法を説明しています。

AWS CLI

相互接続でホスト接続を作成するには

次のallocate-hosted-connection例では、指定された相互接続にホスト接続を作成します。

```
aws directconnect allocate-hosted-connection \
--bandwidth 500Mbps \
--connection-name mydcinterconnect \
--owner-account 123456789012
-connection-id dxcon-fgktov66
-vlan 101
```

出力:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffzc51m1",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AllocateHostedConnection](#)」の「」を参照してください。

allocate-private-virtual-interface

次のコード例は、allocate-private-virtual-interface を使用する方法を示しています。

AWS CLI

プライベート仮想インターフェイスをプロビジョニングするには

次のallocate-private-virtual-interface コマンドは、別の顧客が所有するプライベート仮想インターフェイスをプロビジョニングします。

```
aws directconnect allocate-private-virtual-interface --connection-id dxcon-ffjrnx17 --owner-account 123456789012 --new-private-virtual-interface-allocation virtualInterfaceName=PrivateVirtualInterface,vlan=1000,asn=65000,authKey=asdf34ex
```

出力:

```
{
  "virtualInterfaceState": "confirming",
  "asn": 65000,
  "vlan": 1000,
  "customerAddress": "192.168.1.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrnx17",
}
```

```

    "virtualInterfaceId": "dxvif-fgy8orxu",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n <logical_connection id=\"dxvif-fgy8orxu\">\n <vlan>1000</
vlan>\n <customer_address>192.168.1.2/30</customer_address>\n
<amazon_address>192.168.1.1/30</amazon_address>\n <bgp_asn>65000</bgp_asn>\n
<bgp_auth_key>asdf34example</bgp_auth_key>\n <amazon_bgp_asn>7224</amazon_bgp_asn>
\n <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "192.168.1.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "PrivateVirtualInterface"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [AllocatePrivateVirtualInterface](#)」の「」を参照してください。

allocate-public-virtual-interface

次のコード例は、allocate-public-virtual-interface を使用する方法を示しています。

AWS CLI

パブリック仮想インターフェイスをプロビジョニングするには

次のallocate-public-virtual-interfaceコマンドは、別の顧客が所有するパブリック仮想インターフェイスをプロビジョニングします。

```

aws directconnect allocate-public-virtual-interface --connection-id dxcon-ffjrnx17 --owner-account 123456789012 --new-public-virtual-interface-allocation virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34example,cidr=203.0.113.4/30]

```

出力:

```

{
  "virtualInterfaceState": "confirming",
  "asn": 65000,
  "vlan": 2000,
  "customerAddress": "203.0.113.2/30",
}

```

```

"ownerAccount": "123456789012",
"connectionId": "dxcon-ffjrkrx17",
"virtualInterfaceId": "dxvif-fg9xo9vp",
"authKey": "asdf34example",
"routeFilterPrefixes": [
  {
    "cidr": "203.0.113.0/30"
  },
  {
    "cidr": "203.0.113.4/30"
  }
],
"location": "TIVIT",
"customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<logical_connection id=\"dxvif-fg9xo9vp\">
  <vlan>2000</vlan>
  <customer_address>203.0.113.2/30</customer_address>
  <amazon_address>203.0.113.1/30</amazon_address>
  <bgp_asn>65000</bgp_asn>
  <bgp_auth_key>asdf34example</bgp_auth_key>
  <amazon_bgp_asn>7224</amazon_bgp_asn>
</logical_connection>
",
"amazonAddress": "203.0.113.1/30",
"virtualInterfaceType": "public",
"virtualInterfaceName": "PublicVirtualInterface"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [AllocatePublicVirtualInterface](#)」の「」を参照してください。

allocate-transit-virtual-interface

次のコード例は、allocate-transit-virtual-interface を使用する方法を示しています。

AWS CLI

指定された AWS アカウントが所有するトランジット仮想インターフェイスをプロビジョニングするには

次のallocate-transit-virtual-interface例では、指定されたアカウントのトランジット仮想インターフェイスをプロビジョニングします。

```

aws directconnect allocate-transit-virtual-interface \
  --connection-id dxlag-fEXAMPLE \
  --owner-account 123456789012 \

```

```
--new-transit-virtual-interface-allocation "virtualInterfaceName=Example Transit
Virtual
Interface,vlan=126,asn=65110,mtu=1500,authKey=0xzxcgA9YoW9h58u8SEXAMPLE,amazonAddress=192.16
```

出力:

```
{
  "virtualInterface": {
    "ownerAccount": "123456789012",
    "virtualInterfaceId": "dxvif-fEXAMPLE",
    "location": "loc1",
    "connectionId": "dxlag-fEXAMPLE",
    "virtualInterfaceType": "transit",
    "virtualInterfaceName": "Example Transit Virtual Interface",
    "vlan": 126,
    "asn": 65110,
    "amazonSideAsn": 7224,
    "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",
    "amazonAddress": "192.168.1.1/30",
    "customerAddress": "192.168.1.2/30",
    "addressFamily": "ipv4",
    "virtualInterfaceState": "confirming",
    "customerRouterConfig": "<?xml version='1.0' encoding=
  \ UTF-8'?'>\n<logical_connection id='dxvif-fEXAMPLE'>\n  <vlan>126</
  vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
  <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65110</bgp_asn>\n
  <bgp_auth_key>0xzxcgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>7224</
  amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
  \n",
    "mtu": 1500,
    "jumboFrameCapable": true,
    "virtualGatewayId": "",
    "directConnectGatewayId": "",
    "routeFilterPrefixes": [],
    "bgpPeers": [
      {
        "bgpPeerId": "dxpeer-fEXAMPLE",
        "asn": 65110,
        "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",
        "addressFamily": "ipv4",
        "amazonAddress": "192.168.1.1/30",
        "customerAddress": "192.168.1.2/30",
        "bgpPeerState": "pending",
```

```
        "bgpStatus": "down",
        "awsDeviceV2": "loc1-26wz6vEXAMPLE"
    }
],
"region": "sa-east-1",
"awsDeviceV2": "loc1-26wz6vEXAMPLE",
"tags": [
    {
        "key": "Tag",
        "value": "Example"
    }
]
}
}
```

詳細については、AWS Direct Connect ユーザーガイドの「[ホストされたトランジット仮想インターフェイスの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AllocateTransitVirtualInterface](#)」の「」を参照してください。

associate-connection-with-lag

次の例は、associate-connection-with-lag を使用する方法を説明しています。

AWS CLI

接続を に関連付けるには LAG

次の例では、指定された接続を指定された に関連付けますLAG。

コマンド:

```
aws directconnect associate-connection-with-lag --lag-id dxlag-fhccu14t --
connection-id dxcon-fg9607vm
```

出力:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
```

```
"connectionState": "requested",
"bandwidth": "1Gbps",
"location": "EqDC2",
"connectionName": "Con2ForLag",
"region": "us-east-1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateConnectionWithLag](#)」の「」を参照してください。

associate-hosted-connection

次の例は、associate-hosted-connection を使用する方法を説明しています。

AWS CLI

ホスト接続を に関連付けるには LAG

次の例では、指定されたホスト接続を指定された に関連付けますLAG。

コマンド:

```
aws directconnect associate-hosted-connection --parent-connection-id dxlag-fhccu14t
--connection-id dxcon-fg9607vm
```

出力:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateHostedConnection](#)」の「」を参照してください。

associate-virtual-interface

次のコード例は、`associate-virtual-interface` を使用する方法を示しています。

AWS CLI

仮想インターフェイスを接続に関連付けるには

次の例では、指定された仮想インターフェイスを指定されたに関連付けますLAG。または、仮想インターフェイスを接続に関連付けるには、の AWS Direct Connect 接続の ID を指定します `--connection-id`。例えば、です `dxcon-ffnikghc`。

コマンド:

```
aws directconnect associate-virtual-interface --connection-id dxlag-ffjhj9lx --virtual-interface-id dxvif-fgputw0j
```

出力:

```
{
  "virtualInterfaceState": "pending",
  "asn": 65000,
  "vlan": 123,
  "customerAddress": "169.254.255.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxlag-ffjhj9lx",
  "addressFamily": "ipv4",
  "virtualGatewayId": "vgw-38e90b51",
  "virtualInterfaceId": "dxvif-fgputw0j",
  "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
  "routeFilterPrefixes": [],
  "location": "CSVA1",
  "bgpPeers": [
    {
      "bgpStatus": "down",
      "customerAddress": "169.254.255.2/30",
      "addressFamily": "ipv4",
      "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
      "bgpPeerState": "deleting",
      "amazonAddress": "169.254.255.1/30",
      "asn": 65000
    }
  ],
  {
```

```

    "bgpStatus": "down",
    "customerAddress": "169.254.255.2/30",
    "addressFamily": "ipv4",
    "authKey": "0x123pK5_VBqv.UQ3kJ4123_",
    "bgpPeerState": "pending",
    "amazonAddress": "169.254.255.1/30",
    "asn": 65000
  }
],
  "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<logical_connection id=\"dxvif-fgputw0j\">
  <vlan>123</vlan>
  <customer_address>169.254.255.2/30</customer_address>
  <amazon_address>169.254.255.1/30</amazon_address>
  <bgp_asn>65000</bgp_asn>
  <bgp_auth_key>0x123pK5_VBqv.UQ3kJ4123_</bgp_auth_key>
  <amazon_bgp_asn>7224</amazon_bgp_asn>
  <connection_type>private</connection_type>
</logical_connection>
",
  "amazonAddress": "169.254.255.1/30",
  "virtualInterfaceType": "private",
  "virtualInterfaceName": "VIF1A"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateVirtualInterface](#)」の「」を参照してください。

confirm-connection

次の例は、confirm-connection を使用する方法を説明しています。

AWS CLI

相互接続でホスト接続の作成を確認するには

次のconfirm-connectionコマンドは、相互接続でホスト接続の作成を確認します。

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

出力:

```
{
  "connectionState": "pending"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfirmConnection](#)」の「」を参照してください。

confirm-private-virtual-interface

次のコード例は、confirm-private-virtual-interface を使用する方法を示しています。

AWS CLI

プライベート仮想インターフェイスの所有権を受け入れるには

次のconfirm-private-virtual-interfaceコマンドは、別の顧客が作成したプライベート仮想インターフェイスの所有権を受け入れます。

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

出力:

```
{
  "virtualInterfaceState": "pending"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfirmPrivateVirtualInterface](#)」の「」を参照してください。

confirm-public-virtual-interface

次の例は、confirm-public-virtual-interface を使用する方法を説明しています。

AWS CLI

パブリック仮想インターフェイスの所有権を受け入れるには

次のconfirm-public-virtual-interfaceコマンドは、別の顧客が作成したパブリック仮想インターフェイスの所有権を受け入れます。

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

出力:

```
{
  "virtualInterfaceState": "verifying"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfirmPublicVirtualInterface](#)」の「」を参照してください。

confirm-transit-virtual-interface

次のコード例は、confirm-transit-virtual-interface を使用する方法を示しています。

AWS CLI

トランジット仮想インターフェイスの所有権を受け入れるには

以下は、別の顧客が作成したトランジット仮想インターフェイスの所有権confirm-transit-virtual-interfaceを受け入れます。

```
aws directconnect confirm-transit-virtual-interface \
  --virtual-interface-id dxvif-fEXAMPLE \
  --direct-connect-gateway-id 4112ccf9-25e9-4111-8237-b6c5dEXAMPLE
```

出力:

```
{
  "virtualInterfaceState": "pending"
}
```

詳細については、AWS Direct Connect ユーザーガイドの「[ホストされた仮想インターフェイスの受け入れ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfirmTransitVirtualInterface](#)」の「」を参照してください。

create-bgp-peer

次の例は、create-bgp-peer を使用する方法を説明しています。

AWS CLI

IPv6BGPピアリングセッションを作成するには

次の例では、プライベート仮想インターフェイスにIPv6BGPピアリングセッションを作成します。ピアIPv6アドレスは Amazon によって自動的に割り当てられます。

コマンド:

```
aws directconnect create-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --new-bgp-peer asn=64600,addressFamily=ipv6
```

出力:

```
{
  "virtualInterface": {
    "virtualInterfaceState": "available",
    "asn": 65000,
    "vlan": 125,
    "customerAddress": "169.254.255.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fguhmq1c",
    "addressFamily": "ipv4",
    "virtualGatewayId": "vgw-f9eb0c90",
    "virtualInterfaceId": "dxvif-fg1vuj3d",
    "authKey": "0xC_ukbCer16EYA0example",
    "routeFilterPrefixes": [],
    "location": "EqDC2",
    "bgpPeers": [
      {
        "bgpStatus": "down",
        "customerAddress": "169.254.255.2/30",
        "addressFamily": "ipv4",
        "authKey": "0xC_ukbCer16EYA0uexample",
        "bgpPeerState": "available",
        "amazonAddress": "169.254.255.1/30",
        "asn": 65000
      },
      {
        "bgpStatus": "down",
        "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
        "addressFamily": "ipv6",
        "authKey": "0xS27kAIU_VHPjjAexample",

```

```

        "bgpPeerState": "pending",
        "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
        "asn": 64600
    }
],
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
  <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</
bgp_asn>\n  <bgp_auth_key>0xC_ukbCerl6EYA0uexample</bgp_auth_key>\n
  <ipv6_customer_address>2001:db8:1100:2f0:0:1:9cb4:4216/125</ipv6_customer_address>
\n  <ipv6_amazon_address>2001:db8:1100:2f0:0:1:9cb4:4211/125</ipv6_amazon_address>\n
  <ipv6_bgp_asn>64600</ipv6_bgp_asn>\n  <ipv6_bgp_auth_key>0xS27kAIU_VHPjjAexample</
ipv6_bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>\n
  <connection_type>private</connection_type>\n</logical_connection>\n",
    "amazonAddress": "169.254.255.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "Test"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateBgpPeer](#)」の「」を参照してください。

create-connection

次のコード例は、create-connection を使用する方法を示しています。

AWS CLI

ネットワークから AWS Direct Connect ロケーションへの接続を作成するには

次のcreate-connectionコマンドは、ネットワークから AWS Direct Connect ロケーションへの接続を作成します。

```
aws directconnect create-connection --location TIVIT --bandwidth 1Gbps --connection-name "Connection to AWS"
```

出力:

```
{
  "ownerAccount": "123456789012",
```

```
"connectionId": "dxcon-fg31dyv6",
"connectionState": "requested",
"bandwidth": "1Gbps",
"location": "TIVIT",
"connectionName": "Connection to AWS",
"region": "sa-east-1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConnection](#)」の「」を参照してください。

create-direct-connect-gateway-association-proposal

次のコード例は、`create-direct-connect-gateway-association-proposal` を使用する方法を示しています。

AWS CLI

指定されたトランジットゲートウェイを指定された Direct Connect ゲートウェイに関連付ける提案を作成するには

次の`create-direct-connect-gateway-association-proposal`例では、指定されたトランジットゲートウェイを指定された Direct Connect ゲートウェイに関連付ける提案を作成します。

```
aws directconnect create-direct-connect-gateway-association-proposal \
  --direct-connect-gateway-id 11460968-4ac1-4fd3-bdb2-00599EXAMPLE \
  --direct-connect-gateway-owner-account 111122223333 \
  --gateway-id tgw-02f776b1a7EXAMPLE \
  --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.1.0/30
```

出力:

```
{
  "directConnectGatewayAssociationProposal": {
    "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "directConnectGatewayOwnerAccount": "111122223333",
    "proposalState": "requested",
    "associatedGateway": {
      "id": "tgw-02f776b1a7EXAMPLE",
      "type": "transitGateway",
    }
  }
}
```

```
    "ownerAccount": "111122223333",
    "region": "us-east-1"
  },
  "requestedAllowedPrefixesToDirectConnectGateway": [
    {
      "cidr": "192.168.1.0/30"
    }
  ]
}
```

詳細については、AWS Direct Connect ユーザーガイドの「[トランジットゲートウェイの関連付け提案の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDirectConnectGatewayAssociationProposal](#)」の「」を参照してください。

create-direct-connect-gateway-association

次のコード例は、create-direct-connect-gateway-association を使用する方法を示しています。

AWS CLI

仮想プライベートゲートウェイを Direct Connect ゲートウェイに関連付けるには

次の例では、仮想プライベートゲートウェイを Direct Connect ゲートウェイ vgw-6efe725e に関連付けます 5f294f92-bafb-4011-916d-9b0bexample。仮想プライベートゲートウェイがあるリージョンで コマンドを実行する必要があります。

コマンド:

```
aws directconnect create-direct-connect-gateway-association --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample --virtual-gateway-id vgw-6efe725e
```

出力:

```
{
  "directConnectGatewayAssociation": {
    "associationState": "associating",
    "virtualGatewayOwnerAccount": "123456789012",
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
```



```
    "virtualGatewayId": "vgw-6efe725e",
    "virtualGatewayRegion": "us-east-2"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDirectConnectGatewayAssociation](#)」の「」を参照してください。

create-direct-connect-gateway

次の例は、`create-direct-connect-gateway` を使用する方法を説明しています。

AWS CLI

Direct Connect ゲートウェイを作成するには

次の例では、という名前の Direct Connect ゲートウェイを作成します `DxGateway1`。

コマンド:

```
aws directconnect create-direct-connect-gateway --direct-connect-gateway-name "DxGateway1"
```

出力:

```
{
  "directConnectGateway": {
    "amazonSideAsn": 64512,
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
    "ownerAccount": "123456789012",
    "directConnectGatewayName": "DxGateway1",
    "directConnectGatewayState": "available"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDirectConnectGateway](#)」の「」を参照してください。

create-interconnect

次のコード例は、`create-interconnect` を使用する方法を示しています。

AWS CLI

パートナーのネットワークと の間に相互接続を作成するには AWS

次のcreate-interconnectコマンドは、AWS Direct Connect パートナーのネットワークと特定の AWS Direct Connect ロケーション間の相互接続を作成します。

```
aws directconnect create-interconnect --interconnect-name "1G Interconnect to AWS"
--bandwidth 1Gbps --location TIVIT
```

出力:

```
{
  "region": "sa-east-1",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "interconnectName": "1G Interconnect to AWS",
  "interconnectId": "dxcon-fgktov66",
  "interconnectState": "requested"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInterconnect](#)」の「」を参照してください。

create-lag

次のコード例は、create-lag を使用する方法を示しています。

AWS CLI

新しい接続LAGで を作成するには

次の例では、 を作成しLAG、帯域幅 1 Gbps LAGの に対して 2 つの新しい AWS Direct Connect 接続をリクエストします。

コマンド:

```
aws directconnect create-lag --location CSVA1 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 1GBLag
```

出力:

```
{
  "awsDevice": "CSVA1-23u8t1paz8iks",
  "numberOfConnections": 2,
  "lagState": "pending",
  "ownerAccount": "123456789012",
  "lagName": "1GBLag",
  "connections": [
    {
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffqr6x5q",
      "lagId": "dxlag-ffjhj91x",
      "connectionState": "requested",
      "bandwidth": "1Gbps",
      "location": "CSVA1",
      "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj91x",
      "region": "us-east-1"
    },
    {
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-fflqyj95",
      "lagId": "dxlag-ffjhj91x",
      "connectionState": "requested",
      "bandwidth": "1Gbps",
      "location": "CSVA1",
      "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj91x",
      "region": "us-east-1"
    }
  ],
  "lagId": "dxlag-ffjhj91x",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "CSVA1"
}
```

既存の接続LAGを使用して を作成するには

次の例では、アカウント内の既存の接続LAGから を作成し、既存の接続と同じ帯域幅と場所LAGを持つ の 2 番目の新しい接続をリクエストします。

コマンド:

```
aws directconnect create-lag --location EqDC2 --number-of-connections 2 --  
connections-bandwidth 1Gbps --lag-name 2ConnLAG --connection-id dxcon-fgk145dr
```

出力:

```
{  
  "awsDevice": "EqDC2-4h6ce2r1bes6",  
  "numberOfConnections": 2,  
  "lagState": "pending",  
  "ownerAccount": "123456789012",  
  "lagName": "2ConnLAG",  
  "connections": [  
    {  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-fh6ljcvo",  
      "lagId": "dxlag-fhccu14t",  
      "connectionState": "requested",  
      "bandwidth": "1Gbps",  
      "location": "EqDC2",  
      "connectionName": "Requested Connection 1 for Lag dxlag-fhccu14t",  
      "region": "us-east-1"  
    },  
    {  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-fgk145dr",  
      "lagId": "dxlag-fhccu14t",  
      "connectionState": "down",  
      "bandwidth": "1Gbps",  
      "location": "EqDC2",  
      "connectionName": "VAConn1",  
      "region": "us-east-1"  
    }  
  ],  
  "lagId": "dxlag-fhccu14t",  
  "minimumLinks": 0,  
  "connectionsBandwidth": "1Gbps",  
  "region": "us-east-1",  
  "location": "EqDC2"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateLag](#)」の「」を参照してください。

create-private-virtual-interface

次のコード例は、create-private-virtual-interface を使用する方法を示しています。

AWS CLI

プライベート仮想インターフェイスを作成するには

次のcreate-private-virtual-interfaceコマンドは、プライベート仮想インターフェイスを作成します。

```
aws directconnect create-private-virtual-interface --connection-id dxcon-ffjrkx17 --  
new-private-virtual-  
interface virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,authKey=asdf34exam  
aba37db6
```

出力:

```
{  
  "virtualInterfaceState": "pending",  
  "asn": 65000,  
  "vlan": 101,  
  "customerAddress": "192.168.1.2/30",  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-ffjrkx17",  
  "virtualGatewayId": "vgw-aba37db6",  
  "virtualInterfaceId": "dxvif-ffhkh74f",  
  "authKey": "asdf34example",  
  "routeFilterPrefixes": [],  
  "location": "TIVIT",  
  "customerRouterConfig": "<?xml version=\"1.0\" encoding=  
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhkh74f\">\n  <vlan>101</  
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n  <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n  <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>  
\n  <connection_type>private</connection_type>\n</logical_connection>\n",  
  "amazonAddress": "192.168.1.1/30",  
  "virtualInterfaceType": "private",  
  "virtualInterfaceName": "PrivateVirtualInterface"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePrivateVirtualInterface](#)」の「」を参照してください。

create-public-virtual-interface

次の例は、create-public-virtual-interface を使用する方法を説明しています。

AWS CLI

パブリック仮想インターフェイスを作成するには

次のcreate-public-virtual-interfaceコマンドは、パブリック仮想インターフェイスを作成します。

```
aws directconnect create-public-virtual-interface --connection-id dxcon-ffjrkrx17 --  
new-public-virtual-  
interface virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exam  
{cidr=203.0.113.4/30}]
```

出力:

```
{  
  "virtualInterfaceState": "verifying",  
  "asn": 65000,  
  "vlan": 2000,  
  "customerAddress": "203.0.113.2/30",  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-ffjrkrx17",  
  "virtualInterfaceId": "dxvif-fgh0hcrk",  
  "authKey": "asdf34example",  
  "routeFilterPrefixes": [  
    {  
      "cidr": "203.0.113.0/30"  
    },  
    {  
      "cidr": "203.0.113.4/30"  
    }  
  ],  
  "location": "TIVIT",  
  "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?  
>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</  
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n  <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n  <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>  
\n  <connection_type>public</connection_type>\n</logical_connection>\n",  
  "amazonAddress": "203.0.113.1/30",
```

```
"virtualInterfaceType": "public",  
"virtualInterfaceName": "PublicVirtualInterface"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePublicVirtualInterface](#)」の「」を参照してください。

create-transit-virtual-interface

次のコード例は、create-transit-virtual-interface を使用する方法を示しています。

AWS CLI

トランジット仮想インターフェイスを作成するには

次のcreate-transit-virtual-interface例では、指定された接続のトランジット仮想インターフェイスを作成します。

```
aws directconnect create-transit-virtual-interface \  
  --connection-id dxlag-fEXAMPLE \  
  --new-transit-virtual-interface "virtualInterfaceName=Example Transit Virtual  
Interface, vlan=126, asn=65110, mtu=1500, authKey=0xzxcgA9YoW9h58u8SvEXAMPLE, amazonAddress=192.1  
aada-5a1baEXAMPLE, tags=[{key=Tag, value=Example}]"
```

出力:

```
{  
  "virtualInterface": {  
    "ownerAccount": "1111222233333",  
    "virtualInterfaceId": "dxvif-fEXAMPLE",  
    "location": "loc1",  
    "connectionId": "dxlag-fEXAMPLE",  
    "virtualInterfaceType": "transit",  
    "virtualInterfaceName": "Example Transit Virtual Interface",  
    "vlan": 126,  
    "asn": 65110,  
    "amazonSideAsn": 4200000000,  
    "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",  
    "amazonAddress": "192.168.1.1/30",  
    "customerAddress": "192.168.1.2/30",  
    "addressFamily": "ipv4",
```

```

    "virtualInterfaceState": "pending",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\\\"UTF-8\\\"?>\\n<logical_connection id=\\\"dxvif-fEXAMPLE\\\">\\n  <vlan>126</
vlan>\\n  <customer_address>192.168.1.2/30</customer_address>\\n
  <amazon_address>192.168.1.1/30</amazon_address>\\n  <bgp_asn>65110</
bgp_asn>\\n  <bgp_auth_key>0xzxgA9YoW9h58u8Sv0mXRTw</bgp_auth_key>\\n
  <amazon_bgp_asn>4200000000</amazon_bgp_asn>\\n  <connection_type>transit</
connection_type>\\n</logical_connection>\\n\",
    "mtu": 1500,
    "jumboFrameCapable": true,
    "virtualGatewayId": "",
    "directConnectGatewayId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
    "routeFilterPrefixes": [],
    "bgpPeers": [
      {
        "bgpPeerId": "dxpeer-EXAMPLE",
        "asn": 65110,
        "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
        "addressFamily": "ipv4",
        "amazonAddress": "192.168.1.1/30",
        "customerAddress": "192.168.1.2/30",
        "bgpPeerState": "pending",
        "bgpStatus": "down",
        "awsDeviceV2": "loc1-26wz6vEXAMPLE"
      }
    ],
    "region": "sa-east-1",
    "awsDeviceV2": "loc1-26wz6vEXAMPLE",
    "tags": [
      {
        "key": "Tag",
        "value": "Example"
      }
    ]
  }
}

```

詳細については、[Direct Connect ユーザーガイドの「Direct Connect Gateway へのトランジット仮想インターフェイスの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitVirtualInterface](#)」の「」を参照してください。

delete-bgp-peer

次のコード例は、delete-bgp-peer を使用する方法を示しています。

AWS CLI

仮想インターフェイスからBGPピアを削除するには

次の例では、仮想インターフェイス からIPv6BGPピアを削除しますdxvif-fg1vuj3d。

コマンド:

```
aws directconnect delete-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --asn 64600
--customer-address 2001:db8:1100:2f0:0:1:9cb4:4216/125
```

出力:

```
{
  "virtualInterface": {
    "virtualInterfaceState": "available",
    "asn": 65000,
    "vlan": 125,
    "customerAddress": "169.254.255.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fguhmqlc",
    "addressFamily": "ipv4",
    "virtualGatewayId": "vgw-f9eb0c90",
    "virtualInterfaceId": "dxvif-fg1vuj3d",
    "authKey": "0xC_ukbCerl6EYA0example",
    "routeFilterPrefixes": [],
    "location": "EqDC2",
    "bgpPeers": [
      {
        "bgpStatus": "down",
        "customerAddress": "169.254.255.2/30",
        "addressFamily": "ipv4",
        "authKey": "0xC_ukbCerl6EYA0uexample",
        "bgpPeerState": "available",
        "amazonAddress": "169.254.255.1/30",
        "asn": 65000
      },
      {
        "bgpStatus": "down",
```

```

        "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
        "addressFamily": "ipv6",
        "authKey": "0xS27kAIU_VHPjjAexample",
        "bgpPeerState": "deleting",
        "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
        "asn": 64600
    }
],
"customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
  <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0xC_ukbCer16EYA0example</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
    "amazonAddress": "169.254.255.1/30",
    "virtualInterfaceType": "private",
    "virtualInterfaceName": "Test"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteBgpPeer](#)」の「」を参照してください。

delete-connection

次の例は、delete-connection を使用する方法を説明しています。

AWS CLI

接続を削除するには

次のdelete-connectionコマンドは、指定された接続を削除します。

```
aws directconnect delete-connection --connection-id dxcon-fg31dyv6
```

出力:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg31dyv6",
  "connectionState": "deleted",
}
```

```
"bandwidth": "1Gbps",
"location": "TIVIT",
"connectionName": "Connection to AWS",
"region": "sa-east-1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConnection](#)」の「」を参照してください。

delete-direct-connect-gateway-association

次の例は、delete-direct-connect-gateway-association を使用する方法を説明していません。

AWS CLI

Direct Connect ゲートウェイの関連付けを削除するには

次のdelete-direct-connect-gateway-association例では、指定された関連付け ID を持つトランジットゲートウェイとの Direct Connect ゲートウェイの関連付けを削除します。

```
aws directconnect delete-direct-connect-gateway-association --association-id
be85116d-46eb-4b43-a27a-da0c2ad648de
```

出力:

```
{
  "directConnectGatewayAssociation": {
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "directConnectGatewayOwnerAccount": "123456789012",
    "associationState": "disassociating",
    "associatedGateway": {
      "id": "tgw-095b3b0b54EXAMPLE",
      "type": "transitGateway",
      "ownerAccount": "123456789012",
      "region": "us-east-1"
    },
    "associationId": " be85116d-46eb-4b43-a27a-da0c2ad648deEXAMPLE ",
    "allowedPrefixesToDirectConnectGateway": [
      {
        "cidr": "192.0.1.0/28"
      }
    ]
  }
}
```

```
    }  
  ]  
}  
}
```

詳細については、AWS Direct Connect ユーザーガイドの「[Transit Gateway の関連付けと関連付け解除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDirectConnectGatewayAssociation](#)」の「」を参照してください。

delete-direct-connect-gateway

次のコード例は、delete-direct-connect-gateway を使用する方法を示しています。

AWS CLI

Direct Connect ゲートウェイを削除するには

次の例では、Direct Connect ゲートウェイ を削除します5f294f92-bafb-4011-916d-9b0bexample。

コマンド:

```
aws directconnect delete-direct-connect-gateway --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

出力:

```
{  
  "directConnectGateway": {  
    "amazonSideAsn": 64512,  
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",  
    "ownerAccount": "123456789012",  
    "directConnectGatewayName": "DxGateway1",  
    "directConnectGatewayState": "deleting"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDirectConnectGateway](#)」の「」を参照してください。

delete-interconnect

次のコード例は、delete-interconnect を使用する方法を示しています。

AWS CLI

インターコネクトを削除するには

次のdelete-interconnectコマンドは、指定されたインターコネクトを削除します。

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

出力:

```
{
  "interconnectState": "deleted"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInterconnect](#)」の「」を参照してください。

delete-lag

次のコード例は、delete-lag を使用する方法を示しています。

AWS CLI

を削除するには LAG

次の例では、指定された を削除しますLAG。

コマンド:

```
aws directconnect delete-lag --lag-id dxlag-ffrhowd9
```

出力:

```
{
  "awsDevice": "EqDC2-4h6ce2r1bes6",
  "numberOfConnections": 0,
  "lagState": "deleted",
  "ownerAccount": "123456789012",
}
```

```
"lagName": "TestLAG",
"connections": [],
"lagId": "dxlag-ffrhowd9",
"minimumLinks": 0,
"connectionsBandwidth": "1Gbps",
"region": "us-east-1",
"location": "EqDC2"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLag](#)」の「」を参照してください。

delete-virtual-interface

次の例は、delete-virtual-interface を使用する方法を説明しています。

AWS CLI

仮想インターフェイスを削除するには

次のdelete-virtual-interfaceコマンドは、指定された仮想インターフェイスを削除します。

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhkh74f
```

出力:

```
{
  "virtualInterfaceState": "deleting"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVirtualInterface](#)」の「」を参照してください。

describe-connection-loa

次のコード例は、describe-connection-loa を使用する方法を示しています。

AWS CLI

Linux または Mac OS X を使用した接続の LOA-CFA を記述するには

次の例では、接続の LOA-CFA について説明します `dxcon-fh6ayh1d`。LOA-CFA の内容は base64 でエンコードされています。このコマンドは、`--output` および `--query` パラメータを使用して出力を制御し、`loaContent` 構造の内容を取得します。コマンドの最後の部分は、base64ユーティリティを使用してコンテンツをデコードし、出力を PDF ファイルに送信します。

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --output text --query Loa.LoaContent/base64 --decode > myLoaCfa.pdf
```

Windows を使用した接続で LOA-CFA を記述するには

前の例では、base64ユーティリティを使用して出力をデコードする必要があります。Windows コンピュータでは、`certutil` 代わりに `certutil` を使用できます。次の例では、最初のコマンドで LOA-CFA を接続用に記述 `dxcon-fh6ayh1d` し、`--output` および `--query` パラメータを使用して出力を制御し、`loaContent` 構造の内容から `myLoaCfa.base64` というファイルに抽出します。2 番目のコマンドは、`certutil` ユーティリティを使用してファイルをデコードし、出力を PDF ファイルに送信します。

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --output text --query Loa.LoaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

出力の制御 AWS CLI の詳細については、[「コマンドラインインターフェイスユーザーガイド」の AWS 「コマンドラインインターフェイスからのコマンド出力の制御」](#) を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeConnectionLoa](#)」の「」を参照してください。

describe-connections-on-interconnect

次の例は、`describe-connections-on-interconnect` を使用する方法を説明しています。

AWS CLI

相互接続の接続を一覧表示するには

次の `describe-connections-on-interconnect` コマンドは、特定の相互接続でプロビジョニングされた接続を一覧表示します。

```
aws directconnect describe-connections-on-interconnect --interconnect-id dxcon-  
fgktov66
```

出力:

```
{  
  "connections": [  
    {  
      "partnerName": "TIVIT",  
      "vlan": 101,  
      "ownerAccount": "123456789012",  
      "connectionId": "dxcon-ffzc51m1",  
      "connectionState": "ordering",  
      "bandwidth": "500Mbps",  
      "location": "TIVIT",  
      "connectionName": "mydcinterconnect",  
      "region": "sa-east-1"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeConnectionsOnInterconnect](#)」の「」を参照してください。

describe-connections

次のコード例は、describe-connections を使用する方法を示しています。

AWS CLI

現在のリージョンのすべての接続を一覧表示するには

次のdescribe-connectionsコマンドは、現在のリージョンのすべての接続を一覧表示します。

```
aws directconnect describe-connections
```

出力:

```
{  
  "connections": [  
    {
```



```
{
  "awsDevice": "EqDC2-123h49s71dabc",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fguhmq1c",
  "lagId": "dxlag-ffrz71kw",
  "connectionState": "down",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "My_Connection",
  "loaIssueTime": 1491568964.0,
  "region": "us-east-1"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeConnections](#)」の「」を参照してください。

describe-direct-connect-gateway-association-proposals

次のコード例は、describe-direct-connect-gateway-association-proposals を使用する方法を示しています。

AWS CLI

Direct Connect ゲートウェイの関連付け提案を記述するには

次のdescribe-direct-connect-gateway-association-proposals例では、Direct Connect ゲートウェイの関連付け提案の詳細を表示します。

```
aws directconnect describe-direct-connect-gateway-association-proposals
```

出力:

```
{
  "directConnectGatewayAssociationProposals": [
    {
      "proposalId": "c2ede9b4-bbc6-4d33-923c-bc4feEXAMPLE",
      "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
      "directConnectGatewayOwnerAccount": "111122223333",
      "proposalState": "requested",
      "associatedGateway": {
```

```
        "id": "tgw-02f776b1a7EXAMPLE",
        "type": "transitGateway",
        "ownerAccount": "111122223333",
        "region": "us-east-1"
    },
    "existingAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.2.0/30"
        },
        {
            "cidr": "192.168.1.0/30"
        }
    ],
    "requestedAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.1.0/30"
        }
    ]
},
{
    "proposalId": "cb7f41cb-8128-43a5-93b1-dcaedEXAMPLE",
    "directConnectGatewayId": "11560968-4ac1-4fd3-bcb2-00599EXAMPLE",
    "directConnectGatewayOwnerAccount": "111122223333",
    "proposalState": "accepted",
    "associatedGateway": {
        "id": "tgw-045776b1a7EXAMPLE",
        "type": "transitGateway",
        "ownerAccount": "111122223333",
        "region": "us-east-1"
    },
    "existingAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.4.0/30"
        },
        {
            "cidr": "192.168.5.0/30"
        }
    ],
    "requestedAllowedPrefixesToDirectConnectGateway": [
        {
            "cidr": "192.168.5.0/30"
        }
    ]
}
}
```

```
    ]
  }
```

詳細については、AWS Direct Connect ユーザーガイドの「[Transit Gateway の関連付けと関連付け解除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDirectConnectGatewayAssociationProposals](#)」の「」を参照してください。

describe-direct-connect-gateway-associations

次のコード例は、describe-direct-connect-gateway-associations を使用する方法を示しています。

AWS CLI

Direct Connect ゲートウェイの関連付けを記述するには

次の例では、Direct Connect ゲートウェイ とのすべての関連付けについて説明します `5f294f92-bafb-4011-916d-9b0bexample`。

コマンド:

```
aws directconnect describe-direct-connect-gateway-associations --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

出力:

```
{
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoiOU83OTFodzdycnZCbkN4MExHeHVwQT09IiwieYyI6InIxTEN0UEVHV0I1UF1kaWFnN1",
  "directConnectGatewayAssociations": [
    {
      "associationState": "associating",
      "virtualGatewayOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualGatewayId": "vgw-6efe725e",
      "virtualGatewayRegion": "us-east-2"
    },
    {
      "associationState": "disassociating",
      "virtualGatewayOwnerAccount": "123456789012",
```

```
        "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
        "virtualGatewayId": "vgw-ebaa27db",
        "virtualGatewayRegion": "us-east-2"
    }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDirectConnectGatewayAssociations](#)」の「」を参照してください。

describe-direct-connect-gateway-attachments

次の例は、describe-direct-connect-gateway-attachments を使用する方法を説明しています。

AWS CLI

Direct Connect ゲートウェイアタッチメントを記述するには

次の例では、Direct Connect ゲートウェイ にアタッチされている仮想インターフェイスについて説明します 5f294f92-bafb-4011-916d-9b0bexample。

コマンド:

```
aws directconnect describe-direct-connect-gateway-attachments --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

出力:

```
{
  "directConnectGatewayAttachments": [
    {
      "virtualInterfaceOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualInterfaceRegion": "us-east-2",
      "attachmentState": "attaching",
      "virtualInterfaceId": "dxvif-fg9zyabc"
    }
  ],
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoibEhXd1NpUXF5RzhoL1JyUW52S1V2QT09IiwieYyI6Im5wQjFHQ0RyQUdRS3puNnNXcU
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDirectConnectGatewayAttachments](#)」の「」を参照してください。

describe-direct-connect-gateways

次の例は、describe-direct-connect-gateways を使用する方法を説明しています。

AWS CLI

Direct Connect ゲートウェイを記述するには

次の例では、すべての Direct Connect ゲートウェイについて説明します。

コマンド:

```
aws directconnect describe-direct-connect-gateways
```

出力:

```
{
  "directConnectGateways": [
    {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "cf68415c-f4ae-48f2-87a7-3b52cexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway2",
      "directConnectGatewayState": "available"
    },
    {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway1",
      "directConnectGatewayState": "available"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDirectConnectGateways](#)」の「」を参照してください。

describe-hosted-connections

次のコード例は、describe-hosted-connections を使用する方法を示しています。

AWS CLI

相互接続の接続を一覧表示するには

次の例では、特定の相互接続でプロビジョニングされた接続を一覧表示します。

コマンド:

```
aws directconnect describe-hosted-connections --connection-id dxcon-fgktov66
```

出力:

```
{
  "connections": [
    {
      "partnerName": "TIVIT",
      "vlan": 101,
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffzc51m1",
      "connectionState": "ordering",
      "bandwidth": "500Mbps",
      "location": "TIVIT",
      "connectionName": "mydcinterconnect",
      "region": "sa-east-1"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeHostedConnections](#)」の「」を参照してください。

describe-interconnect-loa

次のコード例は、describe-interconnect-loa を使用する方法を示しています。

AWS CLI

Linux または Mac OS X を使用したインターコネクトの LOA-CFA を記述するには

次の例では、相互接続の LOA-CFA について説明します `dxcon-fh6ayh1d`。LOA-CFA の内容は base64 でエンコードされています。このコマンドは、`--output` および `--query` パラメータを使用して出力を制御し、`loaContent` 構造の内容を取得します。コマンドの最後の部分は、base64 ユーティリティを使用してコンテンツをデコードし、出力を PDF ファイルに送信します。

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --output text --query loa.loaContent/base64 --decode > myLoaCfa.pdf
```

Windows を使用した相互接続の LOA-CFA を記述するには

前の例では、base64 ユーティリティを使用して出力をデコードする必要があります。Windows コンピュータでは、`certutil` 代わりに `certutil` を使用できます。次の例では、最初のコマンドで LOA-CFA for interconnect を記述 `dxcon-fh6ayh1d` し、`--output` および `--query` パラメータを使用して出力を制御し、`loaContent` 構造体の内容を `myLoaCfa.base64` というファイルに抽出します。2 番目のコマンドは、`certutil` ユーティリティを使用してファイルをデコードし、出力を PDF ファイルに送信します。

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

出力の制御 AWS CLI の詳細については、[「コマンドラインインターフェイスユーザーガイド」の AWS 「コマンドラインインターフェイスからのコマンド出力の制御」](#) を参照してください。

AWS

- API 詳細については、AWS CLI 「[コマンドリファレンス DescribeInterconnectLoa](#)」の「」を参照してください。

describe-interconnects

次のコード例は、`describe-interconnects` を使用する方法を示しています。

AWS CLI

相互接続を一覧表示するには

次の `describe-interconnects` コマンドは、AWS アカウントが所有する相互接続を一覧表示します。

```
aws directconnect describe-interconnects
```

出力:

```
{
  "interconnects": [
    {
      "region": "sa-east-1",
      "bandwidth": "1Gbps",
      "location": "TIVIT",
      "interconnectName": "1G Interconnect to AWS",
      "interconnectId": "dxcon-fgktov66",
      "interconnectState": "down"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInterconnects](#)」の「」を参照してください。

describe-lags

次の例は、describe-lags を使用する方法を説明しています。

AWS CLI

を記述するには LAGs

次のコマンドはLAGs、現在のリージョンのすべてのについて説明します。

コマンド:

```
aws directconnect describe-lags
```

出力:

```
{
  "lags": [
    {
      "awsDevice": "EqDC2-19y7z3m17xpuz",
      "numberOfConnections": 2,
      "lagState": "down",
    }
  ]
}
```



```
"ownerAccount": "123456789012",
"lagName": "DA-LAG",
"connections": [
  {
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffnikghc",
    "lagId": "dxlag-fgsu9erb",
    "connectionState": "requested",
    "bandwidth": "10Gbps",
    "location": "EqDC2",
    "connectionName": "Requested Connection 1 for Lag dxlag-fgsu9erb",
    "region": "us-east-1"
  },
  {
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fglgbdea",
    "lagId": "dxlag-fgsu9erb",
    "connectionState": "requested",
    "bandwidth": "10Gbps",
    "location": "EqDC2",
    "connectionName": "Requested Connection 2 for Lag dxlag-fgsu9erb",
    "region": "us-east-1"
  }
],
"lagId": "dxlag-fgsu9erb",
"minimumLinks": 0,
"connectionsBandwidth": "10Gbps",
"region": "us-east-1",
"location": "EqDC2"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLags](#)」の「」を参照してください。

describe-loa

次のコード例は、describe-loa を使用する方法を示しています。

AWS CLI

Linux または Mac OS X を使用した接続の LOA-CFA を記述するには

次の例では、接続の LOA-CFA について説明します `dxcon-fh6ayh1d`。LOA-CFA の内容は base64 でエンコードされています。このコマンドは、`--output` および `--query` パラメータを使用して出力を制御し、`loaContent` 構造の内容を取得します。コマンドの最後の部分は、base64ユーティリティを使用してコンテンツをデコードし、出力を PDF ファイルに送信します。

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query Loa.LoaContent/base64 --decode > myLoaCfa.pdf
```

Windows を使用した接続で LOA-CFA を記述するには

前の例では、base64ユーティリティを使用して出力をデコードする必要があります。Windows コンピュータでは、`certutil` 代わりに `certutil` を使用できます。次の例では、最初のコマンドで LOA-CFA を接続用に記述 `dxcon-fh6ayh1d` し、`--output` および `--query` パラメータを使用して出力を制御し、`loaContent` 構造の内容から `myLoaCfa.base64` というファイルに抽出します。2 番目のコマンドは、`certutil` ユティリティを使用してファイルをデコードし、出力を PDF ファイルに送信します。

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query Loa.LoaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

出力の制御 AWS CLI の詳細については、[「コマンドラインインターフェイスユーザーガイド」の AWS 「コマンドラインインターフェイスからのコマンド出力の制御」](#) を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLoa](#)」の「」を参照してください。

describe-locations

次の例は、`describe-locations` を使用する方法を説明しています。

AWS CLI

AWS Direct Connect のパートナーとロケーションを一覧表示するには

次の `describe-locations` コマンドは、Direct AWS Connect のパートナーと現在のリージョン内のロケーションを一覧表示します。

```
aws directconnect describe-locations
```

出力:

```
{
  "locations": [
    {
      "locationName": "NAP do Brasil, Barueri, Sao Paulo",
      "locationCode": "TNDB"
    },
    {
      "locationName": "Tivit - Site Transamerica (Sao Paulo)",
      "locationCode": "TIVIT"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLocations](#)」の「」を参照してください。

describe-tags

次の例は、describe-tags を使用する方法を説明しています。

AWS CLI

AWS Direct Connect リソースのタグを記述するには

次のコマンドは、接続 のタグについて説明します dxcon-abcabc12。

コマンド:

```
aws directconnect describe-tags --resource-arns arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12
```

出力:

```
{
  "resourceTags": [
    {
```

```
    "resourceArn": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-
abcabc12",
    "tags": [
      {
        "value": "VAConnection",
        "key": "Name"
      }
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

describe-virtual-gateways

次のコード例は、describe-virtual-gateways を使用する方法を示しています。

AWS CLI

仮想プライベートゲートウェイを一覧表示するには

次のdescribe-virtual-gatewaysコマンドは、AWS アカウントが所有する仮想プライベートゲートウェイを一覧表示します。

```
aws directconnect describe-virtual-gateways
```

出力:

```
{
  "virtualGateways": [
    {
      "virtualGatewayId": "vgw-aba37db6",
      "virtualGatewayState": "available"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVirtualGateways](#)」の「」を参照してください。

describe-virtual-interfaces

次のコード例は、describe-virtual-interfaces を使用する方法を示しています。

AWS CLI

すべての仮想インターフェイスを一覧表示するには

次のdescribe-virtual-interfacesコマンドは、AWS アカウントに関連付けられているすべての仮想インターフェイスに関する情報を一覧表示します。

```
aws directconnect describe-virtual-interfaces --connection-id dxcon-ffjrnx17
```

出力:

```
{
  "virtualInterfaces": [
    {
      "virtualInterfaceState": "down",
      "asn": 65000,
      "vlan": 101,
      "customerAddress": "192.168.1.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffjrnx17",
      "virtualGatewayId": "vgw-aba37db6",
      "virtualInterfaceId": "dxvif-ffhkh74f",
      "authKey": "asdf34example",
      "routeFilterPrefixes": [],
      "location": "TIVIT",
      "customerRouterConfig": "<?xml version=\"1.0\" encoding=
UTF-8\"?>\n<logical_connection id=\"dxvif-ffhkh74f\">\n  <vlan>101</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
  <amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>private</connection_type>\n</logical_connection>\n",
      "amazonAddress": "192.168.1.1/30",
      "virtualInterfaceType": "private",
      "virtualInterfaceName": "PrivateVirtualInterface"
    },
    {
      "virtualInterfaceState": "verifying",
      "asn": 65000,
      "vlan": 2000,
```

```

    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualGatewayId": "",
    "virtualInterfaceId": "dxvif-fgh0hcrk",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
      {
        "cidr": "203.0.113.4/30"
      },
      {
        "cidr": "203.0.113.0/30"
      }
    ],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\\\"UTF-8\\\"?>\\n<logical_connection id=\"dxvif-fgh0hcrk\\\">\\n  <vlan>2000</
vlan>\\n  <customer_address>203.0.113.2/30</customer_address>\\n
  <amazon_address>203.0.113.1/30</amazon_address>\\n  <bgp_asn>65000</bgp_asn>\\n
  <bgp_auth_key>asdf34example</bgp_auth_key>\\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\\n  <connection_type>public</connection_type>\\n</logical_connection>\\n",
    "amazonAddress": "203.0.113.1/30",
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVirtualInterfaces](#)」の「」を参照してください。

disassociate-connection-from-lag

次の例は、disassociate-connection-from-lag を使用する方法を説明しています。

AWS CLI

から接続の関連付けを解除するには LAG

次の例では、指定された接続と指定された の関連付けを解除しますLAG。

コマンド:

```
aws directconnect disassociate-connection-from-lag --lag-id dxlag-fhccu14t --  
connection-id dxcon-fg9607vm
```

出力:

```
{  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-fg9607vm",  
  "connectionState": "requested",  
  "bandwidth": "1Gbps",  
  "location": "EqDC2",  
  "connectionName": "Con2ForLag",  
  "region": "us-east-1"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateConnectionFromLag](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

AWS Direct Connect リソースにタグを追加するには

次のコマンドは、キーNameと 値を持つタグVAConnectionを接続 に追加しますdxcon-abcabc12。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-  
east-1:123456789012:dxcon/dxcon-abcabc12 --tags "key=Name,value=VAConnection"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

AWS Direct Connect リソースからタグを削除するには

次のコマンドは、キーを持つ タグを接続 Name から削除します `dxcon-abcabc12`。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tag-keys Name
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-direct-connect-gateway-association

次の例は、`update-direct-connect-gateway-association` を使用する方法を説明していません。

AWS CLI

Direct Connect ゲートウェイ関連付けの指定された属性を更新するには

次の `update-direct-connect-gateway-association` 例では、指定された CIDR ブロックを Direct Connect ゲートウェイの関連付けに追加します。

```
aws directconnect update-direct-connect-gateway-association \  
  --association-id 820a6e4f-5374-4004-8317-3f64bEXAMPLE \  
  --add-allowed-prefixes-to-direct-connect-gateway cidr=192.168.2.0/30
```

出力:

```
{  
  "directConnectGatewayAssociation": {  
    "directConnectGatewayId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",  
    "directConnectGatewayOwnerAccount": "111122223333",  
    "associationState": "updating",  
    "associatedGateway": {  
      "id": "tgw-02f776b1a7EXAMPLE",  
      "type": "transitGateway",
```



```
    "ownerAccount": "111122223333",
    "region": "us-east-1"
  },
  "associationId": "820a6e4f-5374-4004-8317-3f64bEXAMPLE",
  "allowedPrefixesToDirectConnectGateway": [
    {
      "cidr": "192.168.2.0/30"
    },
    {
      "cidr": "192.168.1.0/30"
    }
  ]
}
```

詳細については、[「Direct Connect ユーザーガイド」の「Direct Connect Gateway の使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateDirectConnectGatewayAssociation](#)」の「」を参照してください。

update-lag

次のコード例は、update-lag を使用する方法を示しています。

AWS CLI

を更新するには LAG

次の例では、指定された の名前を変更しますLAG。

コマンド:

```
aws directconnect update-lag --lag-id dxlag-ffjhj9lx --lag-name 2ConnLag
```

出力:

```
{
  "awsDevice": "CSVA1-23u8t1paz8iks",
  "numberOfConnections": 2,
  "lagState": "down",
  "ownerAccount": "123456789012",
```

```
"lagName": "2ConnLag",
"connections": [
  {
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fflqyj95",
    "lagId": "dxlag-ffjhj9lx",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "CSVA1",
    "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
    "region": "us-east-1"
  },
  {
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffqr6x5q",
    "lagId": "dxlag-ffjhj9lx",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "CSVA1",
    "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
    "region": "us-east-1"
  }
],
"lagId": "dxlag-ffjhj9lx",
"minimumLinks": 0,
"connectionsBandwidth": "1Gbps",
"region": "us-east-1",
"location": "CSVA1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLag](#)」の「」を参照してください。

update-virtual-interface-attributes

次のコード例は、update-virtual-interface-attributes を使用する方法を示しています。

AWS CLI

仮想インターフェイスMTUのを更新するには

次のupdate-virtual-interface-attributes例では、指定された仮想インターフェイスMTUのを更新します。

```
aws directconnect update-virtual-interface-attributes \
  --virtual-interface-id dxvif-fEXAMPLE \
  --mtu 1500
```

出力:

```
{
  "ownerAccount": "1111222233333",
  "virtualInterfaceId": "dxvif-fEXAMPLE",
  "location": "loc1",
  "connectionId": "dxlag-fEXAMPLE",
  "virtualInterfaceType": "transit",
  "virtualInterfaceName": "example transit virtual interface",
  "vlan": 125,
  "asn": 650001,
  "amazonSideAsn": 64512,
  "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",
  "amazonAddress": "169.254.248.1/30",
  "customerAddress": "169.254.248.2/30",
  "addressFamily": "ipv4",
  "virtualInterfaceState": "down",
  "customerRouterConfig": "<?xml version='1.0' encoding='UTF-8'?>
<logical_connection id='dxvif-fEXAMPLE'>
  <vlan>125</vlan>
  <customer_address>169.254.248.2/30</customer_address>
  <amazon_address>169.254.248.1/30</amazon_address>
  <bgp_asn>650001</bgp_asn>
  <bgp_auth_key>0xzxcgA9YoW9h58u8SEXAMPLE</bgp_auth_key>
  <amazon_bgp_asn>64512</amazon_bgp_asn>
  <connection_type>transit</connection_type>
</logical_connection>
",
  "mtu": 1500,
  "jumboFrameCapable": true,
  "virtualGatewayId": "",
  "directConnectGatewayId": "879b76a1-403d-4700-8b53-4a56ed85436e",
  "routeFilterPrefixes": [],
  "bgpPeers": [
    {
      "bgpPeerId": "dxpeer-fEXAMPLE",
      "asn": 650001,
      "authKey": "0xzxcgA9YoW9h58u8SEXAMPLE",
      "addressFamily": "ipv4",
      "amazonAddress": "169.254.248.1/30",
      "customerAddress": "169.254.248.2/30",
      "bgpPeerState": "available",
      "bgpStatus": "down",
```

```
        "awsDeviceV2": "loc1-26wz6vEXAMPLE"
    }
],
"region": "sa-east-1",
"awsDeviceV2": "loc1-26wz6vEXAMPLE",
"tags": []
}
```

詳細については、AWS Direct Connect ユーザーガイドの「[プライベート仮想インターフェイスまたはトランジット仮想インターフェイスMTUのネットワークの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVirtualInterfaceAttributes](#)」の「」を参照してください。

AWS Directory Service を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Directory Service。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-directories

次のコード例は、describe-directories を使用する方法を示しています。

AWS CLI

ディレクトリの詳細を取得するには

次のdescribe-directories例では、指定されたディレクトリの詳細を表示します。

```
aws ds describe-directories \  
  --directory-id d-a1b2c3d4e5
```

出力:

```
{  
  "DirectoryDescriptions": [  
    {  
      "DirectoryId": "d-a1b2c3d4e5",  
      "Name": "mydirectory.example.com",  
      "ShortName": "mydirectory",  
      "Size": "Small",  
      "Edition": "Standard",  
      "Alias": "d-a1b2c3d4e5",  
      "AccessUrl": "d-a1b2c3d4e5.awsapps.com",  
      "Stage": "Active",  
      "ShareStatus": "Shared",  
      "ShareMethod": "HANDSHAKE",  
      "ShareNotes": "These are my share notes",  
      "LaunchTime": "2019-07-08T15:33:46.327000-07:00",  
      "StageLastUpdatedDateTime": "2019-07-08T15:59:12.307000-07:00",  
      "Type": "SharedMicrosoftAD",  
      "SsoEnabled": false,  
      "DesiredNumberOfDomainControllers": 0,  
      "OwnerDirectoryDescription": {  
        "DirectoryId": "d-b2c3d4e5f6",  
        "AccountId": "123456789111",  
        "DnsIpAddrs": [  
          "203.113.0.248",  
          "203.113.0.253"  
        ],  
        "VpcSettings": {  
          "VpcId": "vpc-a1b2c3d4",  
          "SubnetIds": [  
            "subnet-a1b2c3d4",  
            "subnet-d4c3b2a1"  
          ],  
          "AvailabilityZones": [  
            "us-west-2a",  
            "us-west-2c"  
          ]  
        }  
      }  
    ]  
  }  
}
```

```
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDirectories](#)」の「」を参照してください。

describe-trusts

次のコード例は、describe-trusts を使用する方法を示しています。

AWS CLI

信頼関係の詳細を取得するには

次のdescribe-trusts例では、指定されたディレクトリの信頼関係の詳細を表示します。

```
aws ds describe-trusts \  
  --directory-id d-a1b2c3d4e5
```

出力:

```
{  
  "Trusts": [  
    {  
      "DirectoryId": "d-a1b2c3d4e5",  
      "TrustId": "t-9a8b7c6d5e",  
      "RemoteDomainName": "other.example.com",  
      "TrustType": "Forest",  
      "TrustDirection": "Two-Way",  
      "TrustState": "Verified",  
      "CreatedDateTime": "2017-06-20T18:08:45.614000-07:00",  
      "LastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",  
      "StateLastUpdatedDateTime": "2019-06-04T10:52:12.410000-07:00",  
      "SelectiveAuth": "Disabled"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrusts](#)」の「」を参照してください。

AWS DMS を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS DMS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags-to-resource

次の例は、add-tags-to-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを追加するには

次のadd-tags-to-resource例では、レプリケーションインスタンスにタグを追加します。

```
aws dms add-tags-to-resource \  
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  \  
  --tags Key=Environment,Value=PROD Key=Project,Value=dbMigration
```

このコマンドでは何も出力されません。

詳細については、AWS 「データベース移行サービスユーザーガイド」の [「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AddTagsToResource](#)」の「」を参照してください。

create-endpoint

次の例は、create-endpoint を使用する方法を説明しています。

AWS CLI

エンドポイントを作成するには

次のcreate-endpoint例では、Amazon S3 ソースのエンドポイントを作成します。

```
aws dms create-endpoint \  
  --endpoint-type source \  
  --engine-name s3 \  
  --endpoint-identifier src-endpoint \  
  --s3-settings file://s3-settings.json
```

s3-settings.json の内容:

```
{  
  "BucketName": "my-corp-data",  
  "BucketFolder": "sourcedata",  
  "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role"  
}
```

出力:

```
{  
  "Endpoint": {  
    "EndpointIdentifier": "src-endpoint",  
    "EndpointType": "SOURCE",  
    "EngineName": "s3",  
    "EngineDisplayName": "Amazon S3",  
    "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-  
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\\n;",  
    "Status": "active",  
    "EndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U",  
    "SslMode": "none",  
    "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",  
    "S3Settings": {  
      "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-  
role",  
      "CsvRowDelimiter": "\\n",  
    }  
  }  
}
```



```
        "CsvDelimiter": ",",
        "BucketFolder": "sourcedata",
        "BucketName": "my-corp-data",
        "CompressionType": "NONE",
        "EnableStatistics": true
    }
}
```

詳細については、AWS「[Database Migration Service ユーザーガイド](#)」の「[エンドポイントの使用 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス CreateEndpoint](#)」の「」を参照してください。

create-event-subscription

次のコード例は、create-event-subscription を使用する方法を示しています。

AWS CLI

イベントサブスクリプションを一覧表示するには

次のcreate-event-subscription例では、Amazon SNSトピック () へのイベントサブスクリプションを作成しますmy-sns-topic。

```
aws dms create-event-subscription \
  --subscription-name my-dms-events \
  --sns-topic-arn arn:aws:sns:us-east-1:123456789012:my-sns-topic
```

出力:

```
{
  "EventSubscription": {
    "CustomerAwsId": "123456789012",
    "CustSubscriptionId": "my-dms-events",
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
    "Status": "creating",
    "SubscriptionCreationTime": "2020-05-21 21:58:38.598",
    "Enabled": true
  }
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[イベントと通知の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEventSubscription](#)」の「」を参照してください。

create-replication-instance

次のコード例は、create-replication-instance を使用する方法を示しています。

AWS CLI

レプリケーションインスタンスを作成するには

次のcreate-replication-instance例では、レプリケーションインスタンスを作成します。

```
aws dms create-replication-instance \  
  --replication-instance-identifier my-repl-instance \  
  --replication-instance-class dms.t2.micro \  
  --allocated-storage 5
```

出力:

```
{  
  "ReplicationInstance": {  
    "ReplicationInstanceIdentifier": "my-repl-instance",  
    "ReplicationInstanceClass": "dms.t2.micro",  
    "ReplicationInstanceStatus": "creating",  
    "AllocatedStorage": 5,  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-f839b688",  
        "Status": "active"  
      }  
    ],  
    "ReplicationSubnetGroup": {  
      "ReplicationSubnetGroupIdentifier": "default",  
      "ReplicationSubnetGroupDescription": "default",  
      "VpcId": "vpc-136a4c6a",  
      "SubnetGroupStatus": "Complete",  
      "Subnets": [  
        {  
          "SubnetIdentifier": "subnet-da327bf6",
```

```
        "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-42599426",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1d"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-bac383e0",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1c"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-6746046b",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-d7c825e8",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1e"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-cbfff283",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
        },
        "SubnetStatus": "Active"
    }
]
},
"PreferredMaintenanceWindow": "sat:12:35-sat:13:05",
"PendingModifiedValues": {},
```

```
    "MultiAZ": false,
    "EngineVersion": "3.3.2",
    "AutoMinorVersionUpgrade": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:ZK2VQBUWFDBAWHIXHAYG5G2PKY",
    "PubliclyAccessible": true
  }
}
```

詳細については、AWS 「 Database Migration Service ユーザーガイド」の[AWS DMS 「レプリケーションインスタンスの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[CreateReplicationInstance](#)」の「」を参照してください。

create-replication-subnet-group

次のコード例は、create-replication-subnet-group を使用する方法を示しています。

AWS CLI

サブネットグループを作成するには

次のcreate-replication-subnet-group例では、3つのサブネットで構成されるグループを作成します。

```
aws dms create-replication-subnet-group \
  --replication-subnet-group-identifier my-subnet-group \
  --replication-subnet-group-description "my subnet group" \
  --subnet-ids subnet-da327bf6 subnet-bac383e0 subnet-d7c825e8
```

出力:

```
{
  "ReplicationSubnetGroup": {
    "ReplicationSubnetGroupIdentifier": "my-subnet-group",
    "ReplicationSubnetGroupDescription": "my subnet group",
    "VpcId": "vpc-136a4c6a",
    "SubnetGroupStatus": "Complete",
```

```
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-da327bf6",
        "SubnetAvailabilityZone": {
          "Name": "us-east-1a"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-bac383e0",
        "SubnetAvailabilityZone": {
          "Name": "us-east-1c"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-d7c825e8",
        "SubnetAvailabilityZone": {
          "Name": "us-east-1e"
        },
        "SubnetStatus": "Active"
      }
    ]
  }
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の[「レプリケーションインスタンスのネットワークのセットアップ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateReplicationSubnetGroup](#)」の「」を参照してください。

create-replication-task

次のコード例は、create-replication-task を使用する方法を示しています。

AWS CLI

レプリケーションタスクを作成するには

次のcreate-replication-task例では、レプリケーションタスクを作成します。

```
aws dms create-replication-task \
```

```

--replication-task-identifier movedata \
--source-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
--target-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U \
--replication-instance-arn $RI_ARN \
--migration-type full-load \
--table-mappings file://table-mappings.json

```

table-mappings.json の内容:

```

{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "prodrep",
        "table-name": "%"
      },
      "rule-action": "include",
      "filters": []
    }
  ]
}

```

出力:

```

{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": ...output omitted... ,
    "ReplicationTaskSettings": ...output omitted... ,
    "Status": "creating",
    "ReplicationTaskCreationDate": 1590524772.505,
  }
}

```

```
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}
```

詳細については、AWS 「 Database Migration Service ユーザーガイド」の「[タスクの操作 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[CreateReplicationTask](#)」の「」を参照してください。

delete-connection

次の例は、delete-connection を使用する方法を説明しています。

AWS CLI

接続を削除するには

次のdelete-connection例では、レプリケーションインスタンスからエンドポイントの関連付けを解除します。

```
aws dms delete-connection \
  --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
  --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

出力:

```
{
  "Connection": {
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "Status": "deleting",
    "EndpointIdentifier": "src-database-1",
    "ReplicationInstanceIdentifier": "my-repl-instance"
  }
}
```

詳細については、AWS 「 Database Migration Service ユーザーガイド」のhttps://docs.aws.amazon.com/dms/latest/userguide/CHAP_1_Endpoints.Creating.htmlを参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteConnection](#)」の「」を参照してください。

delete-endpoint

次の例は、delete-endpoint を使用する方法を説明しています。

AWS CLI

エンドポイントを削除するには

次のdelete-endpoint例では、エンドポイントを削除します。

```
aws dms delete-endpoint \  
  --endpoint-arn arn:aws:dms:us-  
east-1:123456789012:endpoint:0UJJVX04XZ4CYTSEG5XGMN2R3Y
```

出力:

```
{  
  "Endpoint": {  
    "EndpointIdentifier": "src-endpoint",  
    "EndpointType": "SOURCE",  
    "EngineName": "s3",  
    "EngineDisplayName": "Amazon S3",  
    "ExtraConnectionAttributes": "bucketFolder=sourcedata;bucketName=my-corp-  
data;compressionType=NONE;csvDelimiter=,;csvRowDelimiter=\n",  
    "Status": "deleting",  
    "EndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:0UJJVX04XZ4CYTSEG5XGMN2R3Y",  
    "SslMode": "none",  
    "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",  
    "S3Settings": {  
      "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-  
role",  
      "CsvRowDelimiter": "\n",  
      "CsvDelimiter": ",",  
      "BucketFolder": "sourcedata",
```



```
        "BucketName": "my-corp-data",
        "CompressionType": "NONE",
        "EnableStatistics": true
    }
}
```

詳細については、AWS「[Database Migration Service ユーザーガイド](#)」の「[エンドポイントの使用 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス DeleteEndpoint](#)」の「」を参照してください。

delete-event-subscription

次のコード例は、delete-event-subscription を使用する方法を示しています。

AWS CLI

イベントサブスクリプションを削除するには

次のdelete-event-subscription例では、Amazon SNSトピックのサブスクリプションを削除します。

```
aws dms delete-event-subscription \
  --subscription-name "my-dms-events"
```

出力:

```
{
  "EventSubscription": {
    "CustomerAwsId": "123456789012",
    "CustSubscriptionId": "my-dms-events",
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
    "Status": "deleting",
    "SubscriptionCreationTime": "2020-05-21 21:58:38.598",
    "Enabled": true
  }
}
```

詳細については、AWS「[Database Migration Service ユーザーガイド](#)」の「[イベントと通知の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteEventSubscription](#)」の「」を参照してください。

delete-replication-instance

次のコード例は、delete-replication-instance を使用する方法を示しています。

AWS CLI

レプリケーションインスタンスを削除するには

次の delete-replication-instance の例では、レプリケーション インスタンスを削除します。

```
aws dms delete-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

出力:

```
{  
  "ReplicationInstance": {  
    "ReplicationInstanceIdentifier": "my-repl-instance",  
    "ReplicationInstanceClass": "dms.t2.micro",  
    "ReplicationInstanceStatus": "deleting",  
    "AllocatedStorage": 5,  
    "InstanceCreateTime": 1590011235.952,  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-f839b688",  
        "Status": "active"  
      }  
    ],  
    "AvailabilityZone": "us-east-1e",  
    "ReplicationSubnetGroup": {  
      "ReplicationSubnetGroupIdentifier": "default",  
      "ReplicationSubnetGroupDescription": "default",  
      "VpcId": "vpc-136a4c6a",  
      "SubnetGroupStatus": "Complete",  
      "Subnets": [  
        {  
          "SubnetIdentifier": "subnet-da327bf6",
```

```
        "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-42599426",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1d"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-bac383e0",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1c"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-6746046b",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-d7c825e8",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1e"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-cbfff283",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
        },
        "SubnetStatus": "Active"
    }
]
},
"PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
"PendingModifiedValues": {},
```

```
    "MultiAZ": true,
    "EngineVersion": "3.3.2",
    "AutoMinorVersionUpgrade": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "ReplicationInstancePublicIpAddress": "54.225.120.92",
    "ReplicationInstancePrivateIpAddress": "172.31.30.121",
    "ReplicationInstancePublicIpAddresses": [
      "54.225.120.92",
      "3.230.18.248"
    ],
    "ReplicationInstancePrivateIpAddresses": [
      "172.31.30.121",
      "172.31.75.90"
    ],
    "PubliclyAccessible": true,
    "SecondaryAvailabilityZone": "us-east-1b"
  }
}
```

詳細については、AWS 「 Database Migration Service ユーザーガイド」の[AWS DMS 「レプリケーションインスタンスの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[DeleteReplicationInstance](#)」の「」を参照してください。

delete-replication-subnet-group

次の例は、delete-replication-subnet-group を使用する方法を説明しています。

AWS CLI

サブネットグループを削除するには

次のdelete-replication-subnet-group例では、サブネットグループを削除します。

```
aws dms delete-replication-subnet-group \
--replication-subnet-group-identifier my-subnet-group
```

出力:

(none)

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[レプリケーションインスタンスのネットワークのセットアップ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteReplicationSubnetGroup](#)」の「」を参照してください。

delete-replication-task

次の例は、delete-replication-task を使用する方法を説明しています。

AWS CLI

レプリケーションタスクを削除するには

次のdelete-replication-task例では、レプリケーションタスクを削除します。

```
aws dms delete-replication-task \  
  --replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

出力:

```
{  
  "ReplicationTask": {  
    "ReplicationTaskIdentifier": "moveit2",  
    "SourceEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",  
    "TargetEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "MigrationType": "full-load",  
    "TableMappings": "...output omitted...",  
    "ReplicationTaskSettings": "...output omitted...",  
    "Status": "deleting",  
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",  
    "ReplicationTaskCreationDate": 1590524772.505,  
    "ReplicationTaskStartDate": 1590789988.677,  
    "ReplicationTaskArn": "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
```

```
}  
}
```

詳細については、AWS 「 Database Migration Service ユーザーガイド」の「[タスクの操作 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[DeleteReplicationTask](#)」の「」を参照してください。

describe-account-attributes

次のコード例は、describe-account-attributes を使用する方法を示しています。

AWS CLI

アカウントの属性を記述するには

次のdescribe-account-attributes例では、AWS アカウントの属性を一覧表示します。

```
aws dms describe-account-attributes
```

出力:

```
{  
  "AccountQuotas": [  
    {  
      "AccountQuotaName": "ReplicationInstances",  
      "Used": 1,  
      "Max": 20  
    },  
    {  
      "AccountQuotaName": "AllocatedStorage",  
      "Used": 5,  
      "Max": 10000  
    },  
    ...remaining output omitted...  
  ],  
  "UniqueAccountIdentifier": "cqahfbfy5xee"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountAttributes](#)」の「」を参照してください。

describe-certificates

次の例は、describe-certificates を使用する方法を説明しています。

AWS CLI

使用可能な証明書を一覧表示するには

次のdescribe-certificates例では、AWS アカウントで使用可能な証明書を一覧表示します。

```
aws dms describe-certificates
```

出力:

```
{
  "Certificates": [
    {
      "CertificateIdentifier": "my-cert",
      "CertificateCreationDate": 1543259542.506,
      "CertificatePem": "-----BEGIN CERTIFICATE-----
\nMIID9DCCAtygAwIBAgIBQjANBgkqhkiG9w0BAQ ...U"
      ... remaining output omitted ...
    }
  ]
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[の使用SSL](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCertificates](#)」の「」を参照してください。

describe-connections

次の例は、describe-connections を使用する方法を説明しています。

AWS CLI

接続を記述するには

次のdescribe-connections例は、レプリケーションインスタンスとエンドポイントの間でテストした接続を一覧表示します。

```
aws dms describe-connections
```

出力:

```
{
  "Connections": [
    {
      "Status": "successful",
      "ReplicationInstanceIdentifier": "test",
      "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:ZW5UAN6P4E77EC7YWHK4RZZ3BE",
      "EndpointIdentifier": "testsrc1",
      "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:6UTDJGB0US3VI3SUWA66XFJCJQ"
    }
  ]
}
```

詳細については、AWS「[Database Migration Service ユーザーガイド](#)」の「[ソースエンドポイントとターゲットエンドポイントの作成](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeConnections](#)」の「」を参照してください。

describe-endpoint-types

次のコード例は、describe-endpoint-types を使用する方法を示しています。

AWS CLI

使用可能なエンドポイントタイプを一覧表示するには

次のdescribe-endpoint-types例では、使用可能な MySQL エンドポイントタイプを一覧表示します。


```
aws dms describe-endpoint-types \  
  --filters "Name=engine-name,Values=mysql"
```

出力:

```
{  
  "SupportedEndpointTypes": [  
    {  
      "EngineName": "mysql",  
      "SupportsCDC": true,  
      "EndpointType": "source",  
      "EngineDisplayName": "MySQL"  
    },  
    {  
      "EngineName": "mysql",  
      "SupportsCDC": true,  
      "EndpointType": "target",  
      "EngineDisplayName": "MySQL"  
    }  
  ]  
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「エンドポイントの使用 AWS DMS」 <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`_`」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEndpointTypes](#)」の「」を参照してください。

describe-endpoints

次の例は、describe-endpoints を使用する方法を説明しています。

AWS CLI

エンドポイントを記述するには

次のdescribe-endpoints例では、AWS アカウントのエンドポイントを一覧表示します。

```
aws dms describe-endpoints
```

出力:

```
{
  "Endpoints": [
    {
      "Username": "dms",
      "Status": "active",
      "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:SF2W0FLWYWKVE0HID2EKLP3SJI",
      "ServerName": "ec2-52-32-48-61.us-west-2.compute.amazonaws.com",
      "EndpointType": "SOURCE",
      "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/94d5c4e7-4e4c-44be-b58a-c8da7adf57cd",
      "DatabaseName": "test",
      "EngineName": "mysql",
      "EndpointIdentifier": "pri100",
      "Port": 8193
    },
    {
      "Username": "admin",
      "Status": "active",
      "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:TJJZCIH3CJ24TJRU4VC32WEWFR",
      "ServerName": "test.example.com",
      "EndpointType": "SOURCE",
      "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/2431021b-1cf2-a2d4-77b2-59a9e4bce323",
      "DatabaseName": "EMPL",
      "EngineName": "oracle",
      "EndpointIdentifier": "test",
      "Port": 1521
    }
  ]
}
```

詳細については、AWS「[Database Migration Service ユーザーガイド](#)」の「[エンドポイントの使用 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス DescribeEndpoints](#)」の「」を参照してください。

describe-event-categories

次のコード例は、describe-event-categories を使用する方法を示しています。

AWS CLI

イベントカテゴリを記述するには

次のdescribe-event-categories例では、使用可能なイベントカテゴリを一覧表示します。

```
aws dms describe-event-categories
```

出力:

```
{
  "EventCategoryGroupList": [
    {
      "SourceType": "replication-instance",
      "EventCategories": [
        "low storage",
        "configuration change",
        "maintenance",
        "deletion",
        "creation",
        "failover",
        "failure"
      ]
    },
    {
      "SourceType": "replication-task",
      "EventCategories": [
        "configuration change",
        "state change",
        "deletion",
        "creation",
        "failure"
      ]
    }
  ]
}
```

詳細については、AWS「[Database Migration Service ユーザーガイド](#)」の「[イベントと通知の操作](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeEventCategories](#)」の「」を参照してください。

describe-event-subscriptions

次の例は、describe-event-subscriptions を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを記述するには

次のdescribe-event-subscriptions例では、Amazon SNSトピックへのイベントサブスクリプションを一覧表示します。

```
aws dms describe-event-subscriptions
```

出力:

```
{
  "EventSubscriptionsList": [
    {
      "CustomerAwsId": "123456789012",
      "CustSubscriptionId": "my-dms-events",
      "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
      "Status": "deleting",
      "SubscriptionCreationTime": "2020-05-21 22:28:51.924",
      "Enabled": true
    }
  ]
}
```

詳細については、AWS「[Database Migration Service ユーザーガイド](#)」の「[イベントと通知の操作](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeEventSubscriptions](#)」の「」を参照してください。

describe-events

次のコード例は、describe-events を使用する方法を示しています。

AWS CLI

DMSイベントを一覧表示するには

次のdescribe-events例では、レプリケーションインスタンスから発生したイベントを一覧表示します。

```
aws dms describe-events \  
  --source-type "replication-instance"
```

出力:

```
{  
  "Events": [  
    {  
      "SourceIdentifier": "my-repl-instance",  
      "SourceType": "replication-instance",  
      "Message": "Replication application shutdown",  
      "EventCategories": [],  
      "Date": 1590771645.776  
    }  
  ]  
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[イベントと通知の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

describe-orderable-replication-instances

次の例は、describe-orderable-replication-instances を使用する方法を説明しています。

AWS CLI

注文可能なレプリケーションインスタンスを記述するには

次のdescribe-orderable-replication-instances例では、注文できるレプリケーションインスタンスタイプを一覧表示します。

```
aws dms describe-orderable-replication-instances
```

出力:

```
{
  "OrderableReplicationInstances": [
    {
      "EngineVersion": "3.3.2",
      "ReplicationInstanceClass": "dms.c4.2xlarge",
      "StorageType": "gp2",
      "MinAllocatedStorage": 5,
      "MaxAllocatedStorage": 6144,
      "DefaultAllocatedStorage": 100,
      "IncludedAllocatedStorage": 100,
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ]
    },
    {
      "EngineVersion": "3.3.2",
      "ReplicationInstanceClass": "dms.c4.4xlarge",
      "StorageType": "gp2",
      "MinAllocatedStorage": 5,
      "MaxAllocatedStorage": 6144,
      "DefaultAllocatedStorage": 100,
      "IncludedAllocatedStorage": 100,
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ]
    },
    ...remaining output omitted...
  ]
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の[AWS DMS 「レプリケーションインスタンスの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeOrderableReplicationInstances](#)」の「」を参照してください。

describe-refresh-schemas-status

次の例は、describe-refresh-schemas-status を使用する方法を説明しています。

AWS CLI

エンドポイントの更新ステータスを一覧表示するには

次のdescribe-refresh-schemas-status例では、以前の更新リクエストのステータスを返します。

```
aws dms describe-refresh-schemas-status \  
  --endpoint-arn arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA
```

出力:

```
{  
  "RefreshSchemasStatus": {  
    "EndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "Status": "successful",  
    "LastRefreshDate": 1590786544.605  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeRefreshSchemasStatus](#)」の「」を参照してください。

describe-replication-instances

次の例は、describe-replication-instances を使用する方法を説明しています。

AWS CLI

レプリケーションインスタンスを記述するには

次のdescribe-replication-instances例では、AWS アカウントのレプリケーションインスタンスを一覧表示します。

```
aws dms describe-replication-instances
```

出力:

```
{
  "ReplicationInstances": [
    {
      "ReplicationInstanceIdentifier": "my-repl-instance",
      "ReplicationInstanceClass": "dms.t2.micro",
      "ReplicationInstanceStatus": "available",
      "AllocatedStorage": 5,
      "InstanceCreateTime": 1590011235.952,
      "VpcSecurityGroups": [
        {
          "VpcSecurityGroupId": "sg-f839b688",
          "Status": "active"
        }
      ],
      "AvailabilityZone": "us-east-1e",
      "ReplicationSubnetGroup": {
        "ReplicationSubnetGroupIdentifier": "default",
        "ReplicationSubnetGroupDescription": "default",
        "VpcId": "vpc-136a4c6a",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
          {
            "SubnetIdentifier": "subnet-da327bf6",
            "SubnetAvailabilityZone": {
              "Name": "us-east-1a"
            },
            "SubnetStatus": "Active"
          },
          {
            "SubnetIdentifier": "subnet-42599426",
            "SubnetAvailabilityZone": {
              "Name": "us-east-1d"
            },
            "SubnetStatus": "Active"
          }
        ]
      }
    }
  ]
}
```



```
        "SubnetIdentifier": "subnet-bac383e0",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1c"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-6746046b",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-d7c825e8",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1e"
        },
        "SubnetStatus": "Active"
    },
    {
        "SubnetIdentifier": "subnet-cbfff283",
        "SubnetAvailabilityZone": {
            "Name": "us-east-1b"
        },
        "SubnetStatus": "Active"
    }
]
},
"PreferredMaintenanceWindow": "wed:11:42-wed:12:12",
"PendingModifiedValues": {
    "MultiAZ": true
},
"MultiAZ": false,
"EngineVersion": "3.3.2",
"AutoMinorVersionUpgrade": true,
"KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
f7bc0f8e-1a3a-4ace-9faa-e8494fa3921a",
"ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
"ReplicationInstancePublicIpAddress": "3.230.18.248",
"ReplicationInstancePrivateIpAddress": "172.31.75.90",
"ReplicationInstancePublicIpAddresses": [
    "3.230.18.248"
```

```
    ],
    "ReplicationInstancePrivateIpAddresses": [
      "172.31.75.90"
    ],
    "PubliclyAccessible": true,
    "FreeUntil": 1590194829.267
  }
]
}
```

詳細については、AWS「Database Migration Service ユーザーガイド」の[AWS DMS「レプリケーションインスタンスの使用」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeReplicationInstances](#)」の「」を参照してください。

describe-replication-subnet-groups

次のコード例は、describe-replication-subnet-groups を使用する方法を示しています。

AWS CLI

使用可能なサブネットグループを表示するには

次のdescribe-replication-subnet-groups例では、使用可能なサブネットグループを一覧表示します。

```
aws dms describe-replication-subnet-groups \
  --filter "Name=replication-subnet-group-id,Values=my-subnet-group"
```

出力:

```
{
  "ReplicationSubnetGroups": [
    {
      "ReplicationSubnetGroupIdentifier": "my-subnet-group",
      "ReplicationSubnetGroupDescription": "my subnet group",
      "VpcId": "vpc-136a4c6a",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-da327bf6",
          "SubnetAvailabilityZone": {
```

```
        "Name": "us-east-1a"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-bac383e0",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1c"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-d7c825e8",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1e"
      },
      "SubnetStatus": "Active"
    }
  ]
}
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の [「レプリケーションインスタンスのネットワークのセットアップ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeReplicationSubnetGroups](#)」の「」を参照してください。

describe-replication-task-assessment-results

次の例は、describe-replication-task-assessment-results を使用する方法を説明しています。

AWS CLI

レプリケーションタスク評価の結果を一覧表示するには

次のdescribe-replication-task-assessment-results例では、以前のタスク評価の結果を一覧表示します。

```
aws dms describe-replication-task-assessment-results
```

出力:

```
{
  "ReplicationTaskAssessmentResults": [
    {
      "ReplicationTaskIdentifier": "moveit2",
      "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
      "ReplicationTaskLastAssessmentDate": 1590790230.0,
      "AssessmentStatus": "No issues found",
      "AssessmentResultsFile": "moveit2/2020-05-29-22-10"
    }
  ]
}
```

詳細については、[「 Database Migration Service ユーザーガイド」の「タスク評価レポートの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReplicationTaskAssessmentResults](#)」の「」を参照してください。

describe-replication-tasks

次の例は、describe-replication-tasks を使用する方法を説明しています。

AWS CLI

レプリケーションタスクを記述するには

次のdescribe-replication-tasks例では、現在のレプリケーションタスクについて説明します。

```
aws dms describe-replication-tasks
```

出力:

```
{
  "ReplicationTasks": [
    {
      "ReplicationTaskIdentifier": "moveit2",
      "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
```

```

    "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": ...output omitted... ,
    "ReplicationTaskSettings": ...output omitted... ,
    "Status": "stopped",
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590619805.212,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII",
    "ReplicationTaskStats": {
      "FullLoadProgressPercent": 100,
      "ElapsedTimeMillis": 0,
      "TablesLoaded": 0,
      "TablesLoading": 0,
      "TablesQueued": 0,
      "TablesErrored": 0,
      "FreshStartDate": 1590619811.528,
      "StartDate": 1590619811.528,
      "StopDate": 1590619842.068
    }
  }
]
}

```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[タスクの操作 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReplicationTasks](#)」の「」を参照してください。

describe-schemas

次の例は、describe-schemas を使用する方法を説明しています。

AWS CLI

データベーススキーマを記述するには

次のdescribe-schemas例では、エンドポイントで使用可能なテーブルを一覧表示します。

```
aws dms describe-schemas \  
  --endpoint-arn "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPMWGAYUVLKIB732KEVWA"
```

出力:

```
{  
  "Schemas": [  
    "prodrep"  
  ]  
}
```

詳細については、AWS 「データベース移行サービスユーザーガイド」の「[これはトピックタイトルです](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSchemas](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、レプリケーションインスタンスのタグを一覧表示します。

```
aws dms list-tags-for-resource \  
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

出力:

```
{  
  "TagList": [  
    {  
      "Key": "Project",  
      "Value": "dbMigration"  
    },  
    {
```

```

        "Key": "Environment",
        "Value": "PROD"
    }
]
}

```

詳細については、AWS 「データベース移行サービスユーザーガイド」の [「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListTagsForResource](#)」の「」を参照してください。

modify-endpoint

次のコード例は、modify-endpoint を使用する方法を示しています。

AWS CLI

エンドポイントを変更するには

次のmodify-endpoint例では、エンドポイントに追加の接続属性を追加します。

```

aws dms modify-endpoint \
  --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U" \
  --extra-connection-attributes "compressionType=GZIP"

```

出力:

```

{
  "Endpoint": {
    "EndpointIdentifier": "src-endpoint",
    "EndpointType": "SOURCE",
    "EngineName": "s3",
    "EngineDisplayName": "Amazon S3",
    "ExtraConnectionAttributes":
"compressionType=GZIP;csvDelimiter=,;csvRowDelimiter=\n;",
    "Status": "active",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U",
    "SslMode": "none",
    "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role",
    "S3Settings": {

```

```

        "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-
role",
        "CsvRowDelimiter": "\\n",
        "CsvDelimiter": ",",
        "BucketFolder": "",
        "BucketName": "",
        "CompressionType": "GZIP",
        "EnableStatistics": true
    }
}
}

```

詳細については、AWS 「 Database Migration Service ユーザーガイド」の「エンドポイントの使用 AWS DMS」 <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`__」を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[ModifyEndpoint](#)」の「」を参照してください。

modify-event-subscription

次のコード例は、modify-event-subscription を使用する方法を示しています。

AWS CLI

イベントサブスクリプションを変更するには

次のmodify-event-subscription例では、イベントサブスクリプションのソースタイプを変更します。

```

aws dms modify-event-subscription \
  --subscription-name "my-dms-events" \
  --source-type replication-task

```

出力:

```

{
  "EventSubscription": {
    "CustomerAwsId": "123456789012",
    "CustSubscriptionId": "my-dms-events",
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:my-sns-topic",
    "Status": "modifying",
  }
}

```



```
    "SubscriptionCreationTime": "2020-05-29 17:04:40.262",
    "SourceType": "replication-task",
    "Enabled": true
  }
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[イベントと通知の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyEventSubscription](#)」の「」を参照してください。

modify-replication-instance

次のコード例は、modify-replication-instance を使用する方法を示しています。

AWS CLI

レプリケーションインスタンスを変更するには

次のmodify-replication-instance例では、マルチ AZ デプロイを使用するようにレプリケーションインスタンスを変更します。

```
aws dms modify-replication-instance \
  --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \
  --multi-az
```

出力:

```
{
  "ReplicationInstance": {
    "ReplicationInstanceIdentifier": "my-repl-instance",
    "ReplicationInstanceClass": "dms.t2.micro",
    "ReplicationInstanceStatus": "available",
    "AllocatedStorage": 5,
    "InstanceCreateTime": 1590011235.952,
    ...output omitted...
  },
  "PendingModifiedValues": {
    "MultiAZ": true
  }
}
```

```
    },
    "MultiAZ": false,
    "EngineVersion": "3.3.2",
    "AutoMinorVersionUpgrade": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/f7bc0f8e-1a3a-4ace-9faa-
e8494fa3921a",

    ...output omitted...

  }
}
```

詳細については、AWS 「 Database Migration Service ユーザーガイド」の[AWS DMS 「レプリケーションインスタンスの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[ModifyReplicationInstance](#)」の「」を参照してください。

modify-replication-subnet-group

次の例は、modify-replication-subnet-group を使用する方法を説明しています。

AWS CLI

サブネットグループを変更するには

次のmodify-replication-subnet-group例では、サブネットグループに関連付けられたサブネットのリストを変更します。

```
aws dms modify-replication-subnet-group \
  --replication-subnet-group-identifier my-subnet-group \
  --subnet-id subnet-da327bf6 subnet-bac383e0
```

出力:

```
{
  "ReplicationSubnetGroup": {
    "ReplicationSubnetGroupIdentifier": "my-subnet-group",
    "ReplicationSubnetGroupDescription": "my subnet group",
    "VpcId": "vpc-136a4c6a",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
```

```
{
  "SubnetIdentifier": "subnet-da327bf6",
  "SubnetAvailabilityZone": {
    "Name": "us-east-1a"
  },
  "SubnetStatus": "Active"
},
{
  "SubnetIdentifier": "subnet-bac383e0",
  "SubnetAvailabilityZone": {
    "Name": "us-east-1c"
  },
  "SubnetStatus": "Active"
}
]
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[レプリケーションインスタンスのネットワークのセットアップ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyReplicationSubnetGroup](#)」の「」を参照してください。

modify-replication-task

次のコード例は、modify-replication-task を使用する方法を示しています。

AWS CLI

レプリケーションタスクを変更するには

次のmodify-replication-task例では、タスクのテーブルマッピングを変更します。

```
aws dms modify-replication-task \
  --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
  --table-mappings file://table-mappings.json
```

table-mappings.json の内容:

```
{
```

```

"rules": [
  {
    "rule-type": "selection",
    "rule-id": "1",
    "rule-name": "1",
    "object-locator": {
      "schema-name": "prodrep",
      "table-name": "ACCT_%"
    },
    "rule-action": "include",
    "filters": []
  }
]
}

```

出力:

```

{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": ...output omitted...,
    "ReplicationTaskSettings": ...output omitted...,
    "Status": "modifying",
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590789424.653,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

詳細については、AWS 「Database Migration Service ユーザーガイド」の [「タスクの操作 AWS DMS」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyReplicationTask](#)」の「」を参照してください。

reboot-replication-instance

次の例は、reboot-replication-instance を使用する方法を説明しています。

AWS CLI

レプリケーションインスタンスを再起動するには

次の reboot-replication-instance の例では、レプリケーション インスタンスを再起動します。

```
aws dms reboot-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

出力:

```
{  
  "ReplicationInstance": {  
    "ReplicationInstanceIdentifier": "my-repl-instance",  
    "ReplicationInstanceClass": "dms.t2.micro",  
    "ReplicationInstanceStatus": "rebooting",  
    "AllocatedStorage": 5,  
    "InstanceCreateTime": 1590011235.952,  
    ... output omitted ...  
  }  
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の[AWS DMS 「レプリケーションインスタンスの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootReplicationInstance](#)」の「」を参照してください。

refresh-schemas

次のコード例は、refresh-schemas を使用する方法を示しています。

AWS CLI

データベーススキーマを更新するには

次のrefresh-schemas例では、エンドポイントのスキーマのリストを更新するリクエスト AWS DMSをリクエストします。

```
aws dms refresh-schemas \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  --endpoint-arn "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

出力:

```
{  
  "RefreshSchemasStatus": {  
    "EndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "Status": "refreshing",  
    "LastRefreshDate": 1590019949.103  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RefreshSchemas](#)」の「」を参照してください。

reload-tables

次の例は、reload-tables を使用する方法を説明しています。

AWS CLI

エンドポイントで使用できるテーブルのリストを更新するには

次のreload-tables例では、エンドポイントで使用可能なテーブルのリストを再ロードします。

```
aws dms reload-tables \  
  --replication-task-arn "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \  
  --tables-to-reload "SchemaName=prodrep,TableName=ACCT_BAL"
```

出力:

```
{
  "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ReloadTables](#)」の「」を参照してください。

remove-tags-from-resource

次の例は、`remove-tags-from-resource` を使用する方法を説明しています。

AWS CLI

レプリケーションインスタンスからタグを削除するには

次の `remove-tags-from-resource` 例では、レプリケーションインスタンスからタグを削除します。

```
aws dms remove-tags-from-resource \
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \
  --tag-keys Environment Project
```

このコマンドでは何も出力されません。

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveTagsFromResource](#)」の「」を参照してください。

start-replication-task-assessment

次のコード例は、`start-replication-task-assessment` を使用する方法を示しています。

AWS CLI

タスク評価を開始するには

次のstart-replication-task-assessment例では、レプリケーションタスクの評価を開始します。

```
aws dms start-replication-task-assessment \  
  --replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

出力:

```
{  
  "ReplicationTask": {  
    "ReplicationTaskIdentifier": "moveit2",  
    "SourceEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",  
    "TargetEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "MigrationType": "full-load",  
    "TableMappings": "...output omitted...",  
    "ReplicationTaskSettings": "...output omitted...",  
    "Status": "testing",  
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",  
    "ReplicationTaskCreationDate": 1590524772.505,  
    "ReplicationTaskStartDate": 1590789988.677,  
    "ReplicationTaskArn": "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"  
  }  
}
```

詳細については、[「Database Migration Service ユーザーガイド」の「タスク評価レポートの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[StartReplicationTaskAssessment](#)」の「」を参照してください。

start-replication-task

次の例は、start-replication-task を使用する方法を説明しています。

AWS CLI

レプリケーションタスクを開始するには

次のcommand-name例では、AWS アカウントで使用可能なウィジェットを一覧表示します。

```
aws dms start-replication-task \  
  --replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII \  
  --start-replication-task-type reload-target
```

出力:

```
{  
  "ReplicationTask": {  
    "ReplicationTaskIdentifier": "moveit2",  
    "SourceEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",  
    "TargetEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "MigrationType": "full-load",  
    "TableMappings": ...output omitted... ,  
    "ReplicationTaskSettings": ...output omitted... ,  
    "Status": "starting",  
    "ReplicationTaskCreationDate": 1590524772.505,  
    "ReplicationTaskStartDate": 1590619805.212,  
    "ReplicationTaskArn": "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"  
  }  
}
```

詳細については、AWS 「Database Migration Service ユーザーガイド」の「[タスクの操作 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartReplicationTask](#)」の「」を参照してください。

stop-replication-task

次のコード例は、stop-replication-task を使用する方法を示しています。

AWS CLI

タスクを停止するには

次のstop-replication-task例では、タスクを停止します。

```
aws dms stop-replication-task \  
  --replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

出力:

```
{  
  "ReplicationTask": {  
    "ReplicationTaskIdentifier": "moveit2",  
    "SourceEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",  
    "TargetEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "MigrationType": "full-load",  
    "TableMappings": ...output omitted...,  
    "ReplicationTaskSettings": ...output omitted...,  
    "Status": "stopping",  
    "ReplicationTaskCreationDate": 1590524772.505,  
    "ReplicationTaskStartDate": 1590789424.653,  
    "ReplicationTaskArn": "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"  
  }  
}
```

詳細については、AWS「Database Migration Service ユーザーガイド」の「[タスクの操作 AWS DMS](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[StopReplicationTask](#)」の「」を参照してください。

test-connection

次のコード例は、test-connection を使用する方法を示しています。

AWS CLI

エンドポイントへの接続をテストするには

次のtest-connection例では、レプリケーションインスタンスからエンドポイントにアクセスできるかどうかをテストします。

```
aws dms test-connection \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  --endpoint-arn arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

出力:

```
{  
  "Connection": {  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "EndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",  
    "Status": "testing",  
    "EndpointIdentifier": "src-database-1",  
    "ReplicationInstanceIdentifier": "my-repl-instance"  
  }  
}
```

詳細については、AWS「Database Migration Service ユーザーガイド」の[「ソースエンドポイントとターゲットエンドポイントの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TestConnection](#)」の「」を参照してください。

を使用した Amazon DocumentDB の例 AWS CLI

次のコード例は、Amazon DocumentDB AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags-to-resource

次のコード例は、add-tags-to-resource を使用する方法を示しています。

AWS CLI

指定したリソースに 1 つ以上のタグを追加するには

次のadd-tags-to-resource例では、に 3 つのタグを追加しますsample-cluster。1 つのタグ (CropB) にはキー名がありますが、値はありません。

```
aws docdb add-tags-to-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
  --tags Key="CropA",Value="Apple" Key="CropB" Key="CropC",Value="Corn"
```

このコマンドでは何も出力されません。

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB リソースのタグ付け](#)」を参照してください。 Amazon DocumentDB

- API 詳細については、AWS CLI 「[コマンドリファレンスAddTagsToResource](#)」の「」を参照してください。

apply-pending-maintenance-action

次のコード例は、apply-pending-maintenance-action を使用する方法を示しています。

AWS CLI

次のメンテナンスウィンドウで保留中のメンテナンスアクションを実行するには

次のapply-pending-maintenance-action例では、次のスケジュールされたメンテナンスウィンドウ中にすべてのシステム更新アクションを実行します。

```
aws docdb apply-pending-maintenance-action \  
--resource-identifier arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
--apply-action system-update \  
--opt-in-type next-maintenance
```

このコマンドでは何も出力されません。

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB 更新の適用](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ApplyPendingMaintenanceAction](#)」の「」を参照してください。

copy-db-cluster-parameter-group

次の例は、copy-db-cluster-parameter-group を使用する方法を説明しています。

AWS CLI

既存の DB クラスターパラメータグループを複製するには

次のcopy-db-cluster-parameter-group例では、custom-docdb3-6という名前のパラメータグループのコピーを作成しますcustom-docdb3-6-copy。コピーを行うと、新しいパラメータグループにタグが追加されます。

```
aws docdb copy-db-cluster-parameter-group \  
--source-db-cluster-parameter-group-identifier custom-docdb3-6 \  
--target-db-cluster-parameter-group-identifier custom-docdb3-6-copy \  
--target-db-cluster-parameter-group-description "Copy of custom-docdb3-6" \  
--tags Key="CopyNumber",Value="1" Key="Modifiable",Value="Yes"
```

出力:

```
{  
  "DBClusterParameterGroup": {  
    "DBParameterGroupFamily": "docdb3.6",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:12345678901:cluster-pg:custom-docdb3-6-copy",  
    "DBClusterParameterGroupName": "custom-docdb3-6-copy",  
    "Description": "Copy of custom-docdb3-6"  
  }  
}
```

```
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターパラメータグループのコピー](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyDbClusterParameterGroup](#)」の「」を参照してください。

copy-db-cluster-snapshot

次の例は、copy-db-cluster-snapshot を使用する方法を説明しています。

AWS CLI

スナップショットのコピーを作成するには

次の copy-db-cluster-snapshot の例では、sample-cluster-snapshot-copy という名前の sample-cluster-snapshot のコピーを作成します。コピーには、元のすべてのタグに加えて、キー名 の新しいタグがありますCopyNumber。

```
aws docdb copy-db-cluster-snapshot \  
  --source-db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \  
  --copy-tags \  
  --tags Key="CopyNumber",Value="1"
```

このコマンドでは何も出力されません。

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[クラスタースナップショットのコピー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyDbClusterSnapshot](#)」の「」を参照してください。

create-db-cluster-parameter-group

次のコード例は、create-db-cluster-parameter-group を使用する方法を示しています。

AWS CLI

Amazon DocumentDB クラスターパラメータグループを作成するには

次のcreate-db-cluster-parameter-group例では、docdb3.6ファミリーsample-parameter-groupを使用して DB クラスターパラメータグループを作成します。

```
aws docdb create-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group \  
  --db-parameter-group-family docdb3.6 \  
  --description "Sample parameter group based on docdb3.6"
```

出力:

```
{  
  "DBClusterParameterGroup": {  
    "Description": "Sample parameter group based on docdb3.6",  
    "DBParameterGroupFamily": "docdb3.6",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-west-2:123456789012:cluster-pg:sample-parameter-group",  
    "DBClusterParameterGroupName": "sample-parameter-group"  
  }  
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターパラメータグループの作成](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbClusterParameterGroup](#)」の「」を参照してください。

create-db-cluster-snapshot

次の例は、create-db-cluster-snapshot を使用する方法を説明しています。

AWS CLI

手動 Amazon DocumentDB クラスタースナップショットを作成するには

次のcreate-db-cluster-snapshot例では、という名前の Amazon DB クラスタースナップショットを作成します sample-cluster-snapshot。

```
aws docdb create-db-cluster-snapshot \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

出力:

```
{
  "DBClusterSnapshot": {
    "MasterUsername": "master-user",
    "SnapshotCreateTime": "2019-03-18T18:27:14.794Z",
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2b",
      "us-west-2c",
      "us-west-2d",
      "us-west-2e",
      "us-west-2f"
    ],
    "SnapshotType": "manual",
    "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
    "EngineVersion": "3.6.0",
    "PercentProgress": 0,
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
    "Engine": "docdb",
    "DBClusterIdentifier": "sample-cluster",
    "Status": "creating",
    "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
    "Port": 0,
    "StorageEncrypted": false,
    "VpcId": "vpc-91280df6"
  }
}
```

詳細については、[「Amazon DocumentDB デベロッパーガイド」の「手動クラスタースナップショットの作成」](#)を参照してください。 Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbClusterSnapshot](#)」の「」を参照してください。

create-db-cluster

次のコード例は、create-db-cluster を使用する方法を示しています。

AWS CLI

Amazon DocumentDB クラスターを作成するには

次のcreate-db-cluster例では、日曜日の 20:30 から 11:00 の間に、優先メンテナンスウィンドウsample-clusterを使用して という名前の Amazon DocumentDB クラスターを作成します。

```
aws docdb create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --engine docdb \  
  --master-username master-user \  
  --master-user-password password \  
  --preferred-maintenance-window Sun:20:30-Sun:21:00
```

出力:

```
{  
  "DBCluster": {  
    "DBClusterParameterGroup": "default.docdb3.6",  
    "AssociatedRoles": [],  
    "DBSubnetGroup": "default",  
    "ClusterCreateTime": "2019-03-18T18:06:34.616Z",  
    "Status": "creating",  
    "Port": 27017,  
    "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",  
    "HostedZoneId": "ZNKXH85TT8WVW",  
    "DBClusterMembers": [],  
    "Engine": "docdb",  
    "DBClusterIdentifier": "sample-cluster",  
    "PreferredBackupWindow": "10:12-10:42",  
    "AvailabilityZones": [  
      "us-west-2d",  
      "us-west-2f",  
      "us-west-2e"  
    ],  
    "MasterUsername": "master-user",  
    "BackupRetentionPeriod": 1,  
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-west-2.docdb.amazonaws.com",  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-77186e0d",  
        "Status": "active"  
      }  
    ],  
    "StorageEncrypted": false,  
  }  
}
```

```
"DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
"DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
"MultiAZ": false,
"Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
"EngineVersion": "3.6.0"
}
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターの作成](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbCluster](#)」の「」を参照してください。

create-db-instance

次のコード例は、create-db-instance を使用する方法を示しています。

AWS CLI

Amazon DocumentDB クラスターインスタンスを作成するには

次のコードcreate-db-instance例では、Amazon DocumentDB クラスター sample-cluster-instance-2にインスタンスを作成しますsample-cluster。

```
aws docdb create-db-instance \
  --db-cluster-identifier sample-cluster \
  --db-instance-class db.r4.xlarge \
  --db-instance-identifier sample-cluster-instance-2 \
  --engine docdb
```

出力:

```
{
  "DBInstance": {
    "DBInstanceStatus": "creating",
    "PendingModifiedValues": {
      "PendingCloudwatchLogsExports": {
        "LogTypesToEnable": [
          "audit"
        ]
      }
    }
  }
}
```

```
},
"PubliclyAccessible": false,
"PreferredBackupWindow": "00:00-00:30",
"PromotionTier": 1,
"EngineVersion": "3.6.0",
"BackupRetentionPeriod": 3,
"DBInstanceIdentifier": "sample-cluster-instance-2",
"PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
"StorageEncrypted": false,
"Engine": "docdb",
"DBClusterIdentifier": "sample-cluster",
"DBSubnetGroup": {
  "Subnets": [
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-4e26d263"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2c"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-afc329f4"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-53ab3636"
    },
    {
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      },
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-991cb8d0"
    }
  ],
  "DBSubnetGroupDescription": "default",
  "SubnetGroupStatus": "Complete",
```

```
    "VpcId": "vpc-91280df6",
    "DBSubnetGroupName": "default"
  },
  "DBInstanceClass": "db.r4.xlarge",
  "VpcSecurityGroups": [
    {
      "Status": "active",
      "VpcSecurityGroupId": "sg-77186e0d"
    }
  ],
  "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
  "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4H03ITE"
}
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[クラスターへの Amazon DocumentDB インスタンスの追加](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbInstance](#)」の「」を参照してください。

create-db-subnet-group

次の例は、create-db-subnet-group を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB サブネットグループを作成するには

次のcreate-db-subnet-group例では、という名前の Amazon DocumentDB サブネットグループを作成しますsample-subnet-group。

```
aws docdb create-db-subnet-group \
  --db-subnet-group-description "a sample subnet group" \
  --db-subnet-group-name sample-subnet-group \
  --subnet-ids "subnet-29ab1025" "subnet-991cb8d0" "subnet-53ab3636"
```

出力:

```
{
```

```
"DBSubnetGroup": {
  "SubnetGroupStatus": "Complete",
  "DBSubnetGroupName": "sample-subnet-group",
  "DBSubnetGroupDescription": "a sample subnet group",
  "VpcId": "vpc-91280df6",
  "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
  "Subnets": [
    {
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-53ab3636",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      }
    },
    {
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-991cb8d0",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      }
    },
    {
      "SubnetStatus": "Active",
      "SubnetIdentifier": "subnet-29ab1025",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2c"
      }
    }
  ]
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB サブネットグループの作成](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbSubnetGroup](#)」の「」を参照してください。

delete-db-cluster-parameter-group

次のコード例は、delete-db-cluster-parameter-group を使用する方法を示しています。

AWS CLI

Amazon DocumentDB クラスターパラメータグループを削除するには

次のdelete-db-cluster-parameter-group例では、Amazon DocumentDB パラメータグループを削除しますsample-parameter-group。

```
aws docdb delete-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group
```

このコマンドでは何も出力されません。

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターパラメータグループの削除](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbClusterParameterGroup](#)」の「」を参照してください。

delete-db-cluster-snapshot

次の例は、delete-db-cluster-snapshot を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB クラスタースナップショットを削除するには

次のdelete-db-cluster-snapshot例では、Amazon DocumentDB クラスタースナップショットを削除しますsample-cluster-snapshot。

```
aws docdb delete-db-cluster-snapshot \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

出力:

```
{  
  "DBClusterSnapshot": {  
    "DBClusterIdentifier": "sample-cluster",  
    "AvailabilityZones": [  
      "us-west-2a",  
      "us-west-2b",  
      "us-west-2c",  
    ]  
  }  
}
```

```
        "us-west-2d"
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
    "VpcId": "vpc-91280df6",
    "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-
snapshot:sample-cluster-snapshot",
    "EngineVersion": "3.6.0",
    "Engine": "docdb",
    "SnapshotCreateTime": "2019-03-18T18:27:14.794Z",
    "Status": "available",
    "MasterUsername": "master-user",
    "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
    "PercentProgress": 100,
    "StorageEncrypted": false,
    "SnapshotType": "manual",
    "Port": 0
}
}
```

詳細については、Amazon DocumentDB デベロッパーガイド [「クラスタースナップショットの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDbClusterSnapshot](#)」の「」を参照してください。

delete-db-cluster

次のコード例は、delete-db-cluster を使用する方法を示しています。

AWS CLI

Amazon DocumentDB クラスタを削除するには

次のdelete-db-cluster例では、Amazon DocumentDB クラスタ を削除しますsample-cluster。削除する前にクラスタのバックアップは行われません。NOTE: クラスタに関連付けられているすべてのインスタンスを削除する前に削除する必要があります。

```
aws docdb delete-db-cluster \
  --db-cluster-identifier sample-cluster \
  --skip-final-snapshot
```

出力:

```
{
  "DBCluster": {
    "DBClusterIdentifier": "sample-cluster",
    "DBSubnetGroup": "default",
    "EngineVersion": "3.6.0",
    "Engine": "docdb",
    "LatestRestorableTime": "2019-03-18T18:07:24.610Z",
    "PreferredMaintenanceWindow": "sun:20:30-sun:21:00",
    "StorageEncrypted": false,
    "EarliestRestorableTime": "2019-03-18T18:07:24.610Z",
    "Port": 27017,
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ],
    "MultiAZ": false,
    "MasterUsername": "master-user",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
    "Status": "available",
    "PreferredBackupWindow": "10:12-10:42",
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "AvailabilityZones": [
      "us-west-2c",
      "us-west-2b",
      "us-west-2a"
    ],
    "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "DbClusterResourceId": "cluster-L3R4YRSBUYDP4GLMTJ2WF5GH5Q",
    "ClusterCreateTime": "2019-03-18T18:06:34.616Z",
    "AssociatedRoles": [],
    "DBClusterParameterGroup": "default.docdb3.6",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "BackupRetentionPeriod": 1,
    "DBClusterMembers": []
  }
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターの削除](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbCluster](#)」の「」を参照してください。

delete-db-instance

次の例は、delete-db-instance を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB インスタンスを削除するには

次のdelete-db-instance例では、Amazon DocumentDB インスタンス を削除しますsample-cluster-instance-2。

```
aws docdb delete-db-instance \  
--db-instance-identifier sample-cluster-instance-2
```

出力:

```
{  
  "DBInstance": {  
    "DBSubnetGroup": {  
      "Subnets": [  
        {  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2a"  
          },  
          "SubnetStatus": "Active",  
          "SubnetIdentifier": "subnet-4e26d263"  
        },  
        {  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2c"  
          },  
          "SubnetStatus": "Active",  
          "SubnetIdentifier": "subnet-afc329f4"  
        },  
        {  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2d"  
          },  
          "SubnetStatus": "Active",
```

```
        "SubnetIdentifier": "subnet-53ab3636"
      },
      {
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        },
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-991cb8d0"
      }
    ],
    "DBSubnetGroupName": "default",
    "DBSubnetGroupDescription": "default",
    "VpcId": "vpc-91280df6",
    "SubnetGroupStatus": "Complete"
  },
  "PreferredBackupWindow": "00:00-00:30",
  "InstanceCreateTime": "2019-03-18T18:37:33.709Z",
  "DBInstanceClass": "db.r4.xlarge",
  "DbiResourceId": "db-XEKJLEMGRV5ZKCARUVA4H03ITE",
  "BackupRetentionPeriod": 3,
  "Engine": "docdb",
  "VpcSecurityGroups": [
    {
      "Status": "active",
      "VpcSecurityGroupId": "sg-77186e0d"
    }
  ],
  "AutoMinorVersionUpgrade": true,
  "PromotionTier": 1,
  "EngineVersion": "3.6.0",
  "Endpoint": {
    "Address": "sample-cluster-instance-2.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "Port": 27017
  },
  "DBInstanceIdentifier": "sample-cluster-instance-2",
  "PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
  "EnabledCloudwatchLogsExports": [
    "audit"
  ],
  "PendingModifiedValues": {},
  "DBInstanceStatus": "deleting",
  "PubliclyAccessible": false,
```

```
    "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
    "DBClusterIdentifier": "sample-cluster",
    "AvailabilityZone": "us-west-2c",
    "StorageEncrypted": false
  }
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB インスタンスの削除](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbInstance](#)」の「」を参照してください。

delete-db-subnet-group

次のコード例は、delete-db-subnet-group を使用する方法を示しています。

AWS CLI

Amazon DocumentDB サブネットグループを削除するには

次のdelete-db-subnet-group例では、Amazon DocumentDB サブネットグループを削除しますsample-subnet-group。

```
aws docdb delete-db-subnet-group \
  --db-subnet-group-name sample-subnet-group
```

このコマンドでは何も出力されません。

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB サブネットグループの削除](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbSubnetGroup](#)」の「」を参照してください。

describe-db-cluster-parameter-groups

次の例は、describe-db-cluster-parameter-groups を使用する方法を説明しています。

AWS CLI

1 つ以上の Amazon DocumentDB クラスターパラメータグループの詳細を表示するには

次のdescribe-db-cluster-parameter-groups例では、Amazon DocumentDB クラスターパラメータグループの詳細を表示しますcustom3-6-param-grp。

```
aws docdb describe-db-cluster-parameter-groups \  
  --db-cluster-parameter-group-name custom3-6-param-grp
```

出力:

```
{  
  "DBClusterParameterGroups": [  
    {  
      "DBParameterGroupFamily": "docdb3.6",  
      "DBClusterParameterGroupArn": "arn:aws:rds:us-  
east-1:123456789012:cluster-pg:custom3-6-param-grp",  
      "Description": "Custom docdb3.6 parameter group",  
      "DBClusterParameterGroupName": "custom3-6-param-grp"  
    }  
  ]  
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターパラメータグループの表示](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeDbClusterParameterGroups](#)」の「」を参照してください。

describe-db-cluster-parameters

次のコード例は、describe-db-cluster-parameters を使用する方法を示しています。

AWS CLI

Amazon DocumentDB クラスターパラメータグループの詳細なパラメータリストを表示するには。

次のdescribe-db-cluster-parameters例では、Amazon DocumentDB パラメータグループcustom3-6-param-grpのパラメータを一覧表示します。

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name custom3-6-param-grp
```

出力:

```
{  
  "Parameters": [  
    {  
      "DataType": "string",  
      "ParameterName": "audit_logs",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot",  
      "Source": "system",  
      "ApplyType": "dynamic",  
      "AllowedValues": "enabled,disabled",  
      "Description": "Enables auditing on cluster.",  
      "ParameterValue": "disabled"  
    },  
    {  
      "DataType": "string",  
      "ParameterName": "tls",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot",  
      "Source": "system",  
      "ApplyType": "static",  
      "AllowedValues": "disabled,enabled",  
      "Description": "Config to enable/disable TLS",  
      "ParameterValue": "enabled"  
    },  
    {  
      "DataType": "string",  
      "ParameterName": "ttl_monitor",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot",  
      "Source": "user",  
      "ApplyType": "dynamic",  
      "AllowedValues": "disabled,enabled",  
      "Description": "Enables TTL Monitoring",  
      "ParameterValue": "enabled"  
    }  
  ]  
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターパラメータの表示](#)」を参照してください。Amazon DocumentDB

- API 詳細については、「コマンドリファレンス[DescribeDbClusterParameters](#)」の「」を参照してください。AWS CLI

describe-db-cluster-snapshot-attributes

次の例は、describe-db-cluster-snapshot-attributes を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB スナップショットの属性名と値を一覧表示するには

次のdescribe-db-cluster-snapshot-attributes例では、Amazon DocumentDB スナップショットの属性名と値を一覧表示しますsample-cluster-snapshot。

```
aws docdb describe-db-cluster-snapshot-attributes \  
--db-cluster-snapshot-identifier sample-cluster-snapshot
```

出力:

```
{  
  "DBClusterSnapshotAttributesResult": {  
    "DBClusterSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": []  
      }  
    ],  
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"  
  }  
}
```

詳細については、「[Amazon DocumentDB デベロッパーガイド](#)」の「[DescribeDBClusterSnapshotAttributes](#)」を参照してください。Amazon DocumentDB

- API 詳細については、「コマンドリファレンス[DescribeDbClusterSnapshotAttributes](#)」の「」を参照してください。AWS CLI

describe-db-cluster-snapshots

次のコード例は、describe-db-cluster-snapshots を使用する方法を示しています。

AWS CLI

Amazon DocumentDB スナップショットを記述するには

次のdescribe-db-cluster-snapshots例では、Amazon DocumentDB スナップショットの詳細を表示しますsample-cluster-snapshot。

```
aws docdb describe-db-cluster-snapshots \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

出力:

```
{  
  "DBClusterSnapshots": [  
    {  
      "AvailabilityZones": [  
        "us-west-2a",  
        "us-west-2b",  
        "us-west-2c",  
        "us-west-2d"  
      ],  
      "Status": "available",  
      "DBClusterSnapshotArn": "arn:aws:rds:us-west-2:123456789012:cluster-snapshot:sample-cluster-snapshot",  
      "SnapshotCreateTime": "2019-03-15T20:41:26.515Z",  
      "SnapshotType": "manual",  
      "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",  
      "DBClusterIdentifier": "sample-cluster",  
      "MasterUsername": "master-user",  
      "StorageEncrypted": false,  
      "VpcId": "vpc-91280df6",  
      "EngineVersion": "3.6.0",  
      "PercentProgress": 100,  
      "Port": 0,  
      "Engine": "docdb",  
      "ClusterCreateTime": "2019-03-15T20:29:58.836Z"  
    }  
  ]  
}
```

詳細については、Amazon DocumentDB デベロッパーガイド」の「[DescribeDBClusterスナップショット](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbClusterSnapshots](#)」の「」を参照してください。

describe-db-clusters

次の例は、describe-db-clusters を使用する方法を説明しています。

AWS CLI

1 つ以上の Amazon DocumentDB クラスターに関する詳細情報を取得するには。

次のdescribe-db-clusters例では、Amazon DocumentDB クラスターの詳細を表示しますsample-cluster。--db-cluster-identifier パラメータを省略すると、最大 100 個のクラスターの情報を取得できます。

```
aws docdb describe-db-clusters
--db-cluster-identifier sample-cluster
```

出力:

```
{
  "DBClusters": [
    {
      "DBClusterParameterGroup": "default.docdb3.6",
      "Endpoint": "sample-cluster.cluster-corcjzrlsfc.us-west-2.docdb.amazonaws.com",
      "PreferredBackupWindow": "00:00-00:30",
      "DBClusterIdentifier": "sample-cluster",
      "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
      "LatestRestorableTime": "2019-03-18T20:28:03.239Z",
      "MasterUsername": "master-user",
      "DBClusterMembers": [
        {
          "PromotionTier": 1,
          "DBClusterParameterGroupStatus": "in-sync",
          "IsClusterWriter": false,
          "DBInstanceIdentifier": "sample-cluster"
        }
      ]
    }
  ]
}
```



```
        "PromotionTier": 1,
        "DBClusterParameterGroupStatus": "in-sync",
        "IsClusterWriter": true,
        "DBInstanceIdentifier": "sample-cluster2"
      }
    ],
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-77186e0d",
        "Status": "active"
      }
    ],
    "Engine": "docdb",
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "DBSubnetGroup": "default",
    "MultiAZ": true,
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2c",
      "us-west-2b"
    ],
    "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
    "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
    "BackupRetentionPeriod": 3,
    "HostedZoneId": "ZNKXH85TT8VWV",
    "StorageEncrypted": false,
    "EnabledCloudwatchLogsExports": [
      "audit"
    ],
    "AssociatedRoles": [],
    "EngineVersion": "3.6.0",
    "Port": 27017,
    "Status": "available"
  }
]
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターの説明](#)」を参照してください。 Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbClusters](#)」の「」を参照してください。

describe-db-engine-versions

次の例は、describe-db-engine-versions を使用する方法を説明しています。

AWS CLI

使用可能な Amazon DocumentDB エンジンバージョンを一覧表示するには

次のdescribe-db-engine-versions例では、使用可能なすべての Amazon DocumentDB エンジンバージョンを一覧表示します。

```
aws docdb describe-db-engine-versions \  
  --engine docdb
```

出力:

```
{  
  "DBEngineVersions": [  
    {  
      "DBEngineVersionDescription": "DocDB version 1.0.200837",  
      "DBParameterGroupFamily": "docdb3.6",  
      "EngineVersion": "3.6.0",  
      "ValidUpgradeTarget": [],  
      "DBEngineDescription": "Amazon DocumentDB (with MongoDB compatibility)",  
      "SupportsLogExportsToCloudwatchLogs": true,  
      "Engine": "docdb",  
      "ExportableLogTypes": [  
        "audit"  
      ]  
    }  
  ]  
}
```

詳細については、Amazon DocumentDB デベロッパーガイド」の「[DescribeDBEngineバージョン](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbEngineVersions](#)」の「」を参照してください。

describe-db-instances

次の例は、describe-db-instances を使用する方法を説明しています。

AWS CLI

プロビジョニングされた Amazon DocumentDB インスタンスに関する情報を検索するには

次のdescribe-db-instances例では、Amazon DocumentDB インスタンスに関する詳細を表示しますsample-cluster-instance。--db-instance-identifier パラメータを省略すると、最大 100 個のインスタンスに関する情報が得られます。

```
aws docdb describe-db-instances \  
  --db-instance-identifier sample-cluster-instance
```

出力:

```
{  
  "DBInstances": [  
    {  
      "Endpoint": {  
        "HostedZoneId": "ZNKXH85TT8WVW",  
        "Address": "sample-cluster-instance.corcjozrlsfc.us-west-2.docdb.amazonaws.com",  
        "Port": 27017  
      },  
      "PreferredBackupWindow": "00:00-00:30",  
      "DBInstanceStatus": "available",  
      "DBInstanceClass": "db.r4.large",  
      "EnabledCloudwatchLogsExports": [  
        "audit"  
      ],  
      "DBInstanceIdentifier": "sample-cluster-instance",  
      "DBSubnetGroup": {  
        "Subnets": [  
          {  
            "SubnetStatus": "Active",  
            "SubnetIdentifier": "subnet-4e26d263",  
            "SubnetAvailabilityZone": {  
              "Name": "us-west-2a"  
            }  
          }  
        ],  
      }  
    }  
  ]  
}
```

```

        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-afc329f4",
        "SubnetAvailabilityZone": {
            "Name": "us-west-2c"
        }
    },
    {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-53ab3636",
        "SubnetAvailabilityZone": {
            "Name": "us-west-2d"
        }
    },
    {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-991cb8d0",
        "SubnetAvailabilityZone": {
            "Name": "us-west-2b"
        }
    }
],
"DBSubnetGroupName": "default",
"SubnetGroupStatus": "Complete",
"DBSubnetGroupDescription": "default",
"VpcId": "vpc-91280df6"
},
"InstanceCreateTime": "2019-03-15T20:36:06.338Z",
"Engine": "docdb",
"StorageEncrypted": false,
"AutoMinorVersionUpgrade": true,
"DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance",
"PreferredMaintenanceWindow": "tue:08:39-tue:09:09",
"VpcSecurityGroups": [
    {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
    }
],
"DBClusterIdentifier": "sample-cluster",
"PendingModifiedValues": {},
"BackupRetentionPeriod": 3,
"PubliclyAccessible": false,
"EngineVersion": "3.6.0",

```

```
        "PromotionTier": 1,  
        "AvailabilityZone": "us-west-2c",  
        "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA"  
    }  
]  
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB インスタンスの説明](#)」を参照してください。 Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbInstances](#)」の「」を参照してください。

describe-db-subnet-groups

次のコード例は、describe-db-subnet-groups を使用する方法を示しています。

AWS CLI

Amazon DocumentDB サブネットの説明のリストを取得するには

次のdescribe-db-subnet-groups例では、 という名前の Amazon DocumentDB サブネットの詳細について説明しますdefault。

```
aws docdb describe-db-subnet-groups \  
  --db-subnet-group-name default
```

出力:

```
{  
  "DBSubnetGroups": [  
    {  
      "VpcId": "vpc-91280df6",  
      "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:default",  
      "Subnets": [  
        {  
          "SubnetIdentifier": "subnet-4e26d263",  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2a"  
          }  
        }  
      ],  
    }  
  ],  
}
```

```
    {
      "SubnetIdentifier": "subnet-afc329f4",
      "SubnetStatus": "Active",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2c"
      }
    },
    {
      "SubnetIdentifier": "subnet-53ab3636",
      "SubnetStatus": "Active",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      }
    },
    {
      "SubnetIdentifier": "subnet-991cb8d0",
      "SubnetStatus": "Active",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      }
    }
  ],
  "DBSubnetGroupName": "default",
  "SubnetGroupStatus": "Complete",
  "DBSubnetGroupDescription": "default"
}
]
```

詳細については、Amazon DocumentDB [デベロッパーガイド](#)の「[サブネットグループの説明](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeDbSubnetGroups](#)」の「」を参照してください。

describe-engine-default-cluster-parameters

次の例は、describe-engine-default-cluster-parameters を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB のデフォルトのエンジンおよびシステムパラメータ情報を記述するには

次のdescribe-engine-default-cluster-parameters例では、Amazon DocumentDB パラメータグループのデフォルトのエンジンおよびシステムパラメータ情報の詳細を表示しますdocdb3.6。

```
aws docdb describe-engine-default-cluster-parameters \  
  --db-parameter-group-family docdb3.6
```

出力:

```
{  
  "EngineDefaults": {  
    "DBParameterGroupFamily": "docdb3.6",  
    "Parameters": [  
      {  
        "ApplyType": "dynamic",  
        "ParameterValue": "disabled",  
        "Description": "Enables auditing on cluster.",  
        "Source": "system",  
        "DataType": "string",  
        "MinimumEngineVersion": "3.6.0",  
        "AllowedValues": "enabled,disabled",  
        "ParameterName": "audit_logs",  
        "IsModifiable": true  
      },  
      {  
        "ApplyType": "static",  
        "ParameterValue": "enabled",  
        "Description": "Config to enable/disable TLS",  
        "Source": "system",  
        "DataType": "string",  
        "MinimumEngineVersion": "3.6.0",  
        "AllowedValues": "disabled,enabled",  
        "ParameterName": "tls",  
        "IsModifiable": true  
      },  
      {  
        "ApplyType": "dynamic",  
        "ParameterValue": "enabled",  
        "Description": "Enables TTL Monitoring",  
        "Source": "system",  
        "DataType": "string",  
        "MinimumEngineVersion": "3.6.0",  
        "AllowedValues": "disabled,enabled",
```

```
        "ParameterName": "ttl_monitor",
        "IsModifiable": true
    }
]
}
```

詳細については、Amazon DocumentDB デベロッパーガイド [「DescribeEngineDefaultClusterParameters」](#) の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEngineDefaultClusterParameters」](#) の「」を参照してください。

describe-event-categories

次のコード例は、describe-event-categories を使用する方法を示しています。

AWS CLI

すべての Amazon DocumentDB イベントカテゴリを記述するには

次のdescribe-event-categories例では、Amazon DocumentDB イベントソースタイプ のすべてのカテゴリを一覧表示しますdb-instance。

```
aws docdb describe-event-categories \
  --source-type db-cluster
```

出力:

```
{
  "EventCategoriesMapList": [
    {
      "SourceType": "db-cluster",
      "EventCategories": [
        "failover",
        "maintenance",
        "notification",
        "failure"
      ]
    }
  ]
}
```


詳細については、Amazon DocumentDB デベロッパーガイド」の「[イベントカテゴリの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEventCategories](#)」の「」を参照してください。

describe-events

次の例は、describe-events を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB イベントを一覧表示するには

次のdescribe-events例では、過去 24 時間 (1440 分) のすべての Amazon DocumentDB イベントを一覧表示します。

```
aws docdb describe-events \  
  --duration 1440
```

このコマンドでは何も出力されません。出力:

```
{  
  "Events": [  
    {  
      "EventCategories": [  
        "failover"  
      ],  
      "Message": "Started cross AZ failover to DB instance: sample-cluster",  
      "Date": "2019-03-18T21:36:29.807Z",  
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-  
cluster",  
      "SourceIdentifier": "sample-cluster",  
      "SourceType": "db-cluster"  
    },  
    {  
      "EventCategories": [  
        "availability"  
      ],  
      "Message": "DB instance restarted",  
      "Date": "2019-03-18T21:36:40.793Z",  
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster",  
      "SourceIdentifier": "sample-cluster",
```

```

        "SourceType": "db-instance"
    },
    {
        "EventCategories": [],
        "Message": "A new writer was promoted. Restarting database as a
reader.",
        "Date": "2019-03-18T21:36:43.873Z",
        "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "SourceIdentifier": "sample-cluster2",
        "SourceType": "db-instance"
    },
    {
        "EventCategories": [
            "availability"
        ],
        "Message": "DB instance restarted",
        "Date": "2019-03-18T21:36:51.257Z",
        "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "SourceIdentifier": "sample-cluster2",
        "SourceType": "db-instance"
    },
    {
        "EventCategories": [
            "failover"
        ],
        "Message": "Completed failover to DB instance: sample-cluster",
        "Date": "2019-03-18T21:36:53.462Z",
        "SourceArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
        "SourceIdentifier": "sample-cluster",
        "SourceType": "db-cluster"
    },
    {
        "Date": "2019-03-19T16:51:48.847Z",
        "EventCategories": [
            "configuration change"
        ],
        "Message": "Updated parameter audit_logs to enabled with apply method
pending-reboot",
        "SourceIdentifier": "custom3-6-param-grp",
        "SourceType": "db-parameter-group"
    },
    {
        "EventCategories": [

```

```
        "configuration change"
    ],
    "Message": "Applying modification to database instance class",
    "Date": "2019-03-19T17:55:20.095Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
},
{
    "EventCategories": [
        "availability"
    ],
    "Message": "DB instance shutdown",
    "Date": "2019-03-19T17:56:31.127Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
},
{
    "EventCategories": [
        "configuration change"
    ],
    "Message": "Finished applying modification to DB instance class",
    "Date": "2019-03-19T18:00:45.822Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
},
{
    "EventCategories": [
        "availability"
    ],
    "Message": "DB instance restarted",
    "Date": "2019-03-19T18:00:53.397Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
},
{
    "EventCategories": [
        "availability"
    ],
    "Message": "DB instance shutdown",
    "Date": "2019-03-19T18:23:36.045Z",
```

```

    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
  },
  {
    "EventCategories": [
      "availability"
    ],
    "Message": "DB instance restarted",
    "Date": "2019-03-19T18:23:46.209Z",
    "SourceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
    "SourceIdentifier": "sample-cluster2",
    "SourceType": "db-instance"
  },
  {
    "Date": "2019-03-19T18:39:05.822Z",
    "EventCategories": [
      "configuration change"
    ],
    "Message": "Updated parameter ttl_monitor to enabled with apply method
immediate",
    "SourceIdentifier": "custom3-6-param-grp",
    "SourceType": "db-parameter-group"
  },
  {
    "Date": "2019-03-19T18:39:48.067Z",
    "EventCategories": [
      "configuration change"
    ],
    "Message": "Updated parameter audit_logs to disabled with apply method
immediate",
    "SourceIdentifier": "custom3-6-param-grp",
    "SourceType": "db-parameter-group"
  }
]
}

```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB イベントの表示](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeEvents](#)」の「」を参照してください。

describe-orderable-db-instance-options

次のコード例は、describe-orderable-db-instance-options を使用する方法を示しています。

AWS CLI

Amazon DocumentDB インスタンスオプションを検索するには、

次のdescribe-orderable-db-instance-options例では、リージョンの Amazon DocumentDB のすべてのインスタンスオプションを一覧表示します。

```
aws docdb describe-orderable-db-instance-options \  
  --engine docdb \  
  --region us-east-1
```

出力:

```
{  
  "OrderableDBInstanceOptions": [  
    {  
      "Vpc": true,  
      "AvailabilityZones": [  
        {  
          "Name": "us-east-1a"  
        },  
        {  
          "Name": "us-east-1b"  
        },  
        {  
          "Name": "us-east-1c"  
        },  
        {  
          "Name": "us-east-1d"  
        }  
      ],  
      "EngineVersion": "3.6.0",  
      "DBInstanceClass": "db.r4.16xlarge",  
      "LicenseModel": "na",  
      "Engine": "docdb"  
    },  
    {  
      "Vpc": true,
```

```
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.2xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.4xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
```

```
        {
            "Name": "us-east-1a"
        },
        {
            "Name": "us-east-1b"
        },
        {
            "Name": "us-east-1c"
        },
        {
            "Name": "us-east-1d"
        }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.8xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
},
{
    "Vpc": true,
    "AvailabilityZones": [
        {
            "Name": "us-east-1a"
        },
        {
            "Name": "us-east-1b"
        },
        {
            "Name": "us-east-1c"
        },
        {
            "Name": "us-east-1d"
        }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.large",
    "LicenseModel": "na",
    "Engine": "docdb"
},
{
    "Vpc": true,
    "AvailabilityZones": [
        {
            "Name": "us-east-1a"
```

```
    },
    {
      "Name": "us-east-1b"
    },
    {
      "Name": "us-east-1c"
    },
    {
      "Name": "us-east-1d"
    }
  ],
  "EngineVersion": "3.6.0",
  "DBInstanceClass": "db.r4.xlarge",
  "LicenseModel": "na",
  "Engine": "docdb"
}
]
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[クラスターへの Amazon DocumentDB インスタンスの追加](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeOrderableDbInstanceOptions](#)」の「」を参照してください。

describe-pending-maintenance-actions

次のコード例は、describe-pending-maintenance-actions を使用する方法を示しています。

AWS CLI

保留中の Amazon DocumentDB メンテナンスアクションを一覧表示するには

次のdescribe-pending-maintenance-actions例では、保留中のすべての Amazon DocumentDB メンテナンスアクションを一覧表示します。

```
aws docdb describe-pending-maintenance-actions
```

出力:

```
{
  "PendingMaintenanceActions": []
}
```



```
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB の維持](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribePendingMaintenanceActions](#)」の「」を参照してください。

failover-db-cluster

次の例は、failover-db-cluster を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB クラスターをレプリカにフェイルオーバーさせるには

次のfailover-db-cluster例では、Amazon DocumentDB クラスターのサンプルクラスター内のプライマリインスタンスがレプリカにフェイルオーバーします。

```
aws docdb failover-db-cluster \  
  --db-cluster-identifier sample-cluster
```

出力:

```
{  
  "DBCluster": {  
    "AssociatedRoles": [],  
    "DBClusterIdentifier": "sample-cluster",  
    "EngineVersion": "3.6.0",  
    "DBSubnetGroup": "default",  
    "MasterUsername": "master-user",  
    "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",  
    "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-west-2.docdb.amazonaws.com",  
    "AvailabilityZones": [  
      "us-west-2a",  
      "us-west-2c",  
      "us-west-2b"  
    ],  
    "LatestRestorableTime": "2019-03-18T21:35:23.548Z",  
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",  
    "PreferredBackupWindow": "00:00-00:30",
```

```
"Port": 27017,
  "VpcSecurityGroups": [
    {
      "VpcSecurityGroupId": "sg-77186e0d",
      "Status": "active"
    }
  ],
  "StorageEncrypted": false,
  "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
  "MultiAZ": true,
  "Status": "available",
  "DBClusterMembers": [
    {
      "DBClusterParameterGroupStatus": "in-sync",
      "IsClusterWriter": false,
      "DBInstanceIdentifier": "sample-cluster",
      "PromotionTier": 1
    },
    {
      "DBClusterParameterGroupStatus": "in-sync",
      "IsClusterWriter": true,
      "DBInstanceIdentifier": "sample-cluster2",
      "PromotionTier": 2
    }
  ],
  "EnabledCloudwatchLogsExports": [
    "audit"
  ],
  "DBClusterParameterGroup": "default.docdb3.6",
  "HostedZoneId": "ZNKXH85TT8WVW",
  "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
  "BackupRetentionPeriod": 3,
  "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
  "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
  "Engine": "docdb"
}
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB フェイルオーバー](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[FailoverDbCluster](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB リソースのすべてのタグを一覧表示するには

次のlist-tags-for-resource例では、Amazon DocumentDB クラスター のすべてのタグを一覧表示しますsample-cluster。

```
aws docdb list-tags-for-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster
```

出力:

```
{  
  "TagList": [  
    {  
      "Key": "A",  
      "Value": "ALPHA"  
    },  
    {  
      "Key": "B",  
      "Value": ""  
    },  
    {  
      "Key": "C",  
      "Value": "CHARLIE"  
    }  
  ]  
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB リソースのタグの一覧表示](#)」を参照してください。 Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

modify-db-cluster-parameter-group

次の例は、modify-db-cluster-parameter-group を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB DB クラスターパラメータグループを変更するには

次のmodify-db-cluster-parameter-group例では、2つのパラメータ audit_logsと custom3-6-param-grpを有効にttl_monitorして、Amazon DocumentDB クラスターパラメータグループを変更します。変更は次の再起動時に適用されます。

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --  
  parameters ParameterName=audit_logs,ParameterValue=enabled,ApplyMethod=pending-reboot \  
  
  ParameterName=ttl_monitor,ParameterValue=enabled,ApplyMethod=pending-reboot
```

出力:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターパラメータグループの変更](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbClusterParameterGroup](#)」の「」を参照してください。

modify-db-cluster-snapshot-attribute

次のコード例は、modify-db-cluster-snapshot-attribute を使用する方法を示しています。

AWS CLI

例 1: Amazon DocumentDB スナップショットに属性を追加するには

次のmodify-db-cluster-snapshot-attribute例では、Amazon DocumentDB クラスタースナップショットに4つの属性値を追加します。

```
aws docdb modify-db-cluster-snapshot-attribute \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --attribute-name restore \  
  --
```

```
--values-to-add 123456789011 123456789012 123456789013
```

出力:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789011",
          "123456789012",
          "123456789013"
        ]
      }
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
  }
}
```

例 2: Amazon DocumentDB スナップショットから属性を削除するには

次のmodify-db-cluster-snapshot-attribute例では、Amazon DocumentDB クラスター スナップショットから 2 つの属性値を削除します。

```
aws docdb modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier sample-cluster-snapshot \
  --attribute-name restore \
  --values-to-remove 123456789012
```

出力:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789011",
          "123456789013"
        ]
      }
    ]
  }
}
```

```
    ],  
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"  
  }  
}
```

詳細については、「Amazon DocumentDB デベロッパーガイド」の「[ModifyDBClusterSnapshotAttribute](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbClusterSnapshotAttribute](#)」の「」を参照してください。

modify-db-cluster

次の例は、modify-db-cluster を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB クラスターを変更するには

次のmodify-db-cluster例では、自動バックアップの保持期間を7日間とし、バックアップとメンテナンスの両方の優先ウィンドウを変更sample-clusterすることで、Amazon DocumentDB クラスターを変更します。すべての変更は、次のメンテナンスウィンドウで適用されます。

```
aws docdb modify-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --no-apply-immediately \  
  --backup-retention-period 7 \  
  --preferred-backup-window 18:00-18:30 \  
  --preferred-maintenance-window sun:20:00-sun:20:30
```

出力:

```
{  
  "DBCluster": {  
    "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-west-2.docdb.amazonaws.com",  
    "DBClusterMembers": [  
      {  
        "DBClusterParameterGroupStatus": "in-sync",  
        "DBInstanceIdentifier": "sample-cluster",  
        "IsClusterWriter": true,  
        "Role": "sample-cluster-aurora-role",  
        "Status": "available",  
        "SubnetIdentifier": "sample-cluster-subnet",  
        "VpcIdentifier": "sample-cluster-vpc"  
      }  
    ]  
  }  
}
```

```
        "PromotionTier": 1
      },
      {
        "DBClusterParameterGroupStatus": "in-sync",
        "DBInstanceIdentifier": "sample-cluster2",
        "IsClusterWriter": false,
        "PromotionTier": 2
      }
    ],
    "HostedZoneId": "ZNKXH85TT8WVW",
    "StorageEncrypted": false,
    "PreferredBackupWindow": "18:00-18:30",
    "MultiAZ": true,
    "EngineVersion": "3.6.0",
    "MasterUsername": "master-user",
    "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
    "DBSubnetGroup": "default",
    "LatestRestorableTime": "2019-03-18T22:08:13.408Z",
    "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
    "PreferredMaintenanceWindow": "sun:20:00-sun:20:30",
    "AssociatedRoles": [],
    "EnabledCloudwatchLogsExports": [
      "audit"
    ],
    "Engine": "docdb",
    "DBClusterParameterGroup": "default.docdb3.6",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
    "BackupRetentionPeriod": 7,
    "DBClusterIdentifier": "sample-cluster",
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2c",
      "us-west-2b"
    ],
    "Status": "available",
    "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
    "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-77186e0d",
        "Status": "active"
      }
    ],
  ],
```

```
    "Port": 27017
  }
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターの変更](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbCluster](#)」の「」を参照してください。

modify-db-instance

次の例は、`modify-db-instance` を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB インスタンスを変更するには

次の`modify-db-instance`例では、インスタンスクラスを `db.r4.large` に変更し、プロモーション階層を `sample-cluster2` に変更することで、Amazon DocumentDB インスタンスを変更します。変更はすぐに適用されますが、インスタンスのステータスが使用可能になった後に表示されます。

```
aws docdb modify-db-instance \
  --db-instance-identifier sample-cluster2 \
  --apply-immediately \
  --db-instance-class db.r4.large \
  --promotion-tier 5
```

出力:

```
{
  "DBInstance": {
    "EngineVersion": "3.6.0",
    "StorageEncrypted": false,
    "DBInstanceClass": "db.r4.large",
    "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
    "AutoMinorVersionUpgrade": true,
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-77186e0d",
        "Status": "active"
      }
    ]
  }
}
```



```
    }
  ],
  "PreferredBackupWindow": "18:00-18:30",
  "EnabledCloudwatchLogsExports": [
    "audit"
  ],
  "AvailabilityZone": "us-west-2f",
  "DBInstanceIdentifier": "sample-cluster2",
  "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
  "Engine": "docdb",
  "BackupRetentionPeriod": 7,
  "DBSubnetGroup": {
    "DBSubnetGroupName": "default",
    "DBSubnetGroupDescription": "default",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-4e26d263",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-afc329f4",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2c"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-53ab3636",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-991cb8d0",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        },
        "SubnetStatus": "Active"
      }
    ]
  }
}
```

```
    ],
    "VpcId": "vpc-91280df6"
  },
  "PromotionTier": 2,
  "Endpoint": {
    "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
    "HostedZoneId": "ZNKXH85TT8WW",
    "Port": 27017
  },
  "DbiResourceId": "db-A2GIKUV6KP0HITGGKI2NHVISZA",
  "DBClusterIdentifier": "sample-cluster",
  "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
  "PendingModifiedValues": {
    "DBInstanceClass": "db.r4.4xlarge"
  },
  "PubliclyAccessible": false,
  "DBInstanceStatus": "available"
}
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB インスタンスの変更](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbInstance](#)」の「」を参照してください。

modify-db-subnet-group

次の例は、modify-db-subnet-group を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB サブネットグループを変更するには

次のmodify-db-subnet-group例では、指定されたサブネットと新しい説明sample-subnet-groupを追加して、サブネットグループを変更します。

```
aws docdb modify-db-subnet-group \  
  --db-subnet-group-name sample-subnet-group \  
  --subnet-ids subnet-b3806e8f subnet-53ab3636 subnet-991cb8d0 \  
  --db-subnet-group-description "New subnet description"
```

出力:

```
{
  "DBSubnetGroup": {
    "DBSubnetGroupName": "sample-subnet-group",
    "SubnetGroupStatus": "Complete",
    "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
    "VpcId": "vpc-91280df6",
    "DBSubnetGroupDescription": "New subnet description",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-b3806e8f",
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      },
      {
        "SubnetIdentifier": "subnet-53ab3636",
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2c"
        }
      },
      {
        "SubnetIdentifier": "subnet-991cb8d0",
        "SubnetStatus": "Active",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        }
      }
    ]
  }
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB サブネットグループの変更](#)」を参照してください。 Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbSubnetGroup](#)」の「」を参照してください。

reboot-db-instance

次のコード例は、reboot-db-instance を使用する方法を示しています。

AWS CLI

Amazon DocumentDB インスタンスを再起動するには

次のreboot-db-instance例では、Amazon DocumentDB インスタンス を再起動し
ますsample-cluster2。

```
aws docdb reboot-db-instance \  
  --db-instance-identifier sample-cluster2
```

このコマンドでは何も出力されません。出力:

```
{  
  "DBInstance": {  
    "PreferredBackupWindow": "18:00-18:30",  
    "DBInstanceIdentifier": "sample-cluster2",  
    "VpcSecurityGroups": [  
      {  
        "Status": "active",  
        "VpcSecurityGroupId": "sg-77186e0d"  
      }  
    ],  
    "DBSubnetGroup": {  
      "VpcId": "vpc-91280df6",  
      "Subnets": [  
        {  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2a"  
          },  
          "SubnetIdentifier": "subnet-4e26d263"  
        },  
        {  
          "SubnetStatus": "Active",  
          "SubnetAvailabilityZone": {  
            "Name": "us-west-2c"  
          },  
          "SubnetIdentifier": "subnet-afc329f4"  
        }  
      ]  
    }  
  }  
}
```

```
    {
      "SubnetStatus": "Active",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      },
      "SubnetIdentifier": "subnet-53ab3636"
    },
    {
      "SubnetStatus": "Active",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      },
      "SubnetIdentifier": "subnet-991cb8d0"
    }
  ],
  "SubnetGroupStatus": "Complete",
  "DBSubnetGroupName": "default",
  "DBSubnetGroupDescription": "default"
},
"PendingModifiedValues": {},
"Endpoint": {
  "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
  "HostedZoneId": "ZNKXH85TT8WVW",
  "Port": 27017
},
"EnabledCloudwatchLogsExports": [
  "audit"
],
"StorageEncrypted": false,
"DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
"AutoMinorVersionUpgrade": true,
"Engine": "docdb",
"InstanceCreateTime": "2019-03-15T20:36:06.338Z",
"EngineVersion": "3.6.0",
"PromotionTier": 5,
"BackupRetentionPeriod": 7,
"DBClusterIdentifier": "sample-cluster",
"PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
"PubliclyAccessible": false,
"DBInstanceClass": "db.r4.4xlarge",
"AvailabilityZone": "us-west-2d",
"DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
"DBInstanceStatus": "rebooting"
}
```

```
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB の再起動Instance](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootDbInstance](#)」の「」を参照してください。

remove-tags-from-resource

次の例は、`remove-tags-from-resource` を使用する方法を説明しています。

AWS CLI

Amazon DocumentDB リソースからタグを削除するには

次の`remove-tags-from-resource`例では、Amazon DocumentDB クラスター Bから という名前のキーを持つ タグを削除します `sample-cluster`。

```
aws docdb remove-tags-from-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
  --tag-keys B
```

このコマンドでは何も出力されません。

詳細については、「[Amazon DocumentDB デベロッパーガイド](#)」の「[Amazon D からのタグの削除DocumentDBResource](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveTagsFromResource](#)」の「」を参照してください。

reset-db-cluster-parameter-group

次の例は、`reset-db-cluster-parameter-group` を使用する方法を説明しています。

AWS CLI

指定されたパラメータ値を Amazon DocumentDB パラメータグループのデフォルトにリセットするには

次の`reset-db-cluster-parameter-group`例では、Amazon DocumentDB パラメータグループのパラメータ `t1_monitor` をデフォルト値 `custom3-6-param-grp` にリセットします。

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --parameters ParameterName=ttl_monitor,ApplyMethod=immediate
```

出力:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

詳細については、Amazon DocumentDB デベロッパーガイドの「タイトル」を参照してください。

Amazon DocumentDB パラメータグループの指定されたパラメータ値またはすべてのパラメータ値をデフォルトにリセットするには

次のreset-db-cluster-parameter-group例では、Amazon DocumentDB パラメータグループのすべてのパラメータをデフォルト値custom3-6-param-grpにリセットします。

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --reset-all-parameters
```

出力:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターパラメータグループのリセット](#)」を参照してください。 Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetDbClusterParameterGroup](#)」の「」を参照してください。

restore-db-cluster-from-snapshot

次のコード例は、restore-db-cluster-from-snapshot を使用する方法を示しています。

AWS CLI

自動スナップショットまたは手動スナップショットから Amazon DocumentDB クラスターを復元するには

次の `restore-db-cluster-from-snapshot` 例では、スナップショット `sample-cluster-2019-03-16-00-01-restored` から という名前の新しい Amazon DocumentDB クラスターを作成します `rds:sample-cluster-2019-03-16-00-01`。

```
aws docdb restore-db-cluster-from-snapshot \  
  --db-cluster-identifier sample-cluster-2019-03-16-00-01-restored \  
  --engine docdb \  
  --snapshot-identifier rds:sample-cluster-2019-03-16-00-01
```

出力:

```
{  
  "DBCluster": {  
    "ClusterCreateTime": "2019-03-19T18:45:01.857Z",  
    "HostedZoneId": "ZNKXH85TT8WVW",  
    "Engine": "docdb",  
    "DBClusterMembers": [],  
    "MultiAZ": false,  
    "AvailabilityZones": [  
      "us-west-2a",  
      "us-west-2c",  
      "us-west-2b"  
    ],  
    "StorageEncrypted": false,  
    "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-  
corcjzr1sfc.us-west-2.docdb.amazonaws.com",  
    "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-  
corcjzr1sfc.us-west-2.docdb.amazonaws.com",  
    "Port": 27017,  
    "PreferredBackupWindow": "00:00-00:30",  
    "DBSubnetGroup": "default",  
    "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",  
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",  
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-  
cluster-2019-03-16-00-01-restored",  
    "DBClusterParameterGroup": "default.docdb3.6",  
    "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",
```



```

    "MasterUsername": "master-user",
    "EngineVersion": "3.6.0",
    "BackupRetentionPeriod": 3,
    "AssociatedRoles": [],
    "Status": "creating",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ]
  }
}

```

詳細については、[「Amazon DocumentDB デベロッパーガイド」の「クラスタースナップショットからの復元」](#)を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreDbClusterFromSnapshot](#)」の「」を参照してください。

restore-db-cluster-to-point-in-time

次の例は、`restore-db-cluster-to-point-in-time` を使用する方法を説明しています。

AWS CLI

手動スナップショットから Amazon DocumentDB クラスターを point-in-time に復元するには

次の `restore-db-cluster-to-point-in-time` 例では `sample-cluster-snapshot`、を使用して、最新の復元可能時間 `sample-cluster-pit` を使用して新しい Amazon DocumentDB クラスターを作成します。

```

aws docdb restore-db-cluster-to-point-in-time \
  --db-cluster-identifier sample-cluster-pit \
  --source-db-cluster-identifier arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
  --use-latest-restorable-time

```

出力:

```
{
```

```
"DBCluster": {
  "StorageEncrypted": false,
  "BackupRetentionPeriod": 3,
  "MasterUsername": "master-user",
  "HostedZoneId": "ZNKXH85TT8WVW",
  "PreferredBackupWindow": "00:00-00:30",
  "MultiAZ": false,
  "DBClusterIdentifier": "sample-cluster-pit",
  "DBSubnetGroup": "default",
  "ClusterCreateTime": "2019-04-03T15:55:21.320Z",
  "AssociatedRoles": [],
  "DBClusterParameterGroup": "default.docdb3.6",
  "DBClusterMembers": [],
  "Status": "creating",
  "AvailabilityZones": [
    "us-west-2a",
    "us-west-2d",
    "us-west-2b"
  ],
  "ReaderEndpoint": "sample-cluster-pit.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
  "Port": 27017,
  "Engine": "docdb",
  "EngineVersion": "3.6.0",
  "VpcSecurityGroups": [
    {
      "VpcSecurityGroupId": "sg-77186e0d",
      "Status": "active"
    }
  ],
  "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
  "Endpoint": "sample-cluster-pit.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
  "DbClusterResourceId": "cluster-NLCABBX0SE2QPQ4GOLZIFWEPLM",
  "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
pit"
}
```

詳細については、Amazon DocumentDB デベロッパーガイド [「時点へのスナップショットの復元」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RestoreDbClusterToPointInTime](#)」の「」を参照してください。

start-db-cluster

次のコード例は、start-db-cluster を使用する方法を示しています。

AWS CLI

停止した Amazon DocumentDB クラスターを開始するには

次のstart-db-cluster例では、指定された Amazon DocumentDB クラスターを起動します。

```
aws docdb start-db-cluster \  
  --db-cluster-identifier sample-cluster
```

出力:

```
{  
  "DBCluster": {  
    "ClusterCreateTime": "2019-03-19T18:45:01.857Z",  
    "HostedZoneId": "ZNKXH85TT8WVW",  
    "Engine": "docdb",  
    "DBClusterMembers": [],  
    "MultiAZ": false,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1c",  
      "us-east-1f"  
    ],  
    "StorageEncrypted": false,  
    "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-  
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",  
    "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-  
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",  
    "Port": 27017,  
    "PreferredBackupWindow": "00:00-00:30",  
    "DBSubnetGroup": "default",  
    "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",  
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",  
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-  
cluster-2019-03-16-00-01-restored",  
    "DBClusterParameterGroup": "default.docdb3.6",  
    "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",  
    "MasterUsername": "master-user",  
    "EngineVersion": "3.6.0",
```

```
    "BackupRetentionPeriod": 3,
    "AssociatedRoles": [],
    "Status": "creating",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ]
  }
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターの停止と起動](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[StartDbCluster](#)」の「」を参照してください。

stop-db-cluster

次のコード例は、stop-db-cluster を使用する方法を示しています。

AWS CLI

実行中の Amazon DocumentDB クラスターを停止するには

次のstop-db-cluster例では、指定された Amazon DocumentDB クラスターを停止します。

```
aws docdb stop-db-cluster \
  --db-cluster-identifier sample-cluster
```

出力:

```
{
  "DBCluster": {
    "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
    "HostedZoneId": "ZNKXH85TT8WVW",
    "Engine": "docdb",
    "DBClusterMembers": [],
    "MultiAZ": false,
    "AvailabilityZones": [
```

```
        "us-east-1a",
        "us-east-1c",
        "us-east-1f"
    ],
    "StorageEncrypted": false,
    "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",
    "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",
    "Port": 27017,
    "PreferredBackupWindow": "00:00-00:30",
    "DBSubnetGroup": "default",
    "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
    "DBClusterParameterGroup": "default.docdb3.6",
    "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",
    "MasterUsername": "master-user",
    "EngineVersion": "3.6.0",
    "BackupRetentionPeriod": 3,
    "AssociatedRoles": [],
    "Status": "creating",
    "VpcSecurityGroups": [
        {
            "Status": "active",
            "VpcSecurityGroupId": "sg-77186e0d"
        }
    ]
}
}
```

詳細については、[Amazon DocumentDB デベロッパーガイド](#)の「[Amazon DocumentDB クラスターの停止と起動](#)」を参照してください。Amazon DocumentDB

- API 詳細については、AWS CLI 「コマンドリファレンス[StopDbCluster](#)」の「」を参照してください。

を使用した DynamoDB の例 AWS CLI

次のコード例は、DynamoDB AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-get-item

次のコード例は、batch-get-item を使用する方法を示しています。

AWS CLI

テーブルから複数の項目を取得するには

次の batch-get-items の例では、3 つの GetItem リクエストのバッチを使用して MusicCollection テーブルから複数の項目を読み込み、この操作で使用された読み込み容量ユニットの数を取得します。このコマンドは AlbumTitle 属性のみを返します。

```
aws dynamodb batch-get-item \  
  --request-items file://request-items.json \  
  --return-consumed-capacity TOTAL
```

request-items.json の内容:

```
{  
  "MusicCollection": {  
    "Keys": [  
      {  
        "Artist": {"S": "No One You Know"},  
        "SongTitle": {"S": "Call Me Today"}  
      },  
      {  
        "Artist": {"S": "Acme Band"},  
        "SongTitle": {"S": "Happy Day"}  
      }  
    ]  
  }  
}
```

```
    },
    {
      "Artist": {"S": "No One You Know"},
      "SongTitle": {"S": "Scared of My Shadow"}
    }
  ],
  "ProjectionExpression": "AlbumTitle"
}
}
```

出力:

```
{
  "Responses": {
    "MusicCollection": [
      {
        "AlbumTitle": {
          "S": "Somewhat Famous"
        }
      },
      {
        "AlbumTitle": {
          "S": "Blue Sky Blues"
        }
      },
      {
        "AlbumTitle": {
          "S": "Louder Than Ever"
        }
      }
    ]
  },
  "UnprocessedKeys": {},
  "ConsumedCapacity": [
    {
      "TableName": "MusicCollection",
      "CapacityUnits": 1.5
    }
  ]
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[バッチオペレーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetItem](#)」の「」を参照してください。

batch-write-item

次の例は、batch-write-item を使用する方法を説明しています。

AWS CLI

テーブルに複数の項目を追加するには

次の batch-write-item の例は、3 つの PutItem リクエストのバッチを使用して、MusicCollection テーブルに 3 つの新しい項目を追加します。また、このオペレーションによって消費される書き込み容量ユニットの数、およびオペレーションによって変更されるアイテムコレクションに関する情報も要求します。

```
aws dynamodb batch-write-item \  
  --request-items file://request-items.json \  
  --return-consumed-capacity INDEXES \  
  --return-item-collection-metrics SIZE
```

request-items.json の内容:

```
{  
  "MusicCollection": [  
    {  
      "PutRequest": {  
        "Item": {  
          "Artist": {"S": "No One You Know"},  
          "SongTitle": {"S": "Call Me Today"},  
          "AlbumTitle": {"S": "Somewhat Famous"}  
        }  
      }  
    },  
    {  
      "PutRequest": {  
        "Item": {  
          "Artist": {"S": "Acme Band"},  
          "SongTitle": {"S": "Happy Day"},  
          "AlbumTitle": {"S": "Songs About Life"}  
        }  
      }  
    }  
  ]  
}
```



```
    },
    {
      "PutRequest": {
        "Item": {
          "Artist": {"S": "No One You Know"},
          "SongTitle": {"S": "Scared of My Shadow"},
          "AlbumTitle": {"S": "Blue Sky Blues"}
        }
      }
    }
  ]
}
```

出力:

```
{
  "UnprocessedItems": {},
  "ItemCollectionMetrics": {
    "MusicCollection": [
      {
        "ItemCollectionKey": {
          "Artist": {
            "S": "No One You Know"
          }
        },
        "SizeEstimateRangeGB": [
          0.0,
          1.0
        ]
      },
      {
        "ItemCollectionKey": {
          "Artist": {
            "S": "Acme Band"
          }
        },
        "SizeEstimateRangeGB": [
          0.0,
          1.0
        ]
      }
    ]
  },
}
```

```
"ConsumedCapacity": [
  {
    "TableName": "MusicCollection",
    "CapacityUnits": 6.0,
    "Table": {
      "CapacityUnits": 3.0
    },
    "LocalSecondaryIndexes": {
      "AlbumTitleIndex": {
        "CapacityUnits": 3.0
      }
    }
  }
]
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[バッチオペレーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchWriteItem](#)」の「」を参照してください。

create-backup

次のコード例は、create-backup を使用する方法を示しています。

AWS CLI

既存の DynamoDB テーブルのバックアップを作成するには

次のcreate-backup例では、MusicCollectionテーブルのバックアップを作成します。

```
aws dynamodb create-backup \
  --table-name MusicCollection \
  --backup-name MusicCollectionBackup
```

出力:

```
{
  "BackupDetails": {
    "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a",
```

```
    "BackupName": "MusicCollectionBackup",
    "BackupSizeBytes": 0,
    "BackupStatus": "CREATING",
    "BackupType": "USER",
    "BackupCreationDateTime": 1576616366.715
  }
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBackup](#)」の「」を参照してください。

create-global-table

次の例は、create-global-table を使用する方法を説明しています。

AWS CLI

グローバルテーブルを作成するには

次のcreate-global-table例では、指定された別々の AWS リージョンにある 2 つの同じテーブルからグローバルテーブルを作成します。

```
aws dynamodb create-global-table \
  --global-table-name MusicCollection \
  --replication-group RegionName=us-east-2 RegionName=us-east-1 \
  --region us-east-2
```

出力:

```
{
  "GlobalTableDescription": {
    "ReplicationGroup": [
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-east-1"
      }
    ]
  },
}
```

```

    "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
    "CreationDateTime": 1576625818.532,
    "GlobalTableStatus": "CREATING",
    "GlobalTableName": "MusicCollection"
  }
}

```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateGlobalTable](#)」の「」を参照してください。

create-table

次の例は、create-table を使用する方法を説明しています。

AWS CLI

例 1: タグ付きのテーブルを作成するには

次の create-table の例では、指定された属性とキースキーマを使用して、MusicCollection という名前のテーブルを作成します。このテーブルは、プロビジョニングされたスループットを使用し、保管時にデフォルト AWS 所有の を使用して暗号化されます CMK。またこのコマンドは、Owner キーと blueTeam 値を使用して、テーブルにタグを適用します。

```

aws dynamodb create-table \
  --table-name MusicCollection \
  --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
\
  --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
  --tags Key=Owner,Value=blueTeam

```

出力:

```

{
  "TableDescription": {

```

```
"AttributeDefinitions": [
  {
    "AttributeName": "Artist",
    "AttributeType": "S"
  },
  {
    "AttributeName": "SongTitle",
    "AttributeType": "S"
  }
],
"ProvisionedThroughput": {
  "NumberOfDecreasesToday": 0,
  "WriteCapacityUnits": 5,
  "ReadCapacityUnits": 5
},
"TableSizeBytes": 0,
"TableName": "MusicCollection",
"TableStatus": "CREATING",
"KeySchema": [
  {
    "KeyType": "HASH",
    "AttributeName": "Artist"
  },
  {
    "KeyType": "RANGE",
    "AttributeName": "SongTitle"
  }
],
"ItemCount": 0,
"CreationDateTime": "2020-05-26T16:04:41.627000-07:00",
"TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
"TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB テーブルの基本的なオペレーション](#)」を参照してください。

例 2: オンデマンドモードでテーブルを作成するには

次の例は、プロビジョニングされたスループットモードではなく、オンデマンドモードを使用して MusicCollection というテーブルを作成します。これは、ワークロードが予測できないテーブルに役立ちます。

```
aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-  
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S  
 \  
  --key-  
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \  
  --billing-mode PAY_PER_REQUEST
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "CREATING",  
    "CreationDateTime": "2020-05-27T11:44:10.807000-07:00",  
    "ProvisionedThroughput": {  
      "NumberOfDecreasesToday": 0,  
      "ReadCapacityUnits": 0,  
      "WriteCapacityUnits": 0  
    },  
    "TableSizeBytes": 0,  
    "ItemCount": 0,  
  }  
}
```

```

    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "BillingModeSummary": {
      "BillingMode": "PAY_PER_REQUEST"
    }
  }
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB テーブルの基本的なオペレーション](#)」を参照してください。

例 3: テーブルを作成して Customer Managed で暗号化するには CMK

次の例では、という名前のテーブルを作成し MusicCollection、カスタマーマネージドを使用して暗号化します CMK。

```

aws dynamodb create-table \
  --table-name MusicCollection \
  --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S
  \
  --key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
  --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=abcd1234-abcd-1234-
a123-ab1234a1b234

```

出力:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "MusicCollection",

```

```

    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "CREATING",
    "CreationDateTime": "2020-05-27T11:12:16.431000-07:00",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "SSEDescription": {
      "Status": "ENABLED",
      "SSEType": "KMS",
      "KMSMasterKeyArn": "arn:aws:kms:us-west-2:123456789012:key/abcd1234-
abcd-1234-a123-ab1234a1b234"
    }
  }
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB テーブルの基本的なオペレーション](#)」を参照してください。

例 4: ローカルセカンダリインデックスを持つテーブルを作成するには

次の例では、指定された属性とキースキーマを使用して、AlbumTitleIndex という名前のローカルセカンダリインデックスを持つ MusicCollection という名前のテーブルを作成します。

```

aws dynamodb create-table \
  --table-name MusicCollection \
  --attribute-
definitions AttributeName=Artist,AttributeType=S AttributeName=SongTitle,AttributeType=S Att
  \

```



```

--key-
schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \
--provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
--local-secondary-indexes \
  "[
    {
      \"IndexName\": \"AlbumTitleIndex\",
      \"KeySchema\": [
        {\"AttributeName\": \"Artist\", \"KeyType\": \"HASH\"},
        {\"AttributeName\": \"AlbumTitle\", \"KeyType\": \"RANGE\"}
      ],
      \"Projection\": {
        \"ProjectionType\": \"INCLUDE\",
        \"NonKeyAttributes\": [\"Genre\", \"Year\"]
      }
    }
  ]"

```

出力:

```

{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "AlbumTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "MusicCollection",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {

```

```
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
    }
],
"TableStatus": "CREATING",
"CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
"ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 10,
    "WriteCapacityUnits": 5
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
"TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"LocalSecondaryIndexes": [
    {
        "IndexName": "AlbumTitleIndex",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "AlbumTitle",
                "KeyType": "RANGE"
            }
        ],
        "Projection": {
            "ProjectionType": "INCLUDE",
            "NonKeyAttributes": [
                "Genre",
                "Year"
            ]
        },
        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
    }
]
}
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB テーブルの基本的なオペレーション](#)」を参照してください。

例 5: グローバルセカンダリインデックスを持つテーブルを作成するには

次の例では、GameTitleIndex という名前のグローバルセカンダリインデックスを持つ GameScores という名前のテーブルを作成します。ベーステーブルには のパーティションキーUserIdと のソートキーがありGameTitle、特定のゲームの個々のユーザーの最適なスコアを効率的に見つけることができます。一方、GSIには のパーティションキーGameTitleと のソートキーがありTopScore、特定のゲームの全体的な最高スコアをすばやく見つけることができます。

```
aws dynamodb create-table \  
  --table-name GameScores \  
  --attribute-  
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att  
 \  
  --key-schema AttributeName=UserId,KeyType=HASH \  
               AttributeName=GameTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \  
  --global-secondary-indexes \  
    "[  
      {  
        \"IndexName\": \"GameTitleIndex\",  
        \"KeySchema\": [  
          {\"AttributeName\": \"GameTitle\", \"KeyType\": \"HASH\"},  
          {\"AttributeName\": \"TopScore\", \"KeyType\": \"RANGE\"}  
        ],  
        \"Projection\": {  
          \"ProjectionType\": \"INCLUDE\",  
          \"NonKeyAttributes\": [\"UserId\"]  
        },  
        \"ProvisionedThroughput\": {  
          \"ReadCapacityUnits\": 10,  
          \"WriteCapacityUnits\": 5  
        }  
      }  
    ]"
```

出力:

```
{
```

```
"TableDescription": {
  "AttributeDefinitions": [
    {
      "AttributeName": "GameTitle",
      "AttributeType": "S"
    },
    {
      "AttributeName": "TopScore",
      "AttributeType": "N"
    },
    {
      "AttributeName": "UserId",
      "AttributeType": "S"
    }
  ],
  "TableName": "GameScores",
  "KeySchema": [
    {
      "AttributeName": "UserId",
      "KeyType": "HASH"
    },
    {
      "AttributeName": "GameTitle",
      "KeyType": "RANGE"
    }
  ],
  "TableStatus": "CREATING",
  "CreationDateTime": "2020-05-26T17:28:15.602000-07:00",
  "ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 10,
    "WriteCapacityUnits": 5
  },
  "TableSizeBytes": 0,
  "ItemCount": 0,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
  "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "GlobalSecondaryIndexes": [
    {
      "IndexName": "GameTitleIndex",
      "KeySchema": [
        {
          "AttributeName": "GameTitle",
          "KeyType": "HASH"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "AttributeName": "TopScore",
            "KeyType": "RANGE"
        }
    ],
    "Projection": {
        "ProjectionType": "INCLUDE",
        "NonKeyAttributes": [
            "UserId"
        ]
    },
    "IndexStatus": "CREATING",
    "ProvisionedThroughput": {
        "NumberOfDecreasesToday": 0,
        "ReadCapacityUnits": 10,
        "WriteCapacityUnits": 5
    },
    "IndexSizeBytes": 0,
    "ItemCount": 0,
    "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
    }
]
}
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB テーブルの基本的なオペレーション](#)」を参照してください。

例 6: 複数のグローバルセカンダリインデックスを持つテーブルを一度に作成するには

次の例では、2 つのグローバルセカンダリインデックスを持つ GameScores という名前のテーブルを作成します。GSI スキーマは、コマンドラインではなく ファイルを介して渡されます。

```

aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
  \
  --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \

```

```
--global-secondary-indexes file://gsi.json
```

gsi.json の内容:

```
[
  {
    "IndexName": "GameTitleIndex",
    "KeySchema": [
      {
        "AttributeName": "GameTitle",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "TopScore",
        "KeyType": "RANGE"
      }
    ],
    "Projection": {
      "ProjectionType": "ALL"
    },
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 5
    }
  },
  {
    "IndexName": "GameDateIndex",
    "KeySchema": [
      {
        "AttributeName": "GameTitle",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "Date",
        "KeyType": "RANGE"
      }
    ],
    "Projection": {
      "ProjectionType": "ALL"
    },
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    }
  }
]
```

```
    }  
  }  
]
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Date",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "GameTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "TopScore",  
        "AttributeType": "N"  
      },  
      {  
        "AttributeName": "UserId",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "GameScores",  
    "KeySchema": [  
      {  
        "AttributeName": "UserId",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "GameTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "CREATING",  
    "CreationDateTime": "2020-08-04T16:40:55.524000-07:00",  
    "ProvisionedThroughput": {  
      "NumberOfDecreasesToday": 0,  
      "ReadCapacityUnits": 10,  
      "WriteCapacityUnits": 5  
    }  
  }  
}
```

```
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "GlobalSecondaryIndexes": [
      {
        "IndexName": "GameTitleIndex",
        "KeySchema": [
          {
            "AttributeName": "GameTitle",
            "KeyType": "HASH"
          },
          {
            "AttributeName": "TopScore",
            "KeyType": "RANGE"
          }
        ],
        "Projection": {
          "ProjectionType": "ALL"
        },
        "IndexStatus": "CREATING",
        "ProvisionedThroughput": {
          "NumberOfDecreasesToday": 0,
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 5
        },
        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameTitleIndex"
      },
      {
        "IndexName": "GameDateIndex",
        "KeySchema": [
          {
            "AttributeName": "GameTitle",
            "KeyType": "HASH"
          },
          {
            "AttributeName": "Date",
            "KeyType": "RANGE"
          }
        ]
      }
    ],
```



```

        "Projection": {
            "ProjectionType": "ALL"
        },
        "IndexStatus": "CREATING",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/index/GameDateIndex"
    }
]
}
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB テーブルの基本的なオペレーション](#)」を参照してください。

例 7: ストリームが有効なテーブルを作成するには

次の例では、DynamoDB ストリームを有効にした状態の GameScores という名前のテーブルを作成します。各アイテムの新しいイメージと古いイメージの両方がストリームに書き込まれます。

```

aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
  \
  --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --stream-specification StreamEnabled=TRUE,StreamViewType=NEW_AND_OLD_IMAGES

```

出力:

```

{
  "TableDescription": {
    "AttributeDefinitions": [

```

```
{
  {
    "AttributeName": "GameTitle",
    "AttributeType": "S"
  },
  {
    "AttributeName": "UserId",
    "AttributeType": "S"
  }
],
"TableName": "GameScores",
"KeySchema": [
  {
    "AttributeName": "UserId",
    "KeyType": "HASH"
  },
  {
    "AttributeName": "GameTitle",
    "KeyType": "RANGE"
  }
],
"TableStatus": "CREATING",
"CreationDateTime": "2020-05-27T10:49:34.056000-07:00",
"ProvisionedThroughput": {
  "NumberOfDecreasesToday": 0,
  "ReadCapacityUnits": 10,
  "WriteCapacityUnits": 5
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
"TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"StreamSpecification": {
  "StreamEnabled": true,
  "StreamViewType": "NEW_AND_OLD_IMAGES"
},
"LatestStreamLabel": "2020-05-27T17:49:34.056",
"LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2020-05-27T17:49:34.056"
}
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB テーブルの基本的なオペレーション](#)」を参照してください。

例 8: Keys-Only ストリームが有効なテーブルを作成するには

次の例では、DynamoDB ストリームを有効にした状態の GameScores という名前のテーブルを作成します。変更された項目のキー属性のみがストリームに書き込まれます。

```
aws dynamodb create-table \  
  --table-name GameScores \  
  --attribute-  
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S  
 \  
  --key-  
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \  
  --stream-specification StreamEnabled=TRUE,StreamViewType=KEYS_ONLY
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "GameTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "UserId",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "GameScores",  
    "KeySchema": [  
      {  
        "AttributeName": "UserId",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "GameTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "CREATING",  
    "CreationDateTime": "2023-05-25T18:45:34.140000+00:00",  
    "ProvisionedThroughput": {
```

```

        "NumberOfDecreasesToday": 0,
        "ReadCapacityUnits": 10,
        "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "StreamSpecification": {
        "StreamEnabled": true,
        "StreamViewType": "KEYS_ONLY"
    },
    "LatestStreamLabel": "2023-05-25T18:45:34.140",
    "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2023-05-25T18:45:34.140",
    "DeletionProtectionEnabled": false
}
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB Streams の変更データキャプチャ](#)」を参照してください。

例 9: Standard Infrequent Access クラスでテーブルを作成するには

次の例は、GameScores という名前のテーブルを作成し、Standard-Infrequent Access (DynamoDB Standard-IA) テーブルクラスを割り当てます。このテーブルクラスは、ストレージが主なコストとなるように最適化されています。

```

aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S
 \
  --key-
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --table-class STANDARD_INFREQUENT_ACCESS

```

出力:

```

{
  "TableDescription": {
    "AttributeDefinitions": [

```

```
    {
      "AttributeName": "GameTitle",
      "AttributeType": "S"
    },
    {
      "AttributeName": "UserId",
      "AttributeType": "S"
    }
  ],
  "TableName": "GameScores",
  "KeySchema": [
    {
      "AttributeName": "UserId",
      "KeyType": "HASH"
    },
    {
      "AttributeName": "GameTitle",
      "KeyType": "RANGE"
    }
  ],
  "TableStatus": "CREATING",
  "CreationDateTime": "2023-05-25T18:33:07.581000+00:00",
  "ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 10,
    "WriteCapacityUnits": 5
  },
  "TableSizeBytes": 0,
  "ItemCount": 0,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
  "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "TableClassSummary": {
    "TableClass": "STANDARD_INFREQUENT_ACCESS"
  },
  "DeletionProtectionEnabled": false
}
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[テーブルクラス](#)」を参照してください。

例 10: 削除保護を有効にしたテーブルを作成するには

次の例では、GameScores というテーブルを作成し、削除保護を有効にします。

```
aws dynamodb create-table \  
  --table-name GameScores \  
  --attribute-  
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S  
 \  
  --key-  
schema AttributeName=UserId,KeyType=HASH AttributeName=GameTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \  
  --deletion-protection-enabled
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "GameTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "UserId",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "GameScores",  
    "KeySchema": [  
      {  
        "AttributeName": "UserId",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "GameTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "CREATING",  
    "CreationDateTime": "2023-05-25T23:02:17.093000+00:00",  
    "ProvisionedThroughput": {  
      "NumberOfDecreasesToday": 0,  
      "ReadCapacityUnits": 10,  
      "WriteCapacityUnits": 5  
    },  
    "TableSizeBytes": 0,  
  }  
}
```

```
    "ItemCount": 0,  
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",  
    "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "DeletionProtectionEnabled": true  
  }  
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[削除保護の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTable](#)」の「」を参照してください。

delete-backup

次のコード例は、delete-backup を使用する方法を示しています。

AWS CLI

既存の DynamoDB バックアップを削除するには

次のdelete-backup例では、指定された既存のバックアップを削除します。

```
aws dynamodb delete-backup \  
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/  
  backup/01576616366715-b4e58d3a
```

出力:

```
{  
  "BackupDescription": {  
    "BackupDetails": {  
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/  
MusicCollection/backup/01576616366715-b4e58d3a",  
      "BackupName": "MusicCollectionBackup",  
      "BackupSizeBytes": 0,  
      "BackupStatus": "DELETED",  
      "BackupType": "USER",  
      "BackupCreationDateTime": 1576616366.715  
    },  
    "SourceTableDetails": {
```

```
    "TableName": "MusicCollection",
    "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
    "TableSizeBytes": 0,
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableCreationDateTime": 1576615228.571,
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "ItemCount": 0,
    "BillingMode": "PROVISIONED"
  },
  "SourceTableFeatureDetails": {}
}
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBackup](#)」の「」を参照してください。

delete-item

次のコード例は、delete-item を使用する方法を示しています。

AWS CLI

例 1: 項目を削除するには

次の delete-item の例は、MusicCollection テーブルから項目を削除し、削除した項目とそのリクエストで使用された容量に関する詳細を取得します。


```
aws dynamodb delete-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --return-values ALL_OLD \  
  --return-consumed-capacity TOTAL \  
  --return-item-collection-metrics SIZE
```

key.json の内容:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Scared of My Shadow"}  
}
```

出力:

```
{  
  "Attributes": {  
    "AlbumTitle": {  
      "S": "Blue Sky Blues"  
    },  
    "Artist": {  
      "S": "No One You Know"  
    },  
    "SongTitle": {  
      "S": "Scared of My Shadow"  
    }  
  },  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 2.0  
  },  
  "ItemCollectionMetrics": {  
    "ItemCollectionKey": {  
      "Artist": {  
        "S": "No One You Know"  
      }  
    },  
    "SizeEstimateRangeGB": [  
      0.0,  
      1.0  
    ]  
  }  
}
```

```
}  
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目を書き込みます](#)」を参照してください。

例 2: 条件付きで項目を削除するには

次の例では、ProductCategory が Sporting Goods または Gardening Supplies で、その価格が 500 および 600 の場合のみ、ProductCatalog テーブルから項目を削除します。削除された項目に関する詳細が返されます。

```
aws dynamodb delete-item \  
  --table-name ProductCatalog \  
  --key '{"Id":{"N":"456"}}' \  
  --condition-expression "(ProductCategory IN (:cat1, :cat2)) and (#P between :lo  
and :hi)" \  
  --expression-attribute-names file://names.json \  
  --expression-attribute-values file://values.json \  
  --return-values ALL_OLD
```

names.json の内容:

```
{  
  "#P": "Price"  
}
```

values.json の内容:

```
{  
  ":cat1": {"S": "Sporting Goods"},  
  ":cat2": {"S": "Gardening Supplies"},  
  ":lo": {"N": "500"},  
  ":hi": {"N": "600"}  
}
```

出力:

```
{  
  "Attributes": {  
    "Id": {
```

```
        "N": "456"
      },
      "Price": {
        "N": "550"
      },
      "ProductCategory": {
        "S": "Sporting Goods"
      }
    }
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目を書き込みます](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteItem](#)」の「」を参照してください。

delete-table

次のコード例は、delete-table を使用する方法を示しています。

AWS CLI

テーブルを削除するには

以下の delete-table の例は MusicCollection テーブルを削除します。

```
aws dynamodb delete-table \  
  --table-name MusicCollection
```

出力:

```
{  
  "TableDescription": {  
    "TableStatus": "DELETING",  
    "TableSizeBytes": 0,  
    "ItemCount": 0,  
    "TableName": "MusicCollection",  
    "ProvisionedThroughput": {  
      "NumberOfDecreasesToday": 0,  
      "WriteCapacityUnits": 5,  
      "ReadCapacityUnits": 5  
    }  
  }  
}
```

```
    }  
  }  
}
```

詳細については、「Amazon DynamoDB デイベロPPERガイド」の「[テーブルの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTable](#)」の「」を参照してください。

describe-backup

次のコード例は、describe-backup を使用する方法を示しています。

AWS CLI

テーブルの既存のバックアップに関する情報を取得するには

次のdescribe-backup例では、指定された既存のバックアップに関する情報を表示します。

```
aws dynamodb describe-backup \  
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/  
backup/01576616366715-b4e58d3a
```

出力:

```
{  
  "BackupDescription": {  
    "BackupDetails": {  
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/  
MusicCollection/backup/01576616366715-b4e58d3a",  
      "BackupName": "MusicCollectionBackup",  
      "BackupSizeBytes": 0,  
      "BackupStatus": "AVAILABLE",  
      "BackupType": "USER",  
      "BackupCreationDateTime": 1576616366.715  
    },  
    "SourceTableDetails": {  
      "TableName": "MusicCollection",  
      "TableId": "b0c04bcc-309b-4352-b2ae-9088af169fe2",  
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/  
MusicCollection",  
    }  
  }  
}
```

```
    "TableSizeBytes": 0,
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableCreationDateTime": 1576615228.571,
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "ItemCount": 0,
    "BillingMode": "PROVISIONED"
  },
  "SourceTableFeatureDetails": {}
}
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeBackup](#)」の「」を参照してください。

describe-continuous-backups

次のコード例は、describe-continuous-backups を使用する方法を示しています。

AWS CLI

DynamoDB テーブルの継続的バックアップに関する情報を取得するには

次のdescribe-continuous-backups例では、MusicCollectionテーブルの継続的バックアップ設定の詳細を表示します。

```
aws dynamodb describe-continuous-backups \
  --table-name MusicCollection
```

出力:

```
{
  "ContinuousBackupsDescription": {
    "ContinuousBackupsStatus": "ENABLED",
    "PointInTimeRecoveryDescription": {
      "PointInTimeRecoveryStatus": "DISABLED"
    }
  }
}
```

詳細については、[Point-in-Time](#) 「Amazon DynamoDB デベロッパーガイド」の「DynamoDB の復旧」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeContinuousBackups](#)」の「」を参照してください。

describe-contributor-insights

次のコード例は、describe-contributor-insights を使用する方法を示しています。

AWS CLI

DynamoDB テーブルの Contributor Insights 設定を表示するには

次のdescribe-contributor-insights例は、MusicCollectionテーブルの Contributor Insights 設定とAlbumTitle-indexグローバルセカンダリインデックスを示しています。

```
aws dynamodb describe-contributor-insights \
  --table-name MusicCollection \
  --index-name AlbumTitle-index
```

出力:

```
{
  "TableName": "MusicCollection",
  "IndexName": "AlbumTitle-index",
  "ContributorInsightsRuleList": [
    "DynamoDBContributorInsights-PKC-MusicCollection-1576629651520",
    "DynamoDBContributorInsights-SKC-MusicCollection-1576629651520",
    "DynamoDBContributorInsights-PKT-MusicCollection-1576629651520",
```

```
    "DynamoDBContributorInsights-SKT-MusicCollection-1576629651520"  
  ],  
  "ContributorInsightsStatus": "ENABLED",  
  "LastUpdateDateTime": 1576629654.78  
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の CloudWatch 「Contributor Insights for DynamoDB を使用したデータアクセスの分析」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeContributorInsights](#)」の「」を参照してください。

describe-endpoints

次のコード例は、describe-endpoints を使用する方法を示しています。

AWS CLI

リージョンエンドポイント情報を表示するには

次のdescribe-endpoints例では、現在の AWS リージョンのエンドポイントの詳細を表示します。

```
aws dynamodb describe-endpoints
```

出力:

```
{  
  "Endpoints": [  
    {  
      "Address": "dynamodb.us-west-2.amazonaws.com",  
      "CachePeriodInMinutes": 1440  
    }  
  ]  
}
```

詳細については、AWS 全般のリファレンスの[「Amazon DynamoDB エンドポイントとクォータ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEndpoints](#)」の「」を参照してください。

describe-global-table-settings

次のコード例は、describe-global-table-settings を使用する方法を示しています。

AWS CLI

DynamoDB グローバルテーブルの設定に関する情報を取得するには

次のdescribe-global-table-settings例では、MusicCollectionグローバルテーブルの設定を表示します。

```
aws dynamodb describe-global-table-settings \  
--global-table-name MusicCollection
```

出力:

```
{  
  "GlobalTableName": "MusicCollection",  
  "ReplicaSettings": [  
    {  
      "RegionName": "us-east-1",  
      "ReplicaStatus": "ACTIVE",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 5,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    },  
    {  
      "RegionName": "us-east-2",  
      "ReplicaStatus": "ACTIVE",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 5,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    }  
  ]  
}
```



```
]
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「 [コマンドリファレンスDescribeGlobalTableSettings](#)」の「」を参照してください。

describe-global-table

次の例は、describe-global-table を使用する方法を説明しています。

AWS CLI

DynamoDB グローバルテーブルに関する情報を表示するには

次のdescribe-global-table例では、MusicCollectionグローバルテーブルの詳細を表示します。

```
aws dynamodb describe-global-table \
  --global-table-name MusicCollection
```

出力:

```
{
  "GlobalTableDescription": {
    "ReplicationGroup": [
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-east-1"
      }
    ],
    "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
    "CreationDateTime": 1576625818.532,
    "GlobalTableStatus": "ACTIVE",
    "GlobalTableName": "MusicCollection"
  }
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGlobalTable](#)」の「」を参照してください。

describe-limits

次の例は、describe-limits を使用する方法を説明しています。

AWS CLI

プロビジョンドキャパシティの制限を表示するには

次のdescribe-limits例では、現在の AWS リージョンのアカウントのプロビジョンドキャパシティ制限を表示します。

```
aws dynamodb describe-limits
```

出力:

```
{
  "AccountMaxReadCapacityUnits": 80000,
  "AccountMaxWriteCapacityUnits": 80000,
  "TableMaxReadCapacityUnits": 40000,
  "TableMaxWriteCapacityUnits": 40000
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB の制限](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLimits](#)」の「」を参照してください。

describe-table-replica-auto-scaling

次のコード例は、describe-table-replica-auto-scaling を使用する方法を示しています。

AWS CLI

グローバルテーブルのレプリカ間で自動スケーリング設定を表示するには

次のdescribe-table-replica-auto-scaling例では、MusicCollectionグローバルテーブルのレプリカ全体の自動スケーリング設定を表示します。

```
aws dynamodb describe-table-replica-auto-scaling \  
  --table-name MusicCollection
```

出力:

```
{  
  "TableAutoScalingDescription": {  
    "TableName": "MusicCollection",  
    "TableStatus": "ACTIVE",  
    "Replicas": [  
      {  
        "RegionName": "us-east-1",  
        "GlobalSecondaryIndexes": [],  
        "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
          "MinimumUnits": 5,  
          "MaximumUnits": 40000,  
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/  
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",  
          "ScalingPolicies": [  
            {  
              "PolicyName": "DynamoDBReadCapacityUtilization:table/  
MusicCollection",  
              "TargetTrackingScalingPolicyConfiguration": {  
                "TargetValue": 70.0  
              }  
            }  
          ]  
        },  
        "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
          "MinimumUnits": 5,  
          "MaximumUnits": 40000,  
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/  
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",  
          "ScalingPolicies": [  
            {  
              "PolicyName": "DynamoDBWriteCapacityUtilization:table/  
MusicCollection",  
              "TargetTrackingScalingPolicyConfiguration": {
```

```

        "TargetValue": 70.0
      }
    }
  ],
},
"ReplicaStatus": "ACTIVE"
},
{
  "RegionName": "us-east-2",
  "GlobalSecondaryIndexes": [],
  "ReplicaProvisionedReadCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 70.0
        }
      }
    ]
  },
  "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 70.0
        }
      }
    ]
  },
  "ReplicaStatus": "ACTIVE"
}
}

```

```
    ]
  }
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeTableReplicaAutoScaling](#)」の「」を参照してください。

describe-table

次の例は、describe-table を使用する方法を説明しています。

AWS CLI

テーブルを記述するには

次の describe-table の例は、MusicCollection テーブルを記述します。

```
aws dynamodb describe-table \  
  --table-name MusicCollection
```

出力:

```
{  
  "Table": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "ProvisionedThroughput": {  
      "NumberOfDecreasesToday": 0,  
      "WriteCapacityUnits": 5,  
      "ReadCapacityUnits": 5  
    },  
    "TableSizeBytes": 0,  
  }  
}
```

```
    "TableName": "MusicCollection",
    "TableStatus": "ACTIVE",
    "KeySchema": [
      {
        "KeyType": "HASH",
        "AttributeName": "Artist"
      },
      {
        "KeyType": "RANGE",
        "AttributeName": "SongTitle"
      }
    ],
    "ItemCount": 0,
    "CreationDateTime": 1421866952.062
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[表の説明](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTable](#)」の「」を参照してください。

describe-time-to-live

次の例は、describe-time-to-live を使用する方法を説明しています。

AWS CLI

テーブルの Time to Live 設定を表示するには

次の describe-time-to-live の例は、MusicCollection テーブルの Time to Live 設定を表示します。

```
aws dynamodb describe-time-to-live \
  --table-name MusicCollection
```

出力:

```
{
  "TimeToLiveDescription": {
    "TimeToLiveStatus": "ENABLED",
```

```
    "AttributeName": "ttl"
  }
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[Time to Live \(TTL\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTimeToLive](#)」の「」を参照してください。

get-item

次の例は、get-item を使用する方法を説明しています。

AWS CLI

例 1: テーブル内の項目を読み込むには

次の get-item の例は、MusicCollection テーブルから項目を取得します。テーブルには hash-and-rangeプライマリキー (Artist と SongTitle) があるため、これらの属性の両方を指定する必要があります。このコマンドは、オペレーションによって消費される読み込み容量に関する情報も要求します。

```
aws dynamodb get-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --return-consumed-capacity TOTAL
```

key.json の内容:

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

出力:

```
{  
  "Item": {  
    "AlbumTitle": {  
      "S": "Songs About Life"    }  
  }  
}
```

```
    },
    "SongTitle": {
      "S": "Happy Day"
    },
    "Artist": {
      "S": "Acme Band"
    }
  },
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 0.5
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目の読み込み](#)」を参照してください。

例 2: 整合性のある読み込みを使用して項目を読み込むには

次の例では、強力な整合性のある読み込みを使用して MusicCollection テーブルから項目を読み込みます。

```
aws dynamodb get-item \
  --table-name MusicCollection \
  --key file://key.json \
  --consistent-read \
  --return-consumed-capacity TOTAL
```

key.json の内容:

```
{
  "Artist": {"S": "Acme Band"},
  "SongTitle": {"S": "Happy Day"}
}
```

出力:

```
{
  "Item": {
    "AlbumTitle": {
      "S": "Songs About Life"
    },
```



```
    "SongTitle": {
      "S": "Happy Day"
    },
    "Artist": {
      "S": "Acme Band"
    }
  },
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 1.0
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目の読み込み](#)」を参照してください。

例 3: 項目の特定の属性を取得するには

次の例は、射影式を使用して目的のアイテムの 3 つの属性のみを取得します。

```
aws dynamodb get-item \
  --table-name ProductCatalog \
  --key '{"Id": {"N": "102"}}' \
  --projection-expression "#T, #C, #P" \
  --expression-attribute-names file://names.json
```

names.json の内容:

```
{
  "#T": "Title",
  "#C": "ProductCategory",
  "#P": "Price"
}
```

出力:

```
{
  "Item": {
    "Price": {
      "N": "20"
    },
    "Title": {
```

```
        "S": "Book 102 Title"
      },
      "ProductCategory": {
        "S": "Book"
      }
    }
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目の読み込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetItem](#)」の「」を参照してください。

list-backups

次のコード例は、list-backups を使用する方法を示しています。

AWS CLI

例 1: 既存の DynamoDB バックアップをすべて一覧表示するには

次のlist-backups例では、既存のバックアップをすべて一覧表示します。

```
aws dynamodb list-backups
```

出力:

```
{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/backup/01234567890123-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    },
  ],
}
```

```
{
  "TableName": "MusicCollection",
  "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
  "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
  "BackupName": "MusicCollectionBackup2",
  "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
  "BackupStatus": "AVAILABLE",
  "BackupType": "USER",
  "BackupSizeBytes": 400
}
]
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

例 2: ユーザーが作成したバックアップを特定の時間範囲内に一覧表示するには

次の例では、2020 年 1 月 1 日から 2020 年 3 月 1 日の間に作成日を持つ、ユーザーによって作成された MusicCollection テーブルのバックアップ (DynamoDB によって自動的に作成されたバックアップではありません) のみを一覧表示します。

```
aws dynamodb list-backups \
  --table-name MusicCollection \
  --time-range-lower-bound 1577836800 \
  --time-range-upper-bound 1583020800 \
  --backup-type USER
```

出力:

```
{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
```

```

        "BackupName": "MusicCollectionBackup1",
        "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
        "BackupStatus": "AVAILABLE",
        "BackupType": "USER",
        "BackupSizeBytes": 170
    }
]
}

```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

例 3: ページサイズを制限するには

次の例では、すべての既存のバックアップのリストを返しますが、呼び出しごとに 1 つの項目のみを取得し、リスト全体を取得するために必要に応じて複数の呼び出しを実行します。デフォルトのページサイズ (1000) を使用して大量のリソースに対してリストコマンドを実行する際に「タイムアウト」が発生する場合、ページサイズを制限してください。

```

aws dynamodb list-backups \
  --page-size 1

```

出力:

```

{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/backup/01234567890123-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    },
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",

```

```
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
    "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-b2abc345",
    "BackupName": "MusicCollectionBackup2",
    "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",
    "BackupStatus": "AVAILABLE",
    "BackupType": "USER",
    "BackupSizeBytes": 400
  }
]
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

例 4: 返される項目の数を制限するには

次の例では、返される項目の数を 1 に制限しています。レスポンスには、次の結果ページの取得に使用する NextToken 値が含まれます。

```
aws dynamodb list-backups \
  --max-items 1
```

出力:

```
{
  "BackupSummaries": [
    {
      "TableName": "MusicCollection",
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01234567890123-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    }
  ],
}
```

```
"NextToken":  
"abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"  
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

例 5: 結果の次のページを取得するには

次のコマンドは、list-backups コマンドへの前の呼び出しの NextToken 値を使用して、次の結果ページを取得します。この場合のレスポンスには NextToken 値が含まれていないため、結果の最後のページに達したことがわかります。

```
aws dynamodb list-backups \  
  --starting-  
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

出力

```
{  
  "BackupSummaries": [  
    {  
      "TableName": "MusicCollection",  
      "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/  
MusicCollection",  
      "BackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/  
MusicCollection/backup/01234567890123-b2abc345",  
      "BackupName": "MusicCollectionBackup2",  
      "BackupCreationDateTime": "2020-06-26T11:08:35.431000-07:00",  
      "BackupStatus": "AVAILABLE",  
      "BackupType": "USER",  
      "BackupSizeBytes": 400  
    }  
  ]  
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBackups](#)」の「」を参照してください。

list-contributor-insights

次の例は、list-contributor-insights を使用する方法を説明しています。

AWS CLI

例 1: Contributor Insights の概要のリストを表示するには

次のlist-contributor-insights例は、Contributor Insights の概要のリストを示しています。

```
aws dynamodb list-contributor-insights
```

出力:

```
{
  "ContributorInsightsSummaries": [
    {
      "TableName": "MusicCollection",
      "IndexName": "AlbumTitle-index",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "ProductCatalog",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Forum",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Reply",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Thread",
      "ContributorInsightsStatus": "ENABLED"
    }
  ]
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の CloudWatch 「Contributor Insights for DynamoDB を使用したデータアクセスの分析」](#)を参照してください。 DynamoDB

例 2: 返される項目の数を制限するには

次の例では、返されるアイテムの数を 4 に制限しています。レスポンスには、次の結果ページの取得に使用する NextToken 値が含まれます。

```
aws dynamodb list-contributor-insights \  
  --max-results 4
```

出力:

```
{  
  "ContributorInsightsSummaries": [  
    {  
      "TableName": "MusicCollection",  
      "IndexName": "AlbumTitle-index",  
      "ContributorInsightsStatus": "ENABLED"  
    },  
    {  
      "TableName": "ProductCatalog",  
      "ContributorInsightsStatus": "ENABLED"  
    },  
    {  
      "TableName": "Forum",  
      "ContributorInsightsStatus": "ENABLED"  
    }  
  ],  
  "NextToken":  
  "abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"  
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の CloudWatch 「Contributor Insights for DynamoDB を使用したデータアクセスの分析」](#)を参照してください。 DynamoDB

例 3: 結果の次のページを取得するには

次のコマンドは、list-contributor-insights コマンドへの前の呼び出しの NextToken 値を使用して、次の結果ページを取得します。この場合のレスポンスには NextToken 値が含まれていないため、結果の最後のページに達したことがわかります。

```
aws dynamodb list-contributor-insights \  
  --max-results 4 \  
  --next-token
```



```
--next-  
token abcDeFGhiJKLmnOPqrSTuvwXYZ1aBCdEFghijk7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

出力:

```
{  
  "ContributorInsightsSummaries": [  
    {  
      "TableName": "Reply",  
      "ContributorInsightsStatus": "ENABLED"  
    },  
    {  
      "TableName": "Thread",  
      "ContributorInsightsStatus": "ENABLED"  
    }  
  ]  
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の CloudWatch 「Contributor Insights for DynamoDB を使用したデータアクセスの分析」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListContributorInsights](#)」の「」を参照してください。

list-global-tables

次のコード例は、list-global-tables を使用する方法を示しています。

AWS CLI

既存の DynamoDB グローバルテーブルを一覧表示するには

次のlist-global-tables例では、既存のすべてのグローバルテーブルを一覧表示します。

```
aws dynamodb list-global-tables
```

出力:

```
{  
  "GlobalTables": [  
    {  
      "GlobalTableName": "MusicCollection",
```

```
    "ReplicationGroup": [  
      {  
        "RegionName": "us-east-2"  
      },  
      {  
        "RegionName": "us-east-1"  
      }  
    ]  
  }  
]
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGlobalTables](#)」の「」を参照してください。

list-tables

次のコード例は、list-tables を使用する方法を示しています。

AWS CLI

例 1: テーブルを一覧表示するには

次のlist-tables例では、現在の AWS アカウントとリージョンに関連付けられているすべてのテーブルを一覧表示します。

```
aws dynamodb list-tables
```

出力:

```
{  
  "TableNames": [  
    "Forum",  
    "ProductCatalog",  
    "Reply",  
    "Thread"  
  ]  
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[テーブル名のリスト化](#)」を参照してください。

例 2: ページサイズを制限するには

次の例は、既存のすべてのテーブルのリストを返しますが、1 回の呼び出しで 1 つの項目のみを取得し、必要な場合は複数の呼び出しを実行してリスト全体を取得します。デフォルトのページサイズ (1000) を使用して大量のリソースに対してリストコマンドを実行する際に「タイムアウト」が発生する場合、ページサイズを制限してください。

```
aws dynamodb list-tables \  
  --page-size 1
```

出力:

```
{  
  "TableNames": [  
    "Forum",  
    "ProductCatalog",  
    "Reply",  
    "Thread"  
  ]  
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[テーブル名のリスト化](#)」を参照してください。

例 3: 返される項目の数を制限するには

次の例は、返される項目の数を 2 に制限します。レスポンスには、次の結果ページの取得に使用する NextToken 値が含まれます。

```
aws dynamodb list-tables \  
  --max-items 2
```

出力:

```
{  
  "TableNames": [  
    "Forum",  
    "ProductCatalog"  
  ]  
}
```

```

    ],
    "NextToken":
    "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
  }

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[テーブル名のリスト化](#)」を参照してください。

例 4: 次の結果ページを取得するには

次のコマンドは、list-tables コマンドへの前の呼び出しの NextToken 値を使用して、次の結果ページを取得します。この場合のレスポンスには NextToken 値が含まれていないため、結果の最後のページに達したことがわかります。

```

aws dynamodb list-tables \
  --starting-
  token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9

```

出力:

```

{
  "TableNames": [
    "Reply",
    "Thread"
  ]
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[テーブル名のリスト化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTables](#)」の「」を参照してください。

list-tags-of-resource

次の例は、list-tags-of-resource を使用する方法を説明しています。

AWS CLI

例 1: DynamoDB リソースのタグを一覧表示するには

次のlist-tags-of-resource例では、MusicCollectionテーブルのタグを表示します。

```
aws dynamodb list-tags-of-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Owner",  
      "Value": "blueTeam"  
    },  
    {  
      "Key": "Environment",  
      "Value": "Production"  
    }  
  ]  
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のタグ付け](#)」を参照してください。 DynamoDB

例 2: 返されるタグの数を制限するには

次の例では、返されるタグの数を 1 に制限しています。レスポンスには、次の結果ページの取得に使用する NextToken 値が含まれます。

```
aws dynamodb list-tags-of-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --max-items 1
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Owner",  
      "Value": "blueTeam"  
    }  
  ],  
  "NextToken":  
  "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFghI2Jk3LmnoPQ6RST9"
```

```
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のタグ付け](#)」を参照してください。 DynamoDB

例 3: 結果の次のページを取得するには

次のコマンドは、list-tags-of-resource コマンドへの前の呼び出しの NextToken 値を使用して、次の結果ページを取得します。この場合のレスポンスには NextToken 値が含まれていないため、結果の最後のページに達したことがわかります。

```
aws dynamodb list-tags-of-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --starting-  
token abCDeFGhiJKLmnOPqrSTuvwXYZ1aBCdEFghijK7LM51n0ppqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Environment",  
      "Value": "Production"  
    }  
  ]  
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のタグ付け](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsOfResource](#)」の「」を参照してください。

put-item

次の例は、put-item を使用する方法を説明しています。

AWS CLI

例 1: テーブルに項目を追加するには

次のput-item例では、MusicCollectionテーブルに新しい項目を追加します。

```
aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item file://item.json \  
  --return-consumed-capacity TOTAL \  
  --return-item-collection-metrics SIZE
```

item.json の内容:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Call Me Today"},  
  "AlbumTitle": {"S": "Greatest Hits"}  
}
```

出力:

```
{  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 1.0  
  },  
  "ItemCollectionMetrics": {  
    "ItemCollectionKey": {  
      "Artist": {  
        "S": "No One You Know"  
      }  
    },  
    "SizeEstimateRangeGB": [  
      0.0,  
      1.0  
    ]  
  }  
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目を書き込みます](#)」を参照してください。

例 2: テーブル内の項目を条件付きで上書きするには

次の put-item の例は、MusicCollection テーブル内の既存の項目に Greatest Hits の値を持つ AlbumTitle 属性がある場合にのみ、その項目を上書きします。このコマンドは、その項目の以前の値を返します。

```
aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item file://item.json \  
  --condition-expression "#A = :A" \  
  --expression-attribute-names file://names.json \  
  --expression-attribute-values file://values.json \  
  --return-values ALL_OLD
```

item.json の内容:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Call Me Today"},  
  "AlbumTitle": {"S": "Somewhat Famous"}  
}
```

names.json の内容:

```
{  
  "#A": "AlbumTitle"  
}
```

values.json の内容:

```
{  
  ":A": {"S": "Greatest Hits"}  
}
```

出力:

```
{  
  "Attributes": {  
    "AlbumTitle": {  
      "S": "Greatest Hits"  
    },  
    "Artist": {  
      "S": "No One You Know"  
    },  
    "SongTitle": {  
      "S": "Call Me Today"  
    }  
  }  
}
```



```
    }  
  }  
}
```

キーが存在する場合は、次のような出力が表示されます。

```
A client error (ConditionalCheckFailedException) occurred when calling the PutItem  
operation: The conditional request failed.
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目を書き込みます](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutItem](#)」の「」を参照してください。

query

次の例は、query を使用する方法を説明しています。

AWS CLI

例 1: テーブルにクエリを実行するには

次の query の例では、MusicCollection テーブルの項目にクエリを実行します。テーブルには hash-and-range プライマリキー (Artist と SongTitle) がありますが、このクエリはハッシュキー値のみを指定します。「No One You Know」という名前のアーティストの曲タイトルが返されます。

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --projection-expression "SongTitle" \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json \  
  --return-consumed-capacity TOTAL
```

expression-attributes.json の内容:

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

出力:

```
{
  "Items": [
    {
      "SongTitle": {
        "S": "Call Me Today"
      },
      "SongTitle": {
        "S": "Scared of My Shadow"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 0.5
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB のクエリオペレーション](#)」を参照してください。

例 2: 強力な整合性のある読み込みを使用してテーブルにクエリを実行し、インデックスを降順で走査するには

次の例では、最初の例と同じクエリを実行しますが、結果は逆の順序で返され、強力な整合性のある読み込みが使用されます。

```
aws dynamodb query \
  --table-name MusicCollection \
  --projection-expression "SongTitle" \
  --key-condition-expression "Artist = :v1" \
  --expression-attribute-values file://expression-attributes.json \
  --consistent-read \
  --no-scan-index-forward \
  --return-consumed-capacity TOTAL
```

expression-attributes.json の内容:

```
{
```

```

    ":v1": {"S": "No One You Know"}
  }

```

出力:

```

{
  "Items": [
    {
      "SongTitle": {
        "S": "Scared of My Shadow"
      }
    },
    {
      "SongTitle": {
        "S": "Call Me Today"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 1.0
  }
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB のクエリオペレーション](#)」を参照してください。

例 3: 特定の結果を除外するには

次の例は、MusicCollection をクエリしますが、AlbumTitle 属性に特定の値を含む結果を除外します。このフィルターは項目が読み込まれた後に適用されるため、ScannedCount または ConsumedCapacity には影響しないことに注意してください。

```

aws dynamodb query \
  --table-name MusicCollection \
  --key-condition-expression "#n1 = :v1" \
  --filter-expression "NOT (#n2 IN (:v2, :v3))" \
  --expression-attribute-names file://names.json \
  --expression-attribute-values file://values.json \
  --return-consumed-capacity TOTAL

```

values.json の内容:

```
{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Blue Sky Blues"},
  ":v3": {"S": "Greatest Hits"}
}
```

names.json の内容:

```
{
  "#n1": "Artist",
  "#n2": "AlbumTitle"
}
```

出力:

```
{
  "Items": [
    {
      "AlbumTitle": {
        "S": "Somewhat Famous"
      },
      "Artist": {
        "S": "No One You Know"
      },
      "SongTitle": {
        "S": "Call Me Today"
      }
    }
  ],
  "Count": 1,
  "ScannedCount": 2,
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 0.5
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB のクエリオペレーション](#)」を参照してください。

例 4: 項目数だけを取得するには

次の例は、クエリに一致する項目数を取得しますが、項目自体は取得しません。

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --select COUNT \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json
```

expression-attributes.json の内容:

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

出力:

```
{  
  "Count": 2,  
  "ScannedCount": 2,  
  "ConsumedCapacity": null  
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB のクエリオペレーション](#)」を参照してください。

例 5: インデックスをクエリするには

次の例は、ローカルセカンダリインデックス AlbumTitleIndex をクエリします。クエリは、ローカルセカンダリインデックスに射影されたベーステーブルのすべての属性を返します。ローカルセカンダリインデックスまたはグローバルセカンダリインデックスをクエリする場合は、table-name パラメータを使用してベーステーブルの名前も指定する必要があることに注意してください。

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --index-name AlbumTitleIndex \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json \  
  --select ALL_PROJECTED_ATTRIBUTES \  
  --expression-attribute-values file://expression-attributes.json
```

```
--return-consumed-capacity INDEXES
```

expression-attributes.json の内容:

```
{
  ":v1": {"S": "No One You Know"}
}
```

出力:

```
{
  "Items": [
    {
      "AlbumTitle": {
        "S": "Blue Sky Blues"
      },
      "Artist": {
        "S": "No One You Know"
      },
      "SongTitle": {
        "S": "Scared of My Shadow"
      }
    },
    {
      "AlbumTitle": {
        "S": "Somewhat Famous"
      },
      "Artist": {
        "S": "No One You Know"
      },
      "SongTitle": {
        "S": "Call Me Today"
      }
    }
  ],
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 0.5,
    "Table": {
      "CapacityUnits": 0.0
    }
  },
}
```

```
    "LocalSecondaryIndexes": {
      "AlbumTitleIndex": {
        "CapacityUnits": 0.5
      }
    }
  }
}
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[DynamoDB のクエリオペレーション](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[クエリ](#)」を参照してください。AWS CLI

restore-table-from-backup

次のコード例は、restore-table-from-backup を使用する方法を示しています。

AWS CLI

既存のバックアップから DynamoDB テーブルを復元するには

次のrestore-table-from-backup例では、既存のバックアップから指定されたテーブルを復元します。

```
aws dynamodb restore-table-from-backup \
  --target-table-name MusicCollection \
  --backup-arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
  backup/01576616366715-b4e58d3a
```

出力:

```
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ]
  }
}
```

```

    }
  ],
  "TableName": "MusicCollection2",
  "KeySchema": [
    {
      "AttributeName": "Artist",
      "KeyType": "HASH"
    },
    {
      "AttributeName": "SongTitle",
      "KeyType": "RANGE"
    }
  ],
  "TableStatus": "CREATING",
  "CreationDateTime": 1576618274.326,
  "ProvisionedThroughput": {
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "TableSizeBytes": 0,
  "ItemCount": 0,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection2",
  "TableId": "114865c9-5ef3-496c-b4d1-c4cbdd2d44fb",
  "BillingModeSummary": {
    "BillingMode": "PROVISIONED"
  },
  "RestoreSummary": {
    "SourceBackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
    "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
    "RestoreDateTime": 1576616366.715,
    "RestoreInProgress": true
  }
}
}

```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のオンデマンドバックアップと復元](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreTableFromBackup](#)」の「」を参照してください。

restore-table-to-point-in-time

次のコード例は、restore-table-to-point-in-time を使用する方法を示しています。

AWS CLI

DynamoDB テーブルを特定の時点に復元するには

次のrestore-table-to-point-in-time例では、MusicCollectionテーブルを指定された時点に復元します。

```
aws dynamodb restore-table-to-point-in-time \  
  --source-table-name MusicCollection \  
  --target-table-name MusicCollectionRestore \  
  --restore-date-time 1576622404.0
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollectionRestore",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "TableStatus": "CREATING",  
    "CreationDateTime": 1576623311.86,  
  }  
}
```

```

    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollectionRestore",
    "TableId": "befd9e0e-1843-4dc6-a147-d6d00e85cb1f",
    "BillingModeSummary": {
      "BillingMode": "PROVISIONED"
    },
    "RestoreSummary": {
      "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "RestoreDateTime": 1576622404.0,
      "RestoreInProgress": true
    }
  }
}

```

詳細については、「Amazon [Point-in-Time DynamoDB デベロッパーガイド](#)」の「[DynamoDB の復旧](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreTableToPointInTime](#)」の「」を参照してください。

scan

次のコード例は、scan を使用する方法を示しています。

AWS CLI

テーブルをスキャンするには

次の scan の例は、MusicCollection テーブル全体をスキャンし、その結果をアーティスト「No One You Know」の曲に絞り込みます。各項目について、アルバムタイトルと曲タイトルのみが返されます。

```

aws dynamodb scan \
  --table-name MusicCollection \
  --filter-expression "Artist = :a" \

```

```
--projection-expression "#ST, #AT" \  
--expression-attribute-names file://expression-attribute-names.json \  
--expression-attribute-values file://expression-attribute-values.json
```

expression-attribute-names.json の内容:

```
{  
  "#ST": "SongTitle",  
  "#AT": "AlbumTitle"  
}
```

expression-attribute-values.json の内容:

```
{  
  ":a": {"S": "No One You Know"}  
}
```

出力:

```
{  
  "Count": 2,  
  "Items": [  
    {  
      "SongTitle": {  
        "S": "Call Me Today"  
      },  
      "AlbumTitle": {  
        "S": "Somewhat Famous"  
      }  
    },  
    {  
      "SongTitle": {  
        "S": "Scared of My Shadow"  
      },  
      "AlbumTitle": {  
        "S": "Blue Sky Blues"  
      }  
    }  
  ],  
  "ScannedCount": 3,  
  "ConsumedCapacity": null  
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[DynamoDB でのスキャンの使用](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[スキャン](#)」を参照してください。AWS CLI

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

DynamoDB リソースにタグを追加するには

次のtag-resource例では、タグキーと値のペアをMusicCollectionテーブルに追加します。

```
aws dynamodb tag-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --tags Key=Owner,Value=blueTeam
```

このコマンドでは何も出力されません。

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のタグ付け](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

transact-get-items

次のコード例は、transact-get-items を使用する方法を示しています。

AWS CLI

1 つ以上のテーブルから複数の項目をアトミックに取得するには

次のtransact-get-items例では、複数の項目をアトミックに取得します。

```
aws dynamodb transact-get-items \  
  --transact-items file://transact-items.json \  
  --return-consumed-capacity TOTAL
```

transact-items.json の内容:

```
[
  {
    "Get": {
      "Key": {
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"}
      },
      "TableName": "MusicCollection"
    }
  },
  {
    "Get": {
      "Key": {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"}
      },
      "TableName": "MusicCollection"
    }
  }
]
```

出力:

```
{
  "ConsumedCapacity": [
    {
      "TableName": "MusicCollection",
      "CapacityUnits": 4.0,
      "ReadCapacityUnits": 4.0
    }
  ],
  "Responses": [
    {
      "Item": {
        "AlbumTitle": {
          "S": "Songs About Life"
        },
        "Artist": {
          "S": "Acme Band"
        },
        "SongTitle": {
```

```
        "S": "Happy Day"
      }
    },
    {
      "Item": {
        "AlbumTitle": {
          "S": "Somewhat Famous"
        },
        "Artist": {
          "S": "No One You Know"
        },
        "SongTitle": {
          "S": "Call Me Today"
        }
      }
    }
  ]
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の「DynamoDB トランザクションによる複雑なワークフローの管理」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[TransactGetItems](#)」の「」を参照してください。

transact-write-items

次の例は、transact-write-items を使用する方法を説明しています。

AWS CLI

例 1: 1 つ以上のテーブルに項目をアトミックに書き込むには

次のtransact-write-items例では、1 つの項目を更新し、別の項目を削除します。いずれかの操作が失敗した場合、またはいずれかの項目に Rating 属性が含まれている場合、操作は失敗します。

```
aws dynamodb transact-write-items \  
  --transact-items file://transact-items.json \  
  --return-consumed-capacity TOTAL \  
  --return-item-collection-metrics SIZE
```

transact-items.json ファイルの内容 :

```
[
  {
    "Update": {
      "Key": {
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"}
      },
      "UpdateExpression": "SET AlbumTitle = :newval",
      "ExpressionAttributeValues": {
        ":newval": {"S": "Updated Album Title"}
      },
      "TableName": "MusicCollection",
      "ConditionExpression": "attribute_not_exists(Rating)"
    }
  },
  {
    "Delete": {
      "Key": {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"}
      },
      "TableName": "MusicCollection",
      "ConditionExpression": "attribute_not_exists(Rating)"
    }
  }
]
```

出力:

```
{
  "ConsumedCapacity": [
    {
      "TableName": "MusicCollection",
      "CapacityUnits": 10.0,
      "WriteCapacityUnits": 10.0
    }
  ],
  "ItemCollectionMetrics": {
    "MusicCollection": [
      {
        "ItemCollectionKey": {
```

```

        "Artist": {
            "S": "No One You Know"
        }
    },
    "SizeEstimateRangeGB": [
        0.0,
        1.0
    ]
},
{
    "ItemCollectionKey": {
        "Artist": {
            "S": "Acme Band"
        }
    },
    "SizeEstimateRangeGB": [
        0.0,
        1.0
    ]
}
]
}
}

```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の「DynamoDB トランザクションによる複雑なワークフローの管理」](#)を参照してください。 DynamoDB

例 2: クライアントリクエストトークンを使用して項目をアトミックに書き込むには

次のコマンドは、クライアントリクエストトークンを使用して idempotent `transact-write-items` を呼び出します。つまり、複数の呼び出しは 1 回の呼び出しと同じ効果を持ちます。

```

aws dynamodb transact-write-items \
  --transact-items file://transact-items.json \
  --client-request-token abc123

```

`transact-items.json` ファイルの内容 :

```

[
  {
    "Update": {
      "Key": {

```



```
        "Artist": {"S": "Acme Band"},
        "SongTitle": {"S": "Happy Day"}
    },
    "UpdateExpression": "SET AlbumTitle = :newval",
    "ExpressionAttributeValues": {
        ":newval": {"S": "Updated Album Title"}
    },
    "TableName": "MusicCollection",
    "ConditionExpression": "attribute_not_exists(Rating)"
}
},
{
    "Delete": {
        "Key": {
            "Artist": {"S": "No One You Know"},
            "SongTitle": {"S": "Call Me Today"}
        },
        "TableName": "MusicCollection",
        "ConditionExpression": "attribute_not_exists(Rating)"
    }
}
]
```

このコマンドでは何も出力されません。

詳細については、[「Amazon DynamoDB デベロッパーガイド」の「DynamoDB トランザクションによる複雑なワークフローの管理」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[TransactWriteItems](#)」の「」を参照してください。

untag-resource

次の例は、`untag-resource` を使用する方法を説明しています。

AWS CLI

DynamoDB リソースからタグを削除するには

次の`untag-resource`例では、キーを持つ タグをMusicCollectionテーブルOwnerから削除します。

```
aws dynamodb untag-resource \
```

```
--resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
--tag-keys Owner
```

このコマンドでは何も出力されません。

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB のタグ付け](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「[コマンドリファレンスUntagResource](#)」の「」を参照してください。

update-continuous-backups

次のコード例は、update-continuous-backups を使用する方法を示しています。

AWS CLI

DynamoDB テーブルの継続的なバックアップ設定を更新するには

次のupdate-continuous-backups例では、MusicCollectionテーブルのリカバリを有効にします point-in-time。

```
aws dynamodb update-continuous-backups \  
--table-name MusicCollection \  
--point-in-time-recovery-specification PointInTimeRecoveryEnabled=true
```

出力:

```
{  
  "ContinuousBackupsDescription": {  
    "ContinuousBackupsStatus": "ENABLED",  
    "PointInTimeRecoveryDescription": {  
      "PointInTimeRecoveryStatus": "ENABLED",  
      "EarliestRestorableDateTime": 1576622404.0,  
      "LatestRestorableDateTime": 1576622404.0  
    }  
  }  
}
```

詳細については、[Point-in-Time](#) 「[Amazon DynamoDB デベロッパーガイド](#)」の「[DynamoDB の復旧](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateContinuousBackups](#)」の「」を参照してください。

update-contributor-insights

次のコード例は、update-contributor-insights を使用する方法を示しています。

AWS CLI

テーブルで Contributor Insights を有効にするには

次のupdate-contributor-insights例では、MusicCollectionテーブルとAlbumTitle-indexグローバルセカンダリインデックスで Contributor Insights を有効にします。

```
aws dynamodb update-contributor-insights \  
  --table-name MusicCollection \  
  --index-name AlbumTitle-index \  
  --contributor-insights-action ENABLE
```

出力:

```
{  
  "TableName": "MusicCollection",  
  "IndexName": "AlbumTitle-index",  
  "ContributorInsightsStatus": "ENABLING"  
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の CloudWatch 「Contributor Insights for DynamoDB を使用したデータアクセスの分析」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateContributorInsights](#)」の「」を参照してください。

update-global-table-settings

次のコード例は、update-global-table-settings を使用する方法を示しています。

AWS CLI

DynamoDB グローバルテーブルのプロビジョニングされた書き込み容量設定を更新するには

次のupdate-global-table-settings例では、MusicCollectionグローバルテーブルのプロビジョニングされた書き込み容量を 15 に設定します。

```
aws dynamodb update-global-table-settings \  
  --global-table-name MusicCollection \  
  --global-table-provisioned-write-capacity-units 15
```

出力:

```
{  
  "GlobalTableName": "MusicCollection",  
  "ReplicaSettings": [  
    {  
      "RegionName": "eu-west-1",  
      "ReplicaStatus": "UPDATING",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 10,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    },  
    {  
      "RegionName": "us-east-1",  
      "ReplicaStatus": "UPDATING",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      },  
      "ReplicaProvisionedWriteCapacityUnits": 10,  
      "ReplicaProvisionedWriteCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    },  
    {  
      "RegionName": "us-east-2",  
      "ReplicaStatus": "UPDATING",  
      "ReplicaProvisionedReadCapacityUnits": 10,  
      "ReplicaProvisionedReadCapacityAutoScalingSettings": {  
        "AutoScalingDisabled": true  
      }  
    }  
  ]  
}
```

```

        "ReplicaProvisionedWriteCapacityUnits": 10,
        "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
            "AutoScalingDisabled": true
        }
    }
]
}

```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateGlobalTableSettings](#)」の「」を参照してください。

update-global-table

次のコード例は、update-global-table を使用する方法を示しています。

AWS CLI

DynamoDB グローバルテーブルを更新するには

次のupdate-global-table例では、指定したリージョンのレプリカをMusicCollectionグローバルテーブルに追加します。

```

aws dynamodb update-global-table \
  --global-table-name MusicCollection \
  --replica-updates Create={RegionName=eu-west-1}

```

出力:

```

{
  "GlobalTableDescription": {
    "ReplicationGroup": [
      {
        "RegionName": "eu-west-1"
      },
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-east-1"
      }
    ]
  }
}

```

```
    }
  ],
  "GlobalTableArn": "arn:aws:dynamodb::123456789012:global-table/
MusicCollection",
  "CreationDateTime": 1576625818.532,
  "GlobalTableStatus": "ACTIVE",
  "GlobalTableName": "MusicCollection"
}
}
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateGlobalTable](#)」の「」を参照してください。

update-item

次の例は、update-item を使用する方法を説明しています。

AWS CLI

例 1: テーブル内の項目を更新するには

次の update-item の例では、MusicCollection テーブルの項目を更新します。新しい属性 (Year) を追加して、AlbumTitle 属性を更新します。更新後に表示される項目内の属性はすべて、レスポンスで返されます。

```
aws dynamodb update-item \
  --table-name MusicCollection \
  --key file://key.json \
  --update-expression "SET #Y = :y, #AT = :t" \
  --expression-attribute-names file://expression-attribute-names.json \
  --expression-attribute-values file://expression-attribute-values.json \
  --return-values ALL_NEW \
  --return-consumed-capacity TOTAL \
  --return-item-collection-metrics SIZE
```

key.json の内容:

```
{
```

```
"Artist": {"S": "Acme Band"},
"SongTitle": {"S": "Happy Day"}
}
```

expression-attribute-names.json の内容:

```
{
  "#Y": "Year", "#AT": "AlbumTitle"
}
```

expression-attribute-values.json の内容:

```
{
  ":y": {"N": "2015"},
  ":t": {"S": "Louder Than Ever"}
}
```

出力:

```
{
  "Attributes": {
    "AlbumTitle": {
      "S": "Louder Than Ever"
    },
    "Awards": {
      "N": "10"
    },
    "Artist": {
      "S": "Acme Band"
    },
    "Year": {
      "N": "2015"
    },
    "SongTitle": {
      "S": "Happy Day"
    }
  },
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 3.0
  },
  "ItemCollectionMetrics": {
```

```

    "ItemCollectionKey": {
      "Artist": {
        "S": "Acme Band"
      }
    },
    "SizeEstimateRangeGB": [
      0.0,
      1.0
    ]
  }
}

```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目を書き込みます](#)」を参照してください。

例 2: 項目を条件付きで更新するには

次の例は、既存の項目に Year 属性がない場合にのみ、MusicCollection テーブル内の項目を更新します。

```

aws dynamodb update-item \
  --table-name MusicCollection \
  --key file://key.json \
  --update-expression "SET #Y = :y, #AT = :t" \
  --expression-attribute-names file://expression-attribute-names.json \
  --expression-attribute-values file://expression-attribute-values.json \
  --condition-expression "attribute_not_exists(#Y)"

```

key.json の内容:

```

{
  "Artist": {"S": "Acme Band"},
  "SongTitle": {"S": "Happy Day"}
}

```

expression-attribute-names.json の内容:

```

{
  "#Y": "Year",
  "#AT": "AlbumTitle"
}

```


expression-attribute-values.json の内容:

```
{
  "y":{"N": "2015"},
  "t":{"S": "Louder Than Ever"}
}
```

項目にすでに Year 属性がある場合、DynamoDB は次の出力を返します。

```
An error occurred (ConditionalCheckFailedException) when calling the UpdateItem
operation: The conditional request failed
```

詳細については、「Amazon DynamoDB デイベロッパーガイド」の「[項目を書き込みます](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateItem](#)」の「」を参照してください。

update-table-replica-auto-scaling

次のコード例は、update-table-replica-auto-scaling を使用する方法を示しています。

AWS CLI

グローバルテーブルのレプリカ間で自動スケーリング設定を更新するには

次のupdate-table-replica-auto-scaling例では、指定されたグローバルテーブルのレプリカ間で書き込み容量の自動スケーリング設定を更新します。

```
aws dynamodb update-table-replica-auto-scaling \
  --table-name MusicCollection \
  --provisioned-write-capacity-auto-scaling-update file://auto-scaling-policy.json
```

auto-scaling-policy.json の内容:

```
{
  "MinimumUnits": 10,
  "MaximumUnits": 100,
  "AutoScalingDisabled": false,
  "ScalingPolicyUpdate": {
    "PolicyName": "DynamoDBWriteCapacityUtilization:table/MusicCollection",
```

```

    "TargetTrackingScalingPolicyConfiguration": {
      "TargetValue": 80
    }
  }
}

```

出力:

```

{
  "TableAutoScalingDescription": {
    "TableName": "MusicCollection",
    "TableStatus": "ACTIVE",
    "Replicas": [
      {
        "RegionName": "eu-central-1",
        "GlobalSecondaryIndexes": [],
        "ReplicaProvisionedReadCapacityAutoScalingSettings": {
          "MinimumUnits": 5,
          "MaximumUnits": 40000,
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
          "ScalingPolicies": [
            {
              "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
              "TargetTrackingScalingPolicyConfiguration": {
                "TargetValue": 70.0
              }
            }
          ]
        },
        "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
          "MinimumUnits": 10,
          "MaximumUnits": 100,
          "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
          "ScalingPolicies": [
            {
              "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
              "TargetTrackingScalingPolicyConfiguration": {

```

```
        "TargetValue": 80.0
      }
    ]
  },
  "ReplicaStatus": "ACTIVE"
},
{
  "RegionName": "us-east-1",
  "GlobalSecondaryIndexes": [],
  "ReplicaProvisionedReadCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 70.0
        }
      }
    ]
  },
  "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
    "MinimumUnits": 10,
    "MaximumUnits": 100,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 80.0
        }
      }
    ]
  },
  "ReplicaStatus": "ACTIVE"
},
},
```

```
{
  "RegionName": "us-east-2",
  "GlobalSecondaryIndexes": [],
  "ReplicaProvisionedReadCapacityAutoScalingSettings": {
    "MinimumUnits": 5,
    "MaximumUnits": 40000,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBReadCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 70.0
        }
      }
    ]
  },
  "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
    "MinimumUnits": 10,
    "MaximumUnits": 100,
    "AutoScalingRoleArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable",
    "ScalingPolicies": [
      {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/
MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
          "TargetValue": 80.0
        }
      }
    ]
  },
  "ReplicaStatus": "ACTIVE"
}
]
```

詳細については、「Amazon [DynamoDB デベロッパーガイド](#)」の「[DynamoDB グローバルテーブル](#)」を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateTableReplicaAutoScaling](#)」の「」を参照してください。

update-table

次のコード例は、update-table を使用する方法を示しています。

AWS CLI

例 1: テーブルの請求モードを変更するには

次の update-table 例は、MusicCollection テーブルにプロビジョニングされた読み取り/書き込みキャパシティを増やします。

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --billing-mode PROVISIONED \  
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "AlbumTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      }  
    ]  
  }  
}
```

```

    },
    {
      "AttributeName": "SongTitle",
      "KeyType": "RANGE"
    }
  ],
  "TableStatus": "UPDATING",
  "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
  "ProvisionedThroughput": {
    "LastIncreaseDateTime": "2020-07-28T13:18:18.921000-07:00",
    "NumberOfDecreasesToday": 0,
    "ReadCapacityUnits": 15,
    "WriteCapacityUnits": 10
  },
  "TableSizeBytes": 182,
  "ItemCount": 2,
  "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
  "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
  "BillingModeSummary": {
    "BillingMode": "PROVISIONED",
    "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
  }
}
}
}

```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[テーブルの更新](#)」を参照してください。

例 2: グローバルセカンダリインデックスを作成するには

次の例は、MusicCollection テーブルにグローバルセカンダリインデックスを追加します。

```

aws dynamodb update-table \
  --table-name MusicCollection \
  --attribute-definitions AttributeName=AlbumTitle,AttributeType=S \
  --global-secondary-index-updates file://gsi-updates.json

```

gsi-updates.json の内容:

```

[
  {
    "Create": {
      "IndexName": "AlbumTitle-index",

```

```
    "KeySchema": [  
      {  
        "AttributeName": "AlbumTitle",  
        "KeyType": "HASH"  
      }  
    ],  
    "ProvisionedThroughput": {  
      "ReadCapacityUnits": 10,  
      "WriteCapacityUnits": 10  
    },  
    "Projection": {  
      "ProjectionType": "ALL"  
    }  
  }  
}  
]
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "AlbumTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",
```

```
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "UPDATING",
    "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
    "ProvisionedThroughput": {
      "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 15,
      "WriteCapacityUnits": 10
    },
    "TableSizeBytes": 182,
    "ItemCount": 2,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
    "BillingModeSummary": {
      "BillingMode": "PROVISIONED",
      "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
    },
    "GlobalSecondaryIndexes": [
      {
        "IndexName": "AlbumTitle-index",
        "KeySchema": [
          {
            "AttributeName": "AlbumTitle",
            "KeyType": "HASH"
          }
        ],
        "Projection": {
          "ProjectionType": "ALL"
        },
        "IndexStatus": "CREATING",
        "Backfilling": false,
        "ProvisionedThroughput": {
          "NumberOfDecreasesToday": 0,
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 10
        },
        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
      }
    ]
  ]
}
```



```
}  
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[テーブルの更新](#)」を参照してください。

例 3: テーブルで DynamoDB Streams を有効化するには

次のコマンドは、MusicCollection テーブルで DynamoDB Streams を有効化します。

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE
```

出力:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "AlbumTitle",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
  },  
}
```

```
"TableStatus": "UPDATING",
"CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
"ProvisionedThroughput": {
  "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
  "NumberOfDecreasesToday": 0,
  "ReadCapacityUnits": 15,
  "WriteCapacityUnits": 10
},
"TableSizeBytes": 182,
"ItemCount": 2,
"TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
"TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
"BillingModeSummary": {
  "BillingMode": "PROVISIONED",
  "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
},
"LocalSecondaryIndexes": [
  {
    "IndexName": "AlbumTitleIndex",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "AlbumTitle",
        "KeyType": "RANGE"
      }
    ],
    "Projection": {
      "ProjectionType": "INCLUDE",
      "NonKeyAttributes": [
        "Year",
        "Genre"
      ]
    },
    "IndexSizeBytes": 139,
    "ItemCount": 2,
    "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
  }
],
"GlobalSecondaryIndexes": [
  {
```

```
    "IndexName": "AlbumTitle-index",
    "KeySchema": [
      {
        "AttributeName": "AlbumTitle",
        "KeyType": "HASH"
      }
    ],
    "Projection": {
      "ProjectionType": "ALL"
    },
    "IndexStatus": "ACTIVE",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 10,
      "WriteCapacityUnits": 10
    },
    "IndexSizeBytes": 0,
    "ItemCount": 0,
    "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
  }
],
"StreamSpecification": {
  "StreamEnabled": true,
  "StreamViewType": "NEW_IMAGE"
},
"LatestStreamLabel": "2020-07-28T21:53:39.112",
"LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112"
}
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[テーブルの更新](#)」を参照してください。

例 4: サーバー側の暗号化を有効化するには

次の例は、MusicCollection テーブルでサーバー側の暗号化を有効します。

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --sse-specification Enabled=true,SSEType=KMS
```

出力:

```
{
  "TableDescription": {
    "AttributeDefinitions": [
      {
        "AttributeName": "AlbumTitle",
        "AttributeType": "S"
      },
      {
        "AttributeName": "Artist",
        "AttributeType": "S"
      },
      {
        "AttributeName": "SongTitle",
        "AttributeType": "S"
      }
    ],
    "TableName": "MusicCollection",
    "KeySchema": [
      {
        "AttributeName": "Artist",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "SongTitle",
        "KeyType": "RANGE"
      }
    ],
    "TableStatus": "ACTIVE",
    "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
    "ProvisionedThroughput": {
      "LastIncreaseDateTime": "2020-07-28T12:59:17.537000-07:00",
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 15,
      "WriteCapacityUnits": 10
    },
    "TableSizeBytes": 182,
    "ItemCount": 2,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
    "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
    "BillingModeSummary": {
      "BillingMode": "PROVISIONED",
      "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
    }
  }
}
```

```
    },
    "LocalSecondaryIndexes": [
      {
        "IndexName": "AlbumTitleIndex",
        "KeySchema": [
          {
            "AttributeName": "Artist",
            "KeyType": "HASH"
          },
          {
            "AttributeName": "AlbumTitle",
            "KeyType": "RANGE"
          }
        ],
        "Projection": {
          "ProjectionType": "INCLUDE",
          "NonKeyAttributes": [
            "Year",
            "Genre"
          ]
        },
        "IndexSizeBytes": 139,
        "ItemCount": 2,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitleIndex"
      }
    ],
    "GlobalSecondaryIndexes": [
      {
        "IndexName": "AlbumTitle-index",
        "KeySchema": [
          {
            "AttributeName": "AlbumTitle",
            "KeyType": "HASH"
          }
        ],
        "Projection": {
          "ProjectionType": "ALL"
        },
        "IndexStatus": "ACTIVE",
        "ProvisionedThroughput": {
          "NumberOfDecreasesToday": 0,
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 10
        }
      }
    ]
  }
}
```

```

        },
        "IndexSizeBytes": 0,
        "ItemCount": 0,
        "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/index/AlbumTitle-index"
    }
],
"StreamSpecification": {
    "StreamEnabled": true,
    "StreamViewType": "NEW_IMAGE"
},
"LatestStreamLabel": "2020-07-28T21:53:39.112",
"LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/stream/2020-07-28T21:53:39.112",
"SSEDescription": {
    "Status": "UPDATING"
}
}
}

```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[テーブルの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTable](#)」の「」を参照してください。

update-time-to-live

次の例は、update-time-to-live を使用する方法を説明しています。

AWS CLI

テーブルの Time to Live 設定を更新するには

次の update-time-to-live 例は、指定したテーブルの Time to Live を有効にします。

```

aws dynamodb update-time-to-live \
  --table-name MusicCollection \
  --time-to-live-specification Enabled=true,AttributeName=ttl

```

出力:

```
{
```

```
"TimeToLiveSpecification": {
  "Enabled": true,
  "AttributeName": "ttl"
}
```

詳細については、「Amazon DynamoDB デベロッパーガイド」の「[Time to Live \(TTL\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTimeToLive](#)」の「」を参照してください。

を使用した DynamoDB Streams の例 AWS CLI

次のコード例は、DynamoDB Streams AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-stream

次の例は、describe-stream を使用する方法を説明しています。

AWS CLI

DynamoDB ストリームに関する情報を取得するには

次のdescribe-streamコマンドは、特定の DynamoDB ストリームに関する情報を表示します。

```
aws dynamodbstreams describe-stream \
```

```
--stream-arn arn:aws:dynamodb:us-west-1:123456789012:table/Music/  
stream/2019-10-22T18:02:01.576
```

出力:

```
{  
  "StreamDescription": {  
    "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/  
stream/2019-10-22T18:02:01.576",  
    "StreamLabel": "2019-10-22T18:02:01.576",  
    "StreamStatus": "ENABLED",  
    "StreamViewType": "NEW_AND_OLD_IMAGES",  
    "CreationRequestDateTime": 1571767321.571,  
    "TableName": "Music",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ],  
    "Shards": [  
      {  
        "ShardId": "shardId-00000001571767321804-697ce3d2",  
        "SequenceNumberRange": {  
          "StartingSequenceNumber": "4000000000000642977831",  
          "EndingSequenceNumber": "4000000000000642977831"  
        }  
      },  
      {  
        "ShardId": "shardId-00000001571780995058-40810d86",  
        "SequenceNumberRange": {  
          "StartingSequenceNumber": "75740000000005655171150"  
        },  
        "ParentShardId": "shardId-00000001571767321804-697ce3d2"  
      }  
    ]  
  }  
}
```


詳細については、[「Amazon DynamoDB デベロッパガイド」の「DynamoDB ストリームを使用したテーブルアクティビティのキャプチャ」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStream](#)」の「」を参照してください。

get-records

次のコード例は、get-records を使用する方法を示しています。

AWS CLI

Dynamodb ストリームからレコードを取得するには

次のget-recordsコマンドは、指定された Amazon DynamoDB シャードイテレータを使用してレコードを取得します。

```
aws dynamodbstreams get-records \
  --shard-iterator "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmi0Nmy6xVW4IRcIIbs1z07MNI
+CjNPLqQjnyRSANf0wWmKhL1/KNParWSfz2odf780o00bIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QILOXPc8mRBfNIazfrVCjJK8/jsjCzsQMyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/LEGS05ha1qNP+VL4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5Kpwy+DtcM5BHayBs
+PVYHITaTLiInFLT
+LCwvaz1QH3MY3b8A05Z800wjpkM60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
```

出力:

```
{
  "Records": [
    {
      "eventID": "c3b5d798eef6215d42f8137b19a88e50",
      "eventName": "INSERT",
      "eventVersion": "1.1",
      "eventSource": "aws:dynamodb",
      "awsRegion": "us-west-1",
      "dynamodb": {
        "ApproximateCreationDateTime": 1571849028.0,
        "Keys": {
          "Artist": {
            "S": "No One You Know"
          }
        }
      }
    }
  ]
}
```

```
    },
    "SongTitle": {
      "S": "Call Me Today"
    }
  },
  "NewImage": {
    "AlbumTitle": {
      "S": "Somewhat Famous"
    },
    "Artist": {
      "S": "No One You Know"
    },
    "Awards": {
      "N": "1"
    },
    "SongTitle": {
      "S": "Call Me Today"
    }
  },
  "SequenceNumber": "700000000013256296913",
  "SizeBytes": 119,
  "StreamViewType": "NEW_AND_OLD_IMAGES"
}
},
{
  "eventID": "878960a6967867e2da16b27380a27328",
  "eventName": "INSERT",
  "eventVersion": "1.1",
  "eventSource": "aws:dynamodb",
  "awsRegion": "us-west-1",
  "dynamodb": {
    "ApproximateCreationDateTime": 1571849029.0,
    "Keys": {
      "Artist": {
        "S": "Acme Band"
      },
      "SongTitle": {
        "S": "Happy Day"
      }
    },
    "NewImage": {
      "AlbumTitle": {
        "S": "Songs About Life"
      }
    }
  }
},
```

```
        "Artist": {
            "S": "Acme Band"
        },
        "Awards": {
            "N": "10"
        },
        "SongTitle": {
            "S": "Happy Day"
        }
    },
    "SequenceNumber": "800000000013256297217",
    "SizeBytes": 100,
    "StreamViewType": "NEW_AND_OLD_IMAGES"
}
},
{
    "eventID": "520fabde080e159fc3710b15ee1d4daa",
    "eventName": "MODIFY",
    "eventVersion": "1.1",
    "eventSource": "aws:dynamodb",
    "awsRegion": "us-west-1",
    "dynamodb": {
        "ApproximateCreationDateTime": 1571849734.0,
        "Keys": {
            "Artist": {
                "S": "Acme Band"
            },
            "SongTitle": {
                "S": "Happy Day"
            }
        },
        "NewImage": {
            "AlbumTitle": {
                "S": "Updated Album Title"
            },
            "Artist": {
                "S": "Acme Band"
            },
            "Awards": {
                "N": "10"
            },
            "SongTitle": {
                "S": "Happy Day"
            }
        }
    }
}
```

```

    },
    "OldImage": {
      "AlbumTitle": {
        "S": "Songs About Life"
      },
      "Artist": {
        "S": "Acme Band"
      },
      "Awards": {
        "N": "10"
      },
      "SongTitle": {
        "S": "Happy Day"
      }
    },
    "SequenceNumber": "900000000013256687845",
    "SizeBytes": 170,
    "StreamViewType": "NEW_AND_OLD_IMAGES"
  }
},
"NextShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/stream/2019-10-23T16:41:08.740|1|AAAAAAAAAAAEhEI04jkFLW+LK0wivjT8d/IHEh3iExV2xK00aTxEvVy1C1C7Kbb5+Z0W6bT9VQ2n1/mrs7+PRia0ZCHJu7JHJVW7zlsq0i/ges3fw8GYEymyL+piEk35cx67rQqwKKyq+Q6w9JyjreIOj4F2lWLV26lBwRTrIYC4IB7C3BZZK4715QwYdDxNdVHiSBRZX8UqoS6W0t0F87xZLNB9F/NhYBLXi/wcGvAcBcC0TNI0H+N0Nqwt0B/FGcKNrf8YZ0xRoNN6RgGuVWHF3px0hxEJeFZoSoJTlKeG9YcYxzi5Ci/mhdTm7tBXnbw5c6xmsGsBqTirNjldyJLcWl8C10UOLX63Ufo/5QliztcjEbKsQe28x8LM8o7VH1Is0fF/ITt8awSA4igyJS0P87GN8Qri8kj8iaE35805jBHWf2wvwT6Iy2xGrR2r2HzYps9dwG0arVdEITaJfWzNoL4HajMhmREZ+V04i1YIeHMXJfcwetNRuIbdQXfJht2NQZa4PVV6iknY6d19MrdbSTMkoqAuvp6g3Q2jH4t7GKCLWgodcPAn8g5+43Da"
}

```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の「DynamoDB ストリームを使用したテーブルアクティビティのキャプチャ」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRecords](#)」の「」を参照してください。

get-shard-iterator

次の例は、get-shard-iterator を使用する方法を説明しています。

AWS CLI

シャードイテレータを取得するには

次の `get-shard-iterator` コマンドは、指定されたシャードのシャードイテレータを取得します。

```
aws dynamodbstreams get-shard-iterator \
  --stream-arn arn:aws:dynamodb:us-west-1:12356789012:table/Music/  
stream/2019-10-22T18:02:01.576 \
  --shard-id shardId-00000001571780995058-40810d86 \
  --shard-iterator-type LATEST
```

出力:

```
{
  "ShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmi0Nmy6xVW4IRcIIbs1z07MNI
+CjNPlqQjnyRSAnf0wWmKhL1/KNParWSfz2odf780o00bIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QI10XPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhH+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPI dmTRG
+w/1EGS05ha1qNP+Vl4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw=
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の「DynamoDB ストリームを使用したテーブルアクティビティのキャプチャ」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[GetShardIterator](#)」の「」を参照してください。

list-streams

次の例は、`list-streams` を使用する方法を説明しています。

AWS CLI

DynamoDB ストリームを一覧表示するには

次のlist-streamsコマンドは、デフォルトの AWS リージョン内のすべての既存の Amazon DynamoDB ストリームを一覧表示します。

```
aws dynamodbstreams list-streams
```

出力:

```
{
  "Streams": [
    {
      "StreamArn": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/stream/2019-10-22T18:02:01.576",
      "TableName": "Music",
      "StreamLabel": "2019-10-22T18:02:01.576"
    }
  ]
}
```

詳細については、[「Amazon DynamoDB デベロッパーガイド」の「DynamoDB ストリームを使用したテーブルアクティビティのキャプチャ」](#)を参照してください。 DynamoDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStreams](#)」の「」を参照してください。

を使用した Amazon EC2 の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますEC2。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-address-transfer

次の例は、accept-address-transfer を使用する方法を説明しています。

AWS CLI

アカウントに転送された Elastic IP アドレスを受け入れるには

次のaccept-address-transfer例では、指定された Elastic IP アドレスのアカウントへの転送を受け入れます。

```
aws ec2 accept-address-transfer \  
  --address 100.21.184.216
```

出力:

```
{  
  "AddressTransfer": {  
    "PublicIp": "100.21.184.216",  
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",  
    "TransferAccountId": "123456789012",  
    "TransferOfferExpirationTimestamp": "2023-02-22T20:51:10.000Z",  
    "TransferOfferAcceptedTimestamp": "2023-02-22T22:52:54.000Z",  
    "AddressTransferStatus": "accepted"  
  }  
}
```

詳細については、「Amazon VPCユーザーガイド」の「[Elastic IP アドレスを転送する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptAddressTransfer](#)」の「」を参照してください。

accept-reserved-instances-exchange-quote

次の例は、accept-reserved-instances-exchange-quote を使用する方法を説明しています。

AWS CLI

コンバーティブルリザーブドインスタンス交換を実行するには

この例では、指定されたコンバーティブルリザーブドインスタンスの交換を実行します。

コマンド:

```
aws ec2 accept-reserved-instances-exchange-quote --reserved-  
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-  
configurations OfferingId=b747b472-423c-48f3-8cee-679bcexample
```

出力:

```
{  
  "ExchangeId": "riex-e68ed3c1-8bc8-4c17-af77-811afexample"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AcceptReservedInstancesExchangeQuote](#)」の「」を参照してください。

accept-transit-gateway-peering-attachment

次のコード例は、accept-transit-gateway-peering-attachment を使用する方法を示しています。

AWS CLI

Transit Gateway ピアリングアタッチメントを受け入れるには

次のaccept-transit-gateway-peering-attachment例では、指定された Transit Gateway ピアリングアタッチメントを受け入れます。--region パラメータは、アクセプトトランジットゲートウェイがあるリージョンを指定します。

```
aws ec2 accept-transit-gateway-peering-attachment \  
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \  
  --region us-east-2
```

出力:

```
{
```



```

"TransitGatewayPeeringAttachment": {
  "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbcccd",
  "RequesterTgwInfo": {
    "TransitGatewayId": "tgw-123abc05e04123abc",
    "OwnerId": "123456789012",
    "Region": "us-west-2"
  },
  "AcceptorTgwInfo": {
    "TransitGatewayId": "tgw-11223344aabbcc112",
    "OwnerId": "123456789012",
    "Region": "us-east-2"
  },
  "State": "pending",
  "CreationTime": "2019-12-09T11:38:31.000Z"
}
}

```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ピアリングアタッチメント」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AcceptTransitGatewayPeeringAttachment](#)」の「」を参照してください。

accept-transit-gateway-vpc-attachment

次の例は、accept-transit-gateway-vpc-attachment を使用する方法を説明しています。

AWS CLI

トランジットゲートウェイに をアタッチVPCするリクエストを受け入れるには。

次のaccept-transit-gateway-vpc-attachment例では、指定された添付ファイルを forte リクエストを受け入れます。

```

aws ec2 accept-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE

```

出力:

```

{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",

```

```
"TransitGatewayId": "tgw-0262a0e521EXAMPLE",
"VpcId": "vpc-07e8ffd50fEXAMPLE",
"VpcOwnerId": "123456789012",
"State": "pending",
"SubnetIds": [
  "subnet-0752213d59EXAMPLE"
],
"CreationTime": "2019-07-10T17:33:46.000Z",
"Options": {
  "DnsSupport": "enable",
  "Ipv6Support": "disable"
}
}
```

詳細については、[「Transit Gateways Guide」の「Transit Gateway Attachments to aVPC」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AcceptTransitGatewayVpcAttachment](#)」の「」を参照してください。

accept-vpc-endpoint-connections

次の例は、accept-vpc-endpoint-connections を使用する方法を説明しています。

AWS CLI

インターフェイスエンドポイント接続リクエストを受け入れるには

この例では、指定されたエンドポイントサービスに対して指定されたエンドポイント接続リクエストを受け入れます。

コマンド:

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

出力:

```
{
  "Unsuccessful": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptVpcEndpointConnections](#)」の「」を参照してください。

accept-vpc-peering-connection

次のコード例は、accept-vpc-peering-connection を使用する方法を示しています。

AWS CLI

VPCピアリング接続を受け入れるには

この例では、指定されたVPCピアリング接続リクエストを受け入れます。

コマンド:

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

出力:

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Provisioning",
      "Code": "provisioning"
    },
    "Tags": [],
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-44455566",
      "CidrBlock": "10.0.1.0/28"
    },
    "VpcPeeringConnectionId": "pcx-1a2b3c4d",
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-111abc45",
      "CidrBlock": "10.0.0.0/28"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptVpcPeeringConnection](#)」の「」を参照してください。

advertise-byoip-cidr

次の例は、`advertise-byoip-cidr` を使用する方法を説明しています。

AWS CLI

アドレス範囲をアドバタイズするには

次の`advertise-byoip-cidr`例では、指定されたパブリックIPv4アドレス範囲をアドバタイズします。

```
aws ec2 advertise-byoip-cidr \  
  --cidr 203.0.113.25/24
```

出力:

```
{  
  "ByoipCidr": {  
    "Cidr": "203.0.113.25/24",  
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",  
    "State": "provisioned"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AdvertiseByoipCidr](#)」の「」を参照してください。

allocate-address

次のコード例は、`allocate-address` を使用する方法を示しています。

AWS CLI

例 1: Amazon のアドレスプールから Elastic IP アドレスを割り当てるには

次の `allocate-address` の例では、Elastic IP アドレスを割り当てています。Amazon は、Amazon のアドレスプールからアドレス EC2 を選択します。

```
aws ec2 allocate-address
```

出力:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-01435ba59eEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2",
  "Domain": "vpc"
}
```

詳細については、「Amazon ユーザーガイド」の「[Elastic IP アドレス](#)」を参照してください。
EC2

例 2: Elastic IP アドレスを割り当て、インスタンスまたはネットワークボーダーグループと関連付けるには

次の `allocate-address` の例では、Elastic IP アドレスを割り当て、指定されたネットワークボーダーグループに関連付けます。

```
aws ec2 allocate-address \
  --network-border-group us-west-2-lax-1
```

出力:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-e03dd489ceEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2-lax-1",
  "Domain": "vpc"
}
```

詳細については、「Amazon ユーザーガイド」の「[Elastic IP アドレス](#)」を参照してください。
EC2

例 3: 所有するアドレスプールから Elastic IP アドレスを割り当てるには

次の `allocate-address` の例では、Amazon Web Services アカウントに入れたアドレスプールから Elastic IP アドレスを割り当てています。Amazon は、アドレスプールからアドレス EC2 を選択します。

```
aws ec2 allocate-address \
```

```
--public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

出力:

```
{
  "AllocationId": "eipalloc-02463d08ceEXAMPLE",
  "NetworkBorderGroup": "us-west-2",
  "CustomerOwnedIp": "18.218.95.81",
  "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",
  "Domain": "vpc"
  "NetworkBorderGroup": "us-west-2",
}
```

詳細については、「Amazon ユーザーガイド」の「[Elastic IP アドレス](#)」を参照してください。
EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[AllocateAddress](#)」の「」を参照してください。

allocate-hosts

次の例は、allocate-hosts を使用する方法を説明しています。

AWS CLI

例 1: Dedicated Host を割り当てるには

次のallocate-hosts例では、m5.largeア eu-west-1aベイラビリティゾーンに 1 つの Dedicated Host を割り当て、インスタンスを起動できます。デフォルトでは、Dedicated Host はターゲットインスタンスの起動のみを受け入れ、ホストリカバリをサポートしていません。

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --quantity 1
```

出力:

```
{
  "HostIds": [
    "h-07879acf49EXAMPLE"
  ]
}
```

```
}
```

例 2: 自動配置とホストリカバリが有効になっている Dedicated Host を割り当てるには

次のallocate-hosts例では、自動配置とホストリカバリが有効になっているア eu-west-1a ベイラビリティゾーンに 1 つの Dedicated Host を割り当てます。

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --auto-placement on \  
  --host-recovery on \  
  --quantity 1
```

出力:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

例 3: タグで Dedicated Host を割り当てるには

次のallocate-hosts例では、単一の Dedicated Host を割り当て、 という名前のキーpurposeと の値を持つタグを適用しますproduction。

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --quantity 1 \  
  --tag-specifications 'ResourceType=dedicated-host, Tags={Key=purpose, Value=production}'
```

出力:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「専有ホストの割り当て」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AllocateHosts](#)」の「」を参照してください。

allocate-ipam-pool-cidr

次のコード例は、`allocate-ipam-pool-cidr` を使用する方法を示しています。

AWS CLI

IPAM プールCIDRから を割り当てるには

次の`allocate-ipam-pool-cidr`例では、IPAMプールCIDRから を割り当てます。

(Linux):

```
aws ec2 allocate-ipam-pool-cidr \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --netmask-length 24
```

(Windows):

```
aws ec2 allocate-ipam-pool-cidr ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --netmask-length 24
```

出力:

```
{  
  "IpamPoolAllocation": {  
    "Cidr": "10.0.0.0/24",  
    "IpamPoolAllocationId": "ipam-pool-alloc-018ecc28043b54ba38e2cd99943cebfbfbd",  
    "ResourceType": "custom",  
    "ResourceOwner": "123456789012"  
  }  
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の[「IP アドレス空間を予約するためにプールCIDRに を手動で割り当てる」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AllocatePamPoolCidr](#)」の「」を参照してください。

apply-security-groups-to-client-vpn-target-network

次のコード例は、`apply-security-groups-to-client-vpn-target-network` を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントのターゲットネットワークにセキュリティグループを適用するには

次の`apply-security-groups-to-client-vpn-target-network`例では、指定されたターゲットネットワークとクライアントVPNエンドポイント間の関連付け`sg-01f6e627a89f4db32`にセキュリティグループを適用します。

```
aws ec2 apply-security-groups-to-client-vpn-target-network \
  --security-group-ids sg-01f6e627a89f4db32 \
  --vpc-id vpc-0e2110c2f324332e0 \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

出力:

```
{
  "SecurityGroupIds": [
    "sg-01f6e627a89f4db32"
  ]
}
```

詳細については、クライアント管理者ガイドの「[ターゲットネットワーク](#)」を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス [ApplySecurityGroupsToClientVpnTargetNetwork](#)」の「」を参照してください。

assign-ipv6-addresses

次の例は、`assign-ipv6-addresses` を使用する方法を説明しています。

AWS CLI

ネットワークインターフェイスに特定のIPv6アドレスを割り当てるには

この例では、指定されたIPv6アドレスを指定されたネットワークインターフェイスに割り当てます。

コマンド:

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 2001:db8:1234:1a00:9691:9503:25ad:1761
```

出力:

```
{
  "AssignedIpv6Addresses": [
    "2001:db8:1234:1a00:3304:8879:34cf:4071",
    "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
}
```

Amazon が選択したIPv6アドレスをネットワークインターフェイスに割り当てるには

この例では、指定されたネットワークインターフェイスに2つのIPv6アドレスを割り当てます。Amazon は、サブネットのIPv6CIDRブロック範囲内で使用可能なIPv6アドレスからこれらのIPv6アドレスを自動的に割り当てます。

コマンド:

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-address-count 2
```

出力:

```
{
  "AssignedIpv6Addresses": [
    "2001:db8:1234:1a00:3304:8879:34cf:4071",
    "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス」の[AssignIpv6Addresses](#) を参照してください。

assign-private-ip-addresses

次の例は、`assign-private-ip-addresses` を使用する方法を説明しています。

AWS CLI

特定のセカンダリプライベート IP アドレスをネットワークインターフェイスに割り当てるには

この例では、指定されたセカンダリプライベート IP アドレスを指定されたネットワークインターフェイスに割り当てます。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

Amazon が EC2 選択するセカンダリプライベート IP アドレスをネットワークインターフェイスに割り当てるには

この例では、指定されたネットワークインターフェイスに 2 つのセカンダリプライベート IP アドレスを割り当てます。Amazon は、ネットワークインターフェイスが関連付けられているサブネットの CIDR ブロック範囲内にある使用可能な IP アドレスから、これらの IP アドレス EC2 を自動的に割り当てます。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-private-ip-address-count 2
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssignPrivateIpAddresses](#)」の「」を参照してください。

assign-private-nat-gateway-address

次の例は、`assign-private-nat-gateway-address` を使用する方法を説明しています。

AWS CLI

プライベート IP アドレスをプライベート NAT ゲートウェイに割り当てるには

次の `assign-private-nat-gateway-address` 例では、指定されたプライベート NAT ゲートウェイに 2 つのプライベート IP アドレスを割り当てます。

```
aws ec2 assign-private-nat-gateway-address \
  --nat-gateway-id nat-1234567890abcdef0 \
  --private-ip-address-count 2
```

出力:

```
{
  "NatGatewayId": "nat-1234567890abcdef0",
  "NatGatewayAddresses": [
    {
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
      "IsPrimary": false,
      "Status": "assigning"
    },
    {
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
      "IsPrimary": false,
      "Status": "assigning"
    }
  ]
}
```

詳細については、「Amazon VPC ユーザーガイド」の [NAT「ゲートウェイ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssignPrivateNatGatewayAddress](#)」の「」を参照してください。

associate-address

次のコード例は、`associate-address` を使用する方法を示しています。

AWS CLI

EC2-Classic で Elastic IP アドレスを関連付けるには

この例では、Elastic IP アドレスを EC2-Classical のインスタンスに関連付けます。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 associate-address --instance-id i-07ffe74c7330ebf53 --public-ip 198.51.100.0
```

EC2- で Elastic IP アドレスを関連付けるにはVPC

この例では、Elastic IP アドレスを のインスタンスに関連付けますVPC。

コマンド:

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-id eipalloc-64d5890a
```

出力:

```
{  
  "AssociationId": "eipassoc-2bebb745"  
}
```

この例では、Elastic IP アドレスとネットワークインターフェイスを関連付けています。

コマンド:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d
```

この例では、ネットワークインターフェイスに関連付けられたプライベート IP アドレスに Elastic IP を関連付けています。

コマンド:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d --private-ip-address 10.0.0.85
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateAddress](#)」の「」を参照してください。

associate-client-vpn-target-network

次のコード例は、`associate-client-vpn-target-network` を使用する方法を示しています。

AWS CLI

ターゲットネットワークをクライアントVPNエンドポイントに関連付けるには

次の`associate-client-vpn-target-network`例では、サブネットを指定されたクライアントVPNエンドポイントに関連付けます。

```
aws ec2 associate-client-vpn-target-network \
  --subnet-id subnet-0123456789abcabca \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

出力:

```
{
  "AssociationId": "cvpn-assoc-12312312312312312",
  "Status": {
    "Code": "associating"
  }
}
```

詳細については、クライアント管理者ガイドの「[ターゲットネットワーク](#)」を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateClientVpnTargetNetwork](#)」の「」を参照してください。

associate-dhcp-options

次のコード例は、`associate-dhcp-options` を使用する方法を示しています。

AWS CLI

DHCPオプションセットを に関連付けるには VPC

この例では、指定されたDHCPオプションセットを指定された に関連付けますVPC。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

デフォルトのDHCPオプションセットを に関連付けるには VPC

この例では、デフォルトのDHCPオプションセットを指定された に関連付けますVPC。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateDhcpOptions](#)」の「」を参照してください。

associate-iam-instance-profile

次の例は、associate-iam-instance-profile を使用する方法を説明しています。

AWS CLI

IAMインスタンスプロファイルをインスタンスに関連付けるには

この例では、 という名前のIAMインスタンスプロファイルをインスタンス admin-roleに関連付けますi-123456789abcde123。

コマンド:

```
aws ec2 associate-iam-instance-profile --instance-id i-123456789abcde123 --iam-instance-profile Name=admin-role
```

出力:

```
{
  "IamInstanceProfileAssociation": {
    "InstanceId": "i-123456789abcde123",
    "State": "associating",
    "AssociationId": "iip-assoc-0e7736511a163c209",
    "IamInstanceProfile": {
      "Id": "AIPAJBLK7RKJKWDXVHIEC",
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
    }
  }
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateIamInstanceProfile](#)」の「」を参照してください。

associate-instance-event-window

次の例は、associate-instance-event-window を使用する方法を説明しています。

AWS CLI

例 1: 1 つ以上のインスタンスをイベントウィンドウに関連付けるには

次のassociate-instance-event-window例では、1 つ以上のインスタンスをイベントウィンドウに関連付けます。

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

出力:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [  
        "i-1234567890abcdef0",  
        "i-0598c7d356eba48d7"  
      ],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

イベントウィンドウの制約については、「Amazon ユーザーガイド」の「スケジュールされたイベント」セクションの「[考慮事項](#)」を参照してください。 EC2

例 2: インスタンスタグをイベントウィンドウに関連付けるには

次のassociate-instance-event-window例では、インスタンスタグをイベントウィンドウに関連付けます。instance-event-window-id パラメータを入力して、イベントウィンドウを指定します。インスタンスタグを関連付けるには、association-targetパラメータを指定し、パラメータ値には 1 つ以上のタグを指定します。

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target "InstanceTags=[{Key=k2, Value=v2}, {Key=k1, Value=v1}]"
```

出力:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [],  
      "Tags": [  
        {  
          "Key": "k2",  
          "Value": "v2"  
        },  
        {  
          "Key": "k1",  
          "Value": "v1"  
        }  
      ],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

イベントウィンドウの制約については、「Amazon EC2ユーザーガイド」の「スケジュールされたイベント」セクションの[「考慮事項」](#)を参照してください。

例 3: Dedicated Host をイベントウィンドウに関連付けるには

次のassociate-instance-event-window例では、Dedicated Host をイベントウィンドウに関連付けます。instance-event-window-id パラメータを入力して、イベントウィンドウを指定します。Dedicated Host を関連付けるには、--association-targetパラメータを指定し、パラメータ値には 1 つ以上の Dedicated Host を指定しますIDs。

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

出力:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [],  
      "Tags": [],  
      "DedicatedHostIds": [  
        "h-029fa35a02b99801d"  
      ]  
    },  
    "State": "creating"  
  }  
}
```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの「[考慮事項](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateInstanceEventWindow](#)」の「」を参照してください。

associate-ipam-resource-discovery

次のコード例は、associate-ipam-resource-discovery を使用する方法を示しています。

AWS CLI

リソース検出を に関連付けるには IPAM

この例では、委任IPAMされた管理者であり、リソース検出が別の AWS アカウントによって作成および共有されているため、IPAMを使用して、他のアカウントCIDRsが所有するリソースを管理およびモニタリングできます。

注記

このリクエストを完了するには、で取得できるリソース検出 ID [describe-ipam-resource-discoveries](#)と [describe-ipams](#) で取得できる IPAM ID が必要です。関連付けるリソース検出は、まずを使用してアカウントと共有されている必要があります AWS RAM。入力--regionする IPAMは、関連付ける のホームリージョンと一致する必要があります。

次のassociate-ipam-resource-discovery例では、リソース検出を に関連付けます IPAM。

```
aws ec2 associate-ipam-resource-discovery \
  --ipam-id ipam-005f921c17ebd5107 \
  --ipam-resource-discovery-id ipam-res-disco-03e0406de76a044ee \
  --tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-center,Value=cc123}]' \
  --region us-east-1
```

出力:

```
{
  {
    "IpamResourceDiscoveryAssociation": {
      "OwnerId": "320805250157",
      "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
      "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-
resource-discovery-association/ipam-res-disco-assoc-04382a6346357cf82",
      "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
      "IpamId": "ipam-005f921c17ebd5107",
      "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
      "IpamRegion": "us-east-1",
      "IsDefault": false,
      "ResourceDiscoveryStatus": "active",
      "State": "associate-in-progress",
      "Tags": []
    }
  }
}
```

リソース検出を関連付けると、他のアカウントによって作成されたリソースの IP アドレスをモニタリングおよび/または管理できます。詳細については、「Amazon VPC IPAM ユーザーガイド」の「[組織外のアカウント IPAM との統合](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateIpamResourceDiscovery](#)」の「」を参照してください。

associate-nat-gateway-address

次のコード例は、associate-nat-gateway-address を使用する方法を示しています。

AWS CLI

Elastic IP アドレスをパブリック NAT ゲートウェイに関連付けるには

次の associate-nat-gateway-address 例では、指定された Elastic IP アドレスを指定されたパブリック NAT ゲートウェイに関連付けます。はセカンダリプライベート IPv4 アドレス AWS を自動的に割り当てます。

```
aws ec2 associate-nat-gateway-address \  
  --nat-gateway-id nat-1234567890abcdef0 \  
  --allocation-ids eipalloc-0be6ecac95EXAMPLE
```

出力:

```
{  
  "NatGatewayId": "nat-1234567890abcdef0",  
  "NatGatewayAddresses": [  
    {  
      "AllocationId": "eipalloc-0be6ecac95EXAMPLE",  
      "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
      "IsPrimary": false,  
      "Status": "associating"  
    }  
  ]  
}
```

詳細については、「Amazon VPC ユーザーガイド [NAT](#)」の「[ゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateNatGatewayAddress](#)」の「」を参照してください。

associate-route-table

次のコード例は、associate-route-table を使用する方法を示しています。

AWS CLI

ルートテーブルをサブネットに関連付けるには

この例では、指定されたルートテーブルを指定されたサブネットに関連付けます。

コマンド:

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-id subnet-9d4a7b6c
```

出力:

```
{
  "AssociationId": "rtbassoc-781d0d1a"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateRouteTable](#)」の「」を参照してください。

associate-subnet-cidr-block

次のコード例は、associate-subnet-cidr-block を使用する方法を示しています。

AWS CLI

IPv6CIDRブロックをサブネットに関連付けるには

この例では、IPv6CIDRブロックを指定されたサブネットに関連付けます。

コマンド:

```
aws ec2 associate-subnet-cidr-block --subnet-id subnet-5f46ec3b --ipv6-cidr-block 2001:db8:1234:1a00::/64
```

出力:

```
{
  "SubnetId": "subnet-5f46ec3b",
}
```

```
"Ipv6CidrBlockAssociation": {
  "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
  "AssociationId": "subnet-cidr-assoc-3aa54053",
  "Ipv6CidrBlockState": {
    "State": "associating"
  }
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateSubnetCidrBlock](#)」の「」を参照してください。

associate-transit-gateway-multicast-domain

次のコード例は、associate-transit-gateway-multicast-domain を使用する方法を示しています。

AWS CLI

トランジットゲートウェイをマルチキャストドメインに関連付けるには

次のassociate-transit-gateway-multicast-domain例では、指定されたサブネットとアタッチメントを指定されたマルチキャストドメインに関連付けます。

```
aws ec2 associate-transit-gateway-multicast-domain \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
  --transit-gateway-attachment-id tgw-attach-028c1dd0f8f5cbe8e \
  --subnet-ids subnet-000de86e3b49c932a \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

出力:

```
{
  "Associations": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8f5cbe8e",
    "ResourceId": "vpc-01128d2c240c09bd5",
    "ResourceType": "vpc",
    "Subnets": [
      {
        "SubnetId": "subnet-000de86e3b49c932a",
        "State": "associating"
      }
    ]
  }
}
```

```
    }  
  ]  
}  
}
```

詳細については、「Transit Gateways ガイド」の [「マルチキャストドメインの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateTransitGatewayMulticastDomain](#)」の「」を参照してください。

associate-transit-gateway-route-table

次のコード例は、associate-transit-gateway-route-table を使用する方法を示しています。

AWS CLI

トランジットゲートウェイルートテーブルをトランジットゲートウェイアタッチメントに関連付けるには

次の例では、指定された Transit Gateway ルートテーブルを指定された VPC アタッチメントに関連付けます。

```
aws ec2 associate-transit-gateway-route-table \  
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

出力:

```
{  
  "Association": {  
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",  
    "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
    "ResourceId": "vpc-0065acced4EXAMPLE",  
    "ResourceType": "vpc",  
    "State": "associating"  
  }  
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブルの関連付けAWS」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateTransitGatewayRouteTable](#)」の「」を参照してください。

associate-vpc-cidr-block

次の例は、associate-vpc-cidr-block を使用する方法を説明しています。

AWS CLI

例 1: Amazon が提供する IPv6CIDR ブロックを に関連付けるには VPC

次の associate-vpc-cidr-block 例では、IPv6CIDR ブロックを指定された に関連付けます VPC。

```
aws ec2 associate-vpc-cidr-block \  
  --amazon-provided-ipv6-cidr-block \  
  --ipv6-cidr-block-network-border-group us-west-2-lax-1 \  
  --vpc-id vpc-8EXAMPLE
```

出力:

```
{  
  "Ipv6CidrBlockAssociation": {  
    "AssociationId": "vpc-cidr-assoc-0838ce7d9dEXAMPLE",  
    "Ipv6CidrBlockState": {  
      "State": "associating"  
    },  
    "NetworkBorderGroup": "us-west-2-lax-1"  
  },  
  "VpcId": "vpc-8EXAMPLE"  
}
```

例 2: 追加の IPv4CIDR ブロックを に関連付けるには VPC

次の associate-vpc-cidr-block 例では、IPv4CIDR ブロックを指定された 10.2.0.0/16 に関連付けます VPC。

```
aws ec2 associate-vpc-cidr-block \  
  --vpc-id vpc-1EXAMPLE \  
  --cidr-block 10.2.0.0/16
```


出力:

```
{
  "CidrBlockAssociation": {
    "AssociationId": "vpc-cidr-assoc-2EXAMPLE",
    "CidrBlock": "10.2.0.0/16",
    "CidrBlockState": {
      "State": "associating"
    }
  },
  "VpcId": "vpc-1EXAMPLE"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateVpcCidrBlock](#)」の「」を参照してください。

attach-classic-link-vpc

次の例は、attach-classic-link-vpc を使用する方法を説明しています。

AWS CLI

EC2-Classic インスタンスを にリンク (アタッチ) するには VPC

この例では、VPCセキュリティグループ sg-1234567890 を介してインスタンス i-88888888abcdef0 を VPC vpc-12312312 にリンクします。

コマンド:

```
aws ec2 attach-classic-link-vpc --instance-id i-1234567890abcdef0 --vpc-id vpc-88888888 --groups sg-12312312
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachClassicLinkVpc](#)」の「」を参照してください。

attach-internet-gateway

次の例は、attach-internet-gateway を使用する方法を説明しています。

AWS CLI

にインターネットゲートウェイをアタッチするには VPC

次のattach-internet-gateway例では、指定されたインターネットゲートウェイを特定の にアタッチしますVPC。

```
aws ec2 attach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「Amazon VPCユーザーガイド」の「[インターネットゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachInternetGateway](#)」の「」を参照してください。

attach-network-interface

次の例は、attach-network-interface を使用する方法を説明しています。

AWS CLI

例 1: ネットワークインターフェイスをインスタンスにアタッチするには

次のattach-network-interface例では、指定されたネットワークインターフェイスを指定されたインスタンスにアタッチします。

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-0dc56a8d4640ad10a \  
  --instance-id i-1234567890abcdef0 \  
  --device-index 1
```

出力:

```
{  
  "AttachmentId": "eni-attach-01a8fc87363f07cf9"
```

```
}
```

詳細については、「Amazon ユーザーガイド」の「[Elastic Network Interface](#)」を参照してください。 EC2

例 2: ネットワークインターフェイスを複数のネットワークカードを持つインスタンスにアタッチするには

次のattach-network-interface例では、指定されたネットワークインターフェイスを指定されたインスタンスとネットワークカードにアタッチします。

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-07483b1897541ad83 \  
  --instance-id i-01234567890abcdef \  
  --network-card-index 1 \  
  --device-index 1
```

出力:

```
{  
  "AttachmentId": "eni-attach-0fbd7ee87a88cd06c"  
}
```

詳細については、「Amazon ユーザーガイド」の「[Elastic Network Interface](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachNetworkInterface](#)」の「」を参照してください。

attach-verified-access-trust-provider

次のコード例は、attach-verified-access-trust-provider を使用する方法を示しています。

AWS CLI

信頼プロバイダーをインスタンスにアタッチするには

次のattach-verified-access-trust-provider例では、指定された Verified Access 信頼プロバイダーを指定された Verified Access インスタンスにアタッチします。

```
aws ec2 attach-verified-access-trust-provider \  

```

```
--verified-access-instance-id vai-0ce000c0b7643abea \  
--verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

出力:

```
{  
  "VerifiedAccessTrustProvider": {  
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
    "Description": "",  
    "TrustProviderType": "user",  
    "UserTrustProviderType": "iam-identity-center",  
    "PolicyReferenceName": "idc",  
    "CreationTime": "2023-08-25T19:00:38",  
    "LastUpdatedTime": "2023-08-25T19:00:38"  
  },  
  "VerifiedAccessInstance": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "",  
    "VerifiedAccessTrustProviders": [  
      {  
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
        "TrustProviderType": "user",  
        "UserTrustProviderType": "iam-identity-center"  
      }  
    ],  
    "CreationTime": "2023-08-25T18:27:56",  
    "LastUpdatedTime": "2023-08-25T18:27:56"  
  }  
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access インスタンス」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachVerifiedAccessTrustProvider](#)」の「」を参照してください。

attach-volume

次のコード例は、attach-volume を使用する方法を示しています。

AWS CLI

ボリュームをインスタンスにアタッチするには

この例では、ボリューム (vol-1234567890abcdef0) をとしてインスタンス (i-01474ef662b89480) にアタッチします/dev/sdf。

コマンド:

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-id i-01474ef662b89480 --device /dev/sdf
```

出力:

```
{
  "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
  "InstanceId": "i-01474ef662b89480",
  "VolumeId": "vol-1234567890abcdef0",
  "State": "attaching",
  "Device": "/dev/sdf"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachVolume](#)」の「」を参照してください。

attach-vpn-gateway

次のコード例は、attach-vpn-gateway を使用する方法を示しています。

AWS CLI

仮想プライベートゲートウェイを にアタッチするには VPC

次のattach-vpn-gateway例では、指定された仮想プライベートゲートウェイを指定された にアタッチしますVPC。

```
aws ec2 attach-vpn-gateway \
  --vpn-gateway-id vgw-9a4cacf3 \
  --vpc-id vpc-a01106c2
```

出力:

```
{
  "VpcAttachment": {
```

```
    "State": "attaching",
    "VpcId": "vpc-a01106c2"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachVpnGateway](#)」の「」を参照してください。

authorize-client-vpn-ingress

次の例は、authorize-client-vpn-ingress を使用する方法を説明しています。

AWS CLI

クライアントVPNエンドポイントの承認ルールを追加するには

次のauthorize-client-vpn-ingress例では、すべてのクライアントがインターネット () にアクセスすることを許可する進入認証ルールを追加します0.0.0.0/0。

```
aws ec2 authorize-client-vpn-ingress \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --target-network-cidr 0.0.0.0/0 \
  --authorize-all-groups
```

出力:

```
{
  "Status": {
    "Code": "authorizing"
  }
}
```

詳細については、「クライアント管理者ガイド」の「[認証ルール](#)」を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[AuthorizeClientVpnIngress](#)」の「」を参照してください。

authorize-security-group-egress

次の例は、authorize-security-group-egress を使用する方法を説明しています。

AWS CLI

特定のアドレス範囲へのアウトバウンドトラフィックを許可するルールを追加するには

このコマンド例では、TCPポート 80 で指定されたアドレス範囲へのアクセスを許可するルールを追加します。

コマンド (Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges='[{"CidrIp=10.0.0.0/16}]'
```

コマンド (Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{"CidrIp=10.0.0.0/16}]
```

特定のセキュリティグループへのアウトバウンドトラフィックを許可するルールを追加するには

このコマンド例では、TCPポート 80 で指定されたセキュリティグループへのアクセスを許可するルールを追加します。

コマンド (Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs='[{"GroupId=sg-4b51a32f}]'
```

コマンド (Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{"GroupId=sg-4b51a32f}]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AuthorizeSecurityGroupEgress](#)」の「」を参照してください。

authorize-security-group-ingress

次の例は、authorize-security-group-ingress を使用する方法を説明しています。

AWS CLI

例 1: インバウンドSSHトラフィックを許可するルールを追加するには

次のauthorize-security-group-ingress例では、TCPポート 22 () でインバウンドトラフィックを許可するルールを追加しますSSH。

```
aws ec2 authorize-security-group-ingress \  
  --group-id sg-1234567890abcdef0 \  
  --protocol tcp \  
  --port 22 \  
  --cidr 203.0.113.0/24
```

出力:

```
{  
  "Return": true,  
  "SecurityGroupRules": [  
    {  
      "SecurityGroupRuleId": "sgr-01afa97ef3e1bedfc",  
      "GroupId": "sg-1234567890abcdef0",  
      "GroupOwnerId": "123456789012",  
      "IsEgress": false,  
      "IpProtocol": "tcp",  
      "FromPort": 22,  
      "ToPort": 22,  
      "CidrIpv4": "203.0.113.0/24"  
    }  
  ]  
}
```

例 2: 別のセキュリティグループからのインバウンドHTTPトラフィックを許可するルールを追加するには

次のauthorize-security-group-ingress例では、ソースセキュリティグループからのTCPポート 80 でのインバウンドアクセスを許可するルールを追加しますsg-1a2b3c4d。ソースグループは、同じ VPCまたはピアにある必要があります VPC (VPCピアリング接続が必要です)。着信トラフィックは、ソースセキュリティグループに関連付けられたインスタンスのプライベート IP アドレスに基づいて許可されます (パブリック IP アドレスまたは Elastic IP アドレスは考慮されません)。

```
aws ec2 authorize-security-group-ingress \  
  --group-id sg-1a2b3c4d \  
  --protocol tcp \  
  --port 80 \  
  --source-security-group-id sg-1a2b3c4d
```



```
--group-id sg-1234567890abcdef0 \  
--protocol tcp \  
--port 80 \  
--source-group sg-1a2b3c4d
```

出力:

```
{  
  "Return": true,  
  "SecurityGroupRules": [  
    {  
      "SecurityGroupRuleId": "sgr-01f4be99110f638a7",  
      "GroupId": "sg-1234567890abcdef0",  
      "GroupOwnerId": "123456789012",  
      "IsEgress": false,  
      "IpProtocol": "tcp",  
      "FromPort": 80,  
      "ToPort": 80,  
      "ReferencedGroupInfo": {  
        "GroupId": "sg-1a2b3c4d",  
        "UserId": "123456789012"  
      }  
    }  
  ]  
}
```

例 3: 同じ呼び出しに複数のルールを追加するには

次のauthorize-security-group-ingress例では、ip-permissionsパラメータを使用して2つのインバウンドルールを追加します。1つはTCPポート 3389 (RDP) でのインバウンドアクセスを有効にし、もう1つは ping/ を有効にしますICMP。

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions  
IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges 「[{{CidrIp=172.31.0.0/16}}]  
IpProtocol 「=icmp,FromPort=-1,ToPort=-1,IpRanges」 、 「[{{CidrIp=172.31.0.0/16}}]」
```

出力:

```
{  
  "Return": true,  
  "SecurityGroupRules": [  
    {
```

```

    "SecurityGroupRuleId": "sgr-00e06e5d3690f29f3",
    "GroupId": "sg-1234567890abcdef0",
    "GroupOwnerId": "123456789012",
    "IsEgress": false,
    "IpProtocol": "tcp",
    "FromPort": 3389,
    "ToPort": 3389,
    "CidrIpv4": "172.31.0.0/16"
  },
  {
    "SecurityGroupRuleId": "sgr-0a133dd4493944b87",
    "GroupId": "sg-1234567890abcdef0",
    "GroupOwnerId": "123456789012",
    "IsEgress": false,
    "IpProtocol": "tcp",
    "FromPort": -1,
    "ToPort": -1,
    "CidrIpv4": "172.31.0.0/16"
  }
]
}

```

例 4: ICMPトラフィックのルールを追加するには

次のauthorize-security-group-ingress例では、ip-permissionsパラメータを使用して、どこからでもICMPメッセージ Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (タイプ 3、コード 4) を許可するインバウンドルールを追加します。

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges [{"CidrIp=0.0.0.0/0}]
```

出力:

```

{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0de3811019069b787",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmp",

```

```
        "FromPort": 3,  
        "ToPort": 4,  
        "CidrIpv4": "0.0.0.0/0"  
    }  
]  
}
```

例 5: IPv6トラフィックのルールを追加するには

次のauthorize-security-group-ingress例では、ip-permissionsパラメータを使用して、IPv6範囲からのSSHアクセス(ポート 22)を許可するインバウンドルールを追加します2001:db8:1234:1a00::/64。

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions  
IpProtocol=tcp,FromPort=22,ToPort=22,Ipv6Ranges 「{CidrIpv6=2001:db8:1234:1a00::/64}」
```

出力:

```
{  
  "Return": true,  
  "SecurityGroupRules": [  
    {  
      "SecurityGroupId": "sgr-0455bc68b60805563",  
      "GroupId": "sg-1234567890abcdef0",  
      "GroupOwnerId": "123456789012",  
      "IsEgress": false,  
      "IpProtocol": "tcp",  
      "FromPort": 22,  
      "ToPort": 22,  
      "CidrIpv6": "2001:db8:1234:1a00::/64"  
    }  
  ]  
}
```

例 6: ICMPv6トラフィックのルールを追加するには

次のauthorize-security-group-ingress例では、ip-permissionsパラメータを使用して、どこからでもICMPv6トラフィックを許可するインバウンドルールを追加します。

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions  
IpProtocol=icmpv6,Ipv6Ranges 「{CidrIpv6=::/0}」
```

出力:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmpv6",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv6": ":::/0"
    }
  ]
}
```

例 7: 説明付きのルールを追加する

次の `authorize-security-group-ingress` 例では、`ip-permissions` パラメータを使用して、指定された IPv4 アドレス範囲からの RDP トラフィックを許可するインバウンドルールを追加します。ルールには、後で特定できるように説明が含まれます。

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges 「{CidrIp=203.0.113.0/24,Description='RDP
access from NY office'}」
```

出力:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0397bbcc01e974db3",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 3389,
      "ToPort": 3389,
      "CidrIpv4": "203.0.113.0/24",
    }
  ]
}
```

```
        "Description": "RDP access from NY office"
      }
    ]
  }
}
```

例 8: プレフィックスリストを使用するインバウンドルールを追加するには

次のauthorize-security-group-ingress例では、ip-permissionsパラメータを使用して、指定されたプレフィックスリストCIDRの範囲のすべてのトラフィックを許可するインバウンドルールを追加します。

```
aws ec2 authorize-security-group-ingress --group-id sg-04a351bfe432d4e71 --ip-permissions
IpProtocol=all,PrefixListIds [{"PrefixListId=pl-002dc3ec097de1514}]
```

出力:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "PrefixListId": "pl-0721453c7ac4ec009"
    }
  ]
}
```

詳細については、「Amazon ユーザーガイド」の [「セキュリティグループ」](#) を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス [AuthorizeSecurityGroupIngress](#)」の「」を参照してください。

bundle-instance

次のコード例は、bundle-instance を使用する方法を示しています。

AWS CLI

インスタンスをバンドルするには

この例では、という名前のバケット*i-1234567890abcdef0*にインスタンスをバンドルします*bundletasks*。アクセスキーの値を指定する前にIDs、AWS「アクセスキーの管理に関するベストプラクティス」のガイダンスを確認して従ってください。

コマンド:

```
aws ec2 bundle-instance --instance-id i-1234567890abcdef0 --bucket bundletasks --  
prefix winami --owner-akid AK12AJEXAMPLE --owner-sak example123example
```

出力:

```
{  
  "BundleTask": {  
    "UpdateTime": "2015-09-15T13:30:35.000Z",  
    "InstanceId": "i-1234567890abcdef0",  
    "Storage": {  
      "S3": {  
        "Prefix": "winami",  
        "Bucket": "bundletasks"  
      }  
    },  
    "State": "pending",  
    "StartTime": "2015-09-15T13:30:35.000Z",  
    "BundleId": "bun-294e041f"  
  }  
}
```

- API 詳細については、AWS CLI「コマンドリファレンス[BundleInstance](#)」の「」を参照してください。

cancel-bundle-task

次のコード例は、cancel-bundle-task を使用する方法を示しています。

AWS CLI

バンドルタスクをキャンセルするには

この例では、バンドルタスク をキャンセルします**bun-2a4e041c**。

コマンド:

```
aws ec2 cancel-bundle-task --bundle-id bun-2a4e041c
```

出力:

```
{
  "BundleTask": {
    "UpdateTime": "2015-09-15T13:27:40.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "Storage": {
      "S3": {
        "Prefix": "winami",
        "Bucket": "bundletasks"
      }
    },
    "State": "cancelling",
    "StartTime": "2015-09-15T13:24:35.000Z",
    "BundleId": "bun-2a4e041c"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelBundleTask](#)」の「」を参照してください。

cancel-capacity-reservation-fleets

次のコード例は、cancel-capacity-reservation-fleets を使用する方法を示しています。

AWS CLI

キャパシティ予約フリートをキャンセルするには

次のcancel-capacity-reservation-fleets例では、指定されたキャパシティー予約フリートとそれが予約するキャパシティーをキャンセルします。フリートをキャンセルすると、ステータスは に変わりcancelled、新しいキャパシティ予約を作成できなくなります。さらに、フリート内の個々のキャパシティ予約はすべてキャンセルされ、以前に予約されたキャパシティで実行されていたインスタンスは共有キャパシティで正常に実行され続けます。

```
aws ec2 cancel-capacity-reservation-fleets \  
--capacity-reservation-fleet-ids crf-abcdef01234567890
```

出力:

```
{  
  "SuccessfulFleetCancellations": [  
    {  
      "CurrentFleetState": "cancelling",  
      "PreviousFleetState": "active",  
      "CapacityReservationFleetId": "crf-abcdef01234567890"  
    }  
  ],  
  "FailedFleetCancellations": []  
}
```

キャパシティ予約フリートの詳細については、「Amazon EC2ユーザーガイド」の「[キャパシティ予約フリート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelCapacityReservationFleets](#)」の「」を参照してください。

cancel-capacity-reservation

次の例は、cancel-capacity-reservation を使用する方法を説明しています。

AWS CLI

キャパシティ予約をキャンセルするには

次のcancel-capacity-reservation例では、指定されたキャパシティ予約をキャンセルします。

```
aws ec2 cancel-capacity-reservation \  
--capacity-reservation-id cr-1234abcd56EXAMPLE
```

出力:

```
{  
  "Return": true  
}
```


詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「キャパシティ予約のキャンセル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelCapacityReservation](#)」の「」を参照してください。

cancel-conversion-task

次の例は、cancel-conversion-task を使用する方法を説明しています。

AWS CLI

インスタンスまたはボリュームのアクティブな変換をキャンセルするには

この例では、タスク ID import-i-fh95npoc に関連付けられたアップロードをキャンセルします。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelConversionTask](#)」の「」を参照してください。

cancel-export-task

次の例は、cancel-export-task を使用する方法を説明しています。

AWS CLI

アクティブなエクスポートタスクをキャンセルするには

この例では、タスク ID が export-i-fgelt0i7 のアクティブなエクスポートタスクをキャンセルします。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelExportTask](#)」の「」を参照してください。

cancel-image-launch-permission

次の例は、cancel-image-launch-permission を使用する方法を説明しています。

AWS CLI

Amazon Web Services アカウントとAMIの共有をキャンセルするには

次のcancel-image-launch-permission例では、指定されたAMIの起動アクセス許可からアカウントを削除します。

```
aws ec2 cancel-image-launch-permission \  
  --image-id ami-0123456789example \  
  --region us-east-1
```

出力:

```
{  
  "Return": true  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon Web Services アカウントとAMIの共有のキャンセル」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelImageLaunchPermission](#)」の「」を参照してください。

cancel-import-task

次のコード例は、cancel-import-task を使用する方法を示しています。

AWS CLI

インポートタスクをキャンセルするには

次のcancel-import-task例では、指定されたインポートイメージタスクをキャンセルします。

```
aws ec2 cancel-import-task \  
  --import-task-id import-ami-1234567890abcdef0
```

出力:

```
{
  "ImportTaskId": "import-ami-1234567890abcdef0",
  "PreviousState": "active",
  "State": "deleting"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelImportTask](#)」の「」を参照してください。

cancel-reserved-instances-listing

次の例は、cancel-reserved-instances-listing を使用する方法を説明しています。

AWS CLI

リザーブドインスタンスのリストをキャンセルするには

次のcancel-reserved-instances-listing例では、指定されたリザーブドインスタンスのリストをキャンセルします。

```
aws ec2 cancel-reserved-instances-listing \
  --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelReservedInstancesListing](#)」の「」を参照してください。

cancel-spot-fleet-requests

次のコード例は、cancel-spot-fleet-requests を使用する方法を示しています。

AWS CLI

例 1: スポットフリートリクエストをキャンセルし、関連付けられたインスタンスを終了するには

次のcancel-spot-fleet-requests例では、スポットフリートリクエストをキャンセルし、関連付けられたオンデマンドインスタンスとスポットインスタンスを終了します。

```
aws ec2 cancel-spot-fleet-requests \
```

```
--spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
--terminate-instances
```

出力:

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_terminating",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の [「スポットフリートリクエストのキャンセル」](#) を参照してください。

例 2: 関連付けられたインスタンスを終了せずにスポットフリートリクエストをキャンセルするには

次の `cancel-spot-fleet-requests` 例では、関連付けられたオンデマンドインスタンスとスポットインスタンスを終了せずにスポットフリートリクエストをキャンセルします。

```
aws ec2 cancel-spot-fleet-requests \  
--spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
--no-terminate-instances
```

出力:

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_running",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[スポットフリートリクエストのキャンセル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelSpotFleetRequests](#)」の「」を参照してください。

cancel-spot-instance-requests

次の例は、cancel-spot-instance-requests を使用する方法を説明しています。

AWS CLI

スポットインスタンスリクエストをキャンセルするには

この例では、スポットインスタンスリクエストをキャンセルします。

コマンド:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

出力:

```
{
  "CancelledSpotInstanceRequests": [
    {
      "State": "cancelled",
      "SpotInstanceRequestId": "sir-08b93456"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelSpotInstanceRequests](#)」の「」を参照してください。

confirm-product-instance

次のコード例は、confirm-product-instance を使用する方法を示しています。

AWS CLI

製品インスタンスを確認するには

この例では、指定された製品コードが指定されたインスタンスに関連付けられているかどうかを決定します。

コマンド:

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-id i-1234567890abcdef0
```

出力:

```
{  
  "OwnerId": "123456789012"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfirmProductInstance](#)」の「」を参照してください。

copy-fpga-image

次のコード例は、copy-fpga-image を使用する方法を示しています。

AWS CLI

Amazon FPGAイメージをコピーするには

この例では、指定された をus-east-1リージョンAFIから現在のリージョン () にコピーしますeu-west-1。

コマンド:

```
aws ec2 copy-fpga-image --name copy-afi --source-fpga-image-id afi-0d123e123bfc85abc  
--source-region us-east-1 --region eu-west-1
```

出力:

```
{  
  "FpgaImageId": "afi-06b12350a123fbabc"  
}
```

- API 詳細については、「コマンドリファレンス[CopyFpgaImage](#)」の「」を参照してください。AWS CLI

copy-image

次のコード例は、copy-image を使用する方法を示しています。

AWS CLI

例 1: を別のリージョンにコピーAMIするには

次のコマンドcopy-image例では、指定された をus-west-2リージョンAMIからus-east-1リージョンにコピーし、簡単な説明を追加します。

```
aws ec2 copy-image \  
  --region us-east-1 \  
  --name ami-name \  
  --source-region us-west-2 \  
  --source-image-id ami-066877671789bd71b \  
  --description "This is my copied image."
```

出力:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

詳細については、「[Amazon ユーザーガイド](#)」の「[のコピーAMI](#)」を参照してください。 EC2

例 2: AMIを別のリージョンにコピーし、バックアップスナップショットを暗号化するには

次のcopy-imageコマンドは、リージョンAMIから現在のリージョンus-west-2に指定された をコピーし、指定されたKMSキーを使用してバックアップスナップショットを暗号化します。

```
aws ec2 copy-image \  
  --source-region us-west-2 \  
  --name ami-name \  
  --source-image-id ami-066877671789bd71b \  
  --encrypted \  
  --kms-key-id alias/my-kms-key
```

出力:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

```
}
```

詳細については、「Amazon ユーザーガイド」の「[のコピーAMI](#)」を参照してください。 EC2

例 3: をコピーするときにユーザー定義AMIタグを含めるには AMI

次のcopy-imageコマンドは、--copy-image-tagsパラメータを使用して、 のコピー時にユーザー定義AMIタグをコピーしますAMI。

```
aws ec2 copy-image \  
  --region us-east-1 \  
  --name ami-name \  
  --source-region us-west-2 \  
  --source-image-id ami-066877671789bd71b \  
  --description "This is my copied image." \  
  --copy-image-tags
```

出力:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

詳細については、「Amazon ユーザーガイド」の「[のコピーAMI](#)」を参照してください。 EC2

• API 詳細については、AWS CLI 「コマンドリファレンス[CopyImage](#)」の「」を参照してください。

copy-snapshot

次のコード例は、copy-snapshot を使用する方法を示しています。

AWS CLI

例 1: スナップショットを別のリージョンにコピーするには

次のコマンドcopy-snapshot例では、指定されたスナップショットをus-west-2リージョンからus-east-1リージョンにコピーし、簡単な説明を追加します。

```
aws ec2 copy-snapshot \  
  --region us-east-1 \  
  --source-region us-west-2 \  
  --source-snapshot-id ami-066877671789bd71b \  
  --description "This is my copied snapshot."
```



```
--source-snapshot-id snap-066877671789bd71b \  
--description "This is my copied snapshot."
```

出力:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EBSスナップショットのコピー」](#)を参照してください。 EC2

例 2: 暗号化されていないスナップショットをコピーし、新しいスナップショットを暗号化するには

次のcopy-snapshotコマンドは、指定された暗号化されていないスナップショットをus-west-2リージョンから現在のリージョンにコピーし、指定されたKMSキーを使用して新しいスナップショットを暗号化します。

```
aws ec2 copy-snapshot \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --encrypted \  
  --kms-key-id alias/my-kms-key
```

出力:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EBSスナップショットのコピー」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[CopySnapshot](#)」の「」を参照してください。

create-capacity-reservation-fleet

次のコード例は、create-capacity-reservation-fleet を使用する方法を示しています。

AWS CLI

キャパシティ予約フリートを作成するには

次のcreate-capacity-reservation-fleet例では、リクエストで指定されたインスタンスタイプのキャパシティ予約フリートを、指定された合計ターゲットキャパシティまで作成します。キャパシティー予約フリートがキャパシティーを予約するインスタンスの数は、リクエストで指定する合計ターゲット容量とインスタンスタイプの重みによって異なります。使用するインスタンスタイプと、指定された各インスタンスタイプの優先度を指定します。

```
aws ec2 create-capacity-reservation-fleet \  
--total-target-capacity 24 \  
--allocation-strategy prioritized \  
--instance-match-criteria open \  
--tenancy default \  
--end-date 2022-12-31T23:59:59.000Z \  
--instance-type-specifications file://instanceTypeSpecification.json
```

instanceTypeSpecification.json の内容:

```
[  
  {  
    "InstanceType": "m5.xlarge",  
    "InstancePlatform": "Linux/UNIX",  
    "Weight": 3.0,  
    "AvailabilityZone": "us-east-1a",  
    "EbsOptimized": true,  
    "Priority" : 1  
  }  
]
```

出力:

```
{  
  "Status": "submitted",  
  "TotalFulfilledCapacity": 0.0,  
  "CapacityReservationFleetId": "crf-abcdef01234567890",  
  "TotalTargetCapacity": 24  
}
```

キャパシティ予約フリートの詳細については、「Amazon EC2ユーザーガイド」の [「キャパシティー予約フリート」](#) を参照してください。

インスタンスタイプの重みと合計ターゲット容量の詳細については、「Amazon ユーザーガイド」の「[インスタンスタイプの重みと合計ターゲット容量](#)」を参照してください。 EC2

指定されたインスタンスタイプの優先度の指定の詳細については、「Amazon EC2ユーザーガイド」の「[配分戦略とインスタンスタイプの優先度](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCapacityReservationFleet](#)」の「」を参照してください。

create-capacity-reservation

次のコード例は、create-capacity-reservation を使用する方法を示しています。

AWS CLI

例 1: キャパシティ予約を作成するには

次のcreate-capacity-reservation例では、eu-west-1aアベイラビリティゾーンにキャパシティ予約を作成し、Linux/Unix オペレーティングシステムを実行している 3 つのt2.mediumインスタンスを起動できます。デフォルトでは、キャパシティ予約はオープンインスタンス一致基準で作成され、エフェメラルストレージはサポートされません。手動でキャンセルするまでアクティブのままになります。

```
aws ec2 create-capacity-reservation \  
  --availability-zone eu-west-1a \  
  --instance-type t2.medium \  
  --instance-platform Linux/UNIX \  
  --instance-count 3
```

出力:

```
{  
  "CapacityReservation": {  
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",  
    "EndDateType": "unlimited",  
    "AvailabilityZone": "eu-west-1a",  
    "InstanceMatchCriteria": "open",  
    "EphemeralStorage": false,  
    "CreateDate": "2019-08-16T09:27:35.000Z",  
    "AvailableInstanceCount": 3,  
    "InstancePlatform": "Linux/UNIX",  
    "TotalInstanceCount": 3,
```

```
    "State": "active",
    "Tenancy": "default",
    "EbsOptimized": false,
    "InstanceType": "t2.medium"
  }
}
```

例 2: 指定された日付/時刻に自動的に終了するキャパシティ予約を作成するには

次のcreate-capacity-reservation例では、アeu-west-1aベイラビリティゾーンにキャパシティ予約を作成し、Linux/Unix オペレーティングシステムを実行している 3 つのm5.largeインスタンスを起動できます。このキャパシティ予約は、08/31/2019 の 23:59:59 に自動的に終了します。

```
aws ec2 create-capacity-reservation \
  --availability-zone eu-west-1a \
  --instance-type m5.large \
  --instance-platform Linux/UNIX \
  --instance-count 3 \
  --end-date-type limited \
  --end-date 2019-08-31T23:59:59Z
```

出力:

```
{
  "CapacityReservation": {
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
    "EndDateType": "limited",
    "AvailabilityZone": "eu-west-1a",
    "EndDate": "2019-08-31T23:59:59.000Z",
    "InstanceMatchCriteria": "open",
    "EphemeralStorage": false,
    "CreateDate": "2019-08-16T10:15:53.000Z",
    "AvailableInstanceCount": 3,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 3,
    "State": "active",
    "Tenancy": "default",
    "EbsOptimized": false,
    "InstanceType": "m5.large"
  }
}
```

例 3: ターゲットインスタンスの起動のみを受け入れるキャパシティ予約を作成するには

次のcreate-capacity-reservation例では、ターゲットインスタンスの起動のみを受け入れるキャパシティ予約を作成します。

```
aws ec2 create-capacity-reservation \  
  --availability-zone eu-west-1a \  
  --instance-type m5.large \  
  --instance-platform Linux/UNIX \  
  --instance-count 3 \  
  --instance-match-criteria targeted
```

出力:

```
{  
  "CapacityReservation": {  
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",  
    "EndDateType": "unlimited",  
    "AvailabilityZone": "eu-west-1a",  
    "InstanceMatchCriteria": "targeted",  
    "EphemeralStorage": false,  
    "CreateDate": "2019-08-16T10:21:57.000Z",  
    "AvailableInstanceCount": 3,  
    "InstancePlatform": "Linux/UNIX",  
    "TotalInstanceCount": 3,  
    "State": "active",  
    "Tenancy": "default",  
    "EbsOptimized": false,  
    "InstanceType": "m5.large"  
  }  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[キャパシティ予約の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCapacityReservation](#)」の「」を参照してください。

create-carrier-gateway

次のコード例は、create-carrier-gateway を使用する方法を示しています。

AWS CLI

キャリアゲートウェイを作成するには

次のcreate-carrier-gateway例では、指定された のキャリアゲートウェイを作成します VPC。

```
aws ec2 create-carrier-gateway \  
  --vpc-id vpc-0c529aEXAMPLE1111
```

出力:

```
{  
  "CarrierGateway": {  
    "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",  
    "VpcId": "vpc-0c529aEXAMPLE1111",  
    "State": "pending",  
    "OwnerId": "123456789012"  
  }  
}
```

詳細については、AWS 「Wavelength ユーザーガイド」の「[キャリアゲートウェイ](#)」を参照してください。

- API 詳細については、「[コマンドリファレンスCreateCarrierGateway](#)」の「」を参照してください。AWS CLI

create-client-vpn-endpoint

次のコード例は、create-client-vpn-endpoint を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントを作成するには

次のcreate-client-vpn-endpoint例では、相互認証を使用するクライアントVPNエンドポイントを作成し、クライアントCIDRブロックの値を指定します。

```
aws ec2 create-client-vpn-endpoint \  
  --client-cidr-block "172.31.0.0/16" \  
  --
```

```
--server-certificate-arn arn:aws:acm:ap-south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
--authentication-options Type=certificate-authentication,MutualAuthentication={ClientRootCertificateChainArn=arn:aws:acm:ap-south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE} \  
--connection-log-options Enabled=false
```

出力:

```
{  
  "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",  
  "Status": {  
    "Code": "pending-associate"  
  },  
  "DnsName": "cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-south-1.amazonaws.com"  
}
```

詳細については、[「クライアント管理者ガイド」の「クライアントVPNエンドポイント」](#)を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateClientVpnEndpoint](#)」の「」を参照してください。

create-client-vpn-route

次の例は、create-client-vpn-route を使用する方法を説明しています。

AWS CLI

クライアントVPNエンドポイントのルートを作成するには

次のcreate-client-vpn-route例では、クライアントVPNエンドポイントの指定されたサブネットのインターネット (0.0.0.0/0) にルートを追加します。

```
aws ec2 create-client-vpn-route \  
--client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
--destination-cidr-block 0.0.0.0/0 \  
--target-vpc-subnet-id subnet-0123456789abcabca
```

出力:

```
{
  "Status": {
    "Code": "creating"
  }
}
```

詳細については、AWS 「クライアントVPN管理者ガイド」の「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateClientVpnRoute](#)」の「」を参照してください。

create-coip-cidr

次の例は、create-coip-cidr を使用する方法を説明しています。

AWS CLI

カスタマー所有 IP (CoIP) アドレスの範囲を作成するには

次のcreate-coip-cidr例では、指定された CoIP プールに指定された CoIP アドレスの範囲を作成します。

```
aws ec2 create-coip-cidr \
  --cidr 15.0.0.0/24 \
  --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

出力:

```
{
  "CoipCidr": {
    "Cidr": "15.0.0.0/24",
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
  }
}
```

CoIP の詳細については、AWS Outposts ユーザーガイドの「[カスタマー所有の IP アドレス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCoipCidr](#)」の「」を参照してください。

create-coip-pool

次の例は、create-coip-pool を使用する方法を説明しています。

AWS CLI

カスタマー所有 IP (CoIP) アドレスのプールを作成するには

次のcreate-coip-pool例では、指定されたローカルゲートウェイルートテーブルに CoIP アドレスの CoIP プールを作成します。

```
aws ec2 create-coip-pool \  
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

出力:

```
{  
  "CoipPool": {  
    "PoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-  
coip-1234567890abcdefg"  
  }  
}
```

CoIP の詳細については、AWS Outposts ユーザーガイドの「[カスタマー所有の IP アドレス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCoipPool](#)」の「」を参照してください。

create-customer-gateway

次の例は、create-customer-gateway を使用する方法を説明しています。

AWS CLI

カスタマーゲートウェイを作成するには

この例では、外部インターフェイスに指定された IP アドレスを持つカスタマーゲートウェイを作成します。

コマンド:

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```

出力:

```
{
  "CustomerGateway": {
    "CustomerGatewayId": "cgw-0e11f167",
    "IpAddress": "12.1.2.3",
    "State": "available",
    "Type": "ipsec.1",
    "BgpAsn": "65534"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomerGateway](#)」の「」を参照してください。

create-default-subnet

次の例は、create-default-subnet を使用する方法を説明しています。

AWS CLI

デフォルトサブネットを作成するには

この例では、アベイラビリティゾーンにデフォルトのサブネットを作成しますus-east-2a。

コマンド:

```
aws ec2 create-default-subnet --availability-zone us-east-2a
```

```
{
  "Subnet": {
    "AvailabilityZone": "us-east-2a",
    "Tags": [],
    "AvailableIpAddressCount": 4091,
    "DefaultForAz": true,
    "Ipv6CidrBlockAssociationSet": [],
    "VpcId": "vpc-1a2b3c4d",
    "State": "available",
  }
}
```

```
    "MapPublicIpOnLaunch": true,  
    "SubnetId": "subnet-1122aabb",  
    "CidrBlock": "172.31.32.0/20",  
    "AssignIpv6AddressOnCreation": false  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDefaultSubnet](#)」の「」を参照してください。

create-default-vpc

次のコード例は、create-default-vpc を使用する方法を示しています。

AWS CLI

デフォルトを作成するには VPC

この例では、デフォルトの を作成しますVPC。

コマンド:

```
aws ec2 create-default-vpc
```

出力:

```
{  
  "Vpc": {  
    "VpcId": "vpc-8eaae5ea",  
    "InstanceTenancy": "default",  
    "Tags": [],  
    "Ipv6CidrBlockAssociationSet": [],  
    "State": "pending",  
    "DhcpOptionsId": "dopt-af0c32c6",  
    "CidrBlock": "172.31.0.0/16",  
    "IsDefault": true  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDefaultVpc](#)」の「」を参照してください。

create-dhcp-options

次のコード例は、create-dhcp-options を使用する方法を示しています。

AWS CLI

一連のDHCPオプションを作成するには

次のcreate-dhcp-options例では、ドメイン名、ドメインネームサーバー、NetBIOS ノードタイプを指定する一連のDHCPオプションを作成します。

```
aws ec2 create-dhcp-options \  
  --dhcp-configuration \  
    "Key=domain-name-servers,Values=10.2.5.1,10.2.5.2" \  
    "Key=domain-name,Values=example.com" \  
    "Key=netbios-node-type,Values=2"
```

出力:

```
{  
  "DhcpOptions": {  
    "DhcpConfigurations": [  
      {  
        "Key": "domain-name",  
        "Values": [  
          {  
            "Value": "example.com"  
          }  
        ]  
      },  
      {  
        "Key": "domain-name-servers",  
        "Values": [  
          {  
            "Value": "10.2.5.1"  
          },  
          {  
            "Value": "10.2.5.2"  
          }  
        ]  
      },  
      {  
        "Key": "netbios-node-type",
```

```
        "Values": [
          {
            "Value": "2"
          }
        ]
      },
    ],
    "DhcpOptionsId": "dopt-06d52773eff4c55f3"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDhcpOptions](#)」の「」を参照してください。

create-egress-only-internet-gateway

次の例は、`create-egress-only-internet-gateway` を使用する方法を説明しています。

AWS CLI

Egress-Only Internet Gateway を作成するには

この例では、指定された のエグレス専用インターネットゲートウェイを作成しますVPC。

コマンド:

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

出力:

```
{
  "EgressOnlyInternetGateway": {
    "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
    "Attachments": [
      {
        "State": "attached",
        "VpcId": "vpc-0c62a468"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateEgressOnlyInternetGateway](#)」の「」を参照してください。

create-fleet

次のコード例は、create-fleet を使用する方法を示しています。

AWS CLI

例 1: スポットインスタンスをデフォルトの購入モデルとして起動する EC2 フリートを作成するには

次の create-fleet 例では、EC2 フリーの起動に必要な最小パラメータを使用してフリートを作成します。起動テンプレート、ターゲット容量、デフォルトの購入モデルです。起動テンプレートは、起動テンプレート ID とバージョン番号によって識別されます。フリートのターゲット容量は 2 インスタンスで、デフォルトの購入モデルは spot です。これにより spot、フリートは 2 つのスポットインスタンスを起動します。

EC2 フリートを作成するときは、JSON ファイルを使用して起動するインスタンスに関する情報を指定します。

```
aws ec2 create-fleet \  
  --cli-input-json file:///file_name.json
```

file_name.json の内容 :

```
{  
  "LaunchTemplateConfigs": [  
    {  
      "LaunchTemplateSpecification": {  
        "LaunchTemplateId": "lt-0e8c754449b27161c",  
        "Version": "1"  
      }  
    }  
  ],  
  "TargetCapacitySpecification": {  
    "TotalTargetCapacity": 2,  
    "DefaultTargetCapacityType": "spot"  
  }  
}
```

出力:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

例 2: デフォルトの購入モデルとしてオンデマンドインスタンスを起動する EC2 フリートを作成するには

次の `create-fleet` 例では、EC2 フリーの起動に必要な最小パラメータを使用してフリートを作成します。起動テンプレート、ターゲット容量、デフォルトの購入モデルです。起動テンプレートは、起動テンプレート ID とバージョン番号によって識別されます。フリートのターゲット容量は 2 インスタンスで、デフォルトの購入モデルは `on-demand` です。これにより `on-demand`、フリートは 2 つのオンデマンドインスタンスを起動します。

EC2 フリートを作成するときは、JSON ファイルを使用して起動するインスタンスに関する情報を指定します。

```
aws ec2 create-fleet \
  --cli-input-json file://file_name.json
```

`file_name.json` の内容 :

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
  }
}
```

出力:

```
{
```

```
"FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

例 3: プライマリ容量としてオンデマンドインスタンスを起動する EC2 フリートを作成するには

次の `create-fleet` 例では、EC2 フリートの 2 つのインスタンスの合計ターゲット容量と 1 つのオンデマンドインスタンスのターゲット容量を指定するフリートを作成します。デフォルトの購入モデルは `spot` です。フリートは指定されたとおりに 1 つのオンデマンドインスタンスを起動しますが、合計ターゲット容量を満たすには、もう 1 つのインスタンスを起動する必要があります。差分の購入モデルは `TotalTargetCapacity-OnDemandTargetCapacity` として計算され、フリートは 1 スポットインスタンスを起動します。

EC2 フリートを作成するときは、JSON ファイルを使用して起動するインスタンスに関する情報を指定します。

```
aws ec2 create-fleet \
  --cli-input-json file://file_name.json
```

file_name.json の内容 :

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "OnDemandTargetCapacity": 1,
    "DefaultTargetCapacityType": "spot"
  }
}
```

出力:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```



```
}
```

例 4: 最低料金配分戦略を使用してスポットインスタンスを起動する EC2 フリートを作成するには

スポットインスタンスの配分戦略を指定しない場合、デフォルト配分戦略である `lowest-price` が使用されます。次の `create-fleet` 例では、`lowest-price` 配分戦略を使用して EC2 フリートを作成します。起動テンプレートをオーバーライドする 3 つの起動条件は、インスタンスタイプが異なりますが、加重容量とサブネットは同じです。合計ターゲット容量は 2 インスタンスで、デフォルトの購入モデルは `spot` です。EC2 フリートは、最低料金で起動仕様のインスタンスタイプを使用して 2 つのスポットインスタンスを起動します。

EC2 フリートを作成するときは、JSON ファイルを使用して起動するインスタンスに関する情報を指定します。

```
aws ec2 create-fleet \  
  --cli-input-json file://file_name.json Contents of file_name.json:  
  
{  
  "LaunchTemplateConfigs": [  
    {  
      "LaunchTemplateSpecification": {  
        "LaunchTemplateId": "lt-0e8c754449b27161c",  
        "Version": "1"  
      },  
      "Overrides": [  
        {  
          "InstanceType": "c4.large",  
          "WeightedCapacity": 1,  
          "SubnetId": "subnet-a4f6c5d3"  
        },  
        {  
          "InstanceType": "c3.large",  
          "WeightedCapacity": 1,  
          "SubnetId": "subnet-a4f6c5d3"  
        },  
        {  
          "InstanceType": "c5.large",  
          "WeightedCapacity": 1,  
          "SubnetId": "subnet-a4f6c5d3"  
        }  
      ]  
    }  
  ]  
}
```

```
],  
  "TargetCapacitySpecification": {  
    "TotalTargetCapacity": 2,  
    "DefaultTargetCapacityType": "spot"  
  }  
}
```

出力:

```
{  
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateFleet](#)」の「」を参照してください。

create-flow-logs

次のコード例は、create-flow-logs を使用する方法を示しています。

AWS CLI

例 1: フローログを作成するには

次のcreate-flow-logs例では、指定されたネットワークインターフェイスのすべての拒否されたトラフィックをキャプチャするフローログを作成します。フローログは、指定されたIAM ロールのアクセス許可を使用して CloudWatch Logs のロググループに配信されます。

```
aws ec2 create-flow-logs \  
  --resource-type NetworkInterface \  
  --resource-ids eni-11223344556677889 \  
  --traffic-type REJECT \  
  --log-group-name my-flow-logs \  
  --deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
```

出力:

```
{  
  "ClientToken": "so0eNA2uSHUN1HI0S2cJ305GuIX1CezaRdGtexample",  
  "FlowLogIds": [  
    "  
  ]  
}
```

```
"f1-12345678901234567"  
],  
"Unsuccessful": []  
}
```

詳細については、「Amazon VPCユーザーガイド[VPC](#)」の「[フローログ](#)」を参照してください。

例 2: カスタム形式でフローログを作成するには

次のcreate-flow-logs例では、指定された のすべてのトラフィックをキャプチャするフローログを作成しVPC、フローログを Amazon S3 バケットに配信します。--log-format パラメータにより、フローログレコードのカスタム形式が指定されます。Windows でこのコマンドを実行するには、一重引用符 (') を二重引用符 (") に変更します。

```
aws ec2 create-flow-logs \  
  --resource-type VPC \  
  --resource-ids vpc-00112233344556677 \  
  --traffic-type ALL \  
  --log-destination-type s3 \  
  --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \  
  --log-format '${version} ${vpc-id} ${subnet-id} ${instance-id} ${srcaddr} \  
  ${dstaddr} ${srcport} ${dstport} ${protocol} ${tcp-flags} ${type} ${pkt-srcaddr} \  
  ${pkt-dstaddr}'
```

詳細については、「Amazon VPCユーザーガイド[VPC](#)」の「[フローログ](#)」を参照してください。

例 3: 1 分間の最大集約間隔でフローログを作成するには

次のcreate-flow-logs例では、指定された のすべてのトラフィックをキャプチャするフローログを作成しVPC、フローログを Amazon S3 バケットに配信します。--max-aggregation-interval パラメータは、最大集約間隔を 60 秒 (1 分) に指定します。

```
aws ec2 create-flow-logs \  
  --resource-type VPC \  
  --resource-ids vpc-00112233344556677 \  
  --traffic-type ALL \  
  --log-destination-type s3 \  
  --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \  
  --max-aggregation-interval 60
```

詳細については、「Amazon VPCユーザーガイド」の[VPC](#)「[フローログ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFlowLogs](#)」の「」を参照してください。

create-fpga-image

次のコード例は、create-fpga-image を使用する方法を示しています。

AWS CLI

Amazon FPGAイメージを作成するには

この例では、指定されたバケット内の指定された tarball AFIから を作成します。

コマンド:

```
aws ec2 create-fpga-image --name my-afi --description test-afi --input-storage-location Bucket=my-fpga-bucket,Key=dcp/17_12_22-103226.Developer_CL.tar --logs-storage-location Bucket=my-fpga-bucket,Key=logs
```

出力:

```
{
  "FpgaImageId": "afi-0d123e123bfc85abc",
  "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFpgaImage](#)」の「」を参照してください。

create-image

次のコード例は、create-image を使用する方法を示しています。

AWS CLI

例 1: Amazon EBSバックアップインスタンスAMIから を作成するには

次のcreate-image例では、指定されたインスタンスAMIから を作成します。

```
aws ec2 create-image \
```

```
--instance-id i-1234567890abcdef0 \  
--name "My server" \  
--description "An AMI for my server"
```

出力:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

のブロックデバイスマッピングの指定の詳細についてはAMI、Amazon EC2ユーザーガイドの「[のブロックデバイスマッピングの指定AMI](#)」を参照してください。

例 2: 再起動せずに Amazon EBSでバックアップされたインスタンスAMIから を作成するには

次のcreate-image例ではAMI、 を作成し、--no-reboot パラメータを設定して、イメージの作成前にインスタンスを再起動しないようにします。

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --no-reboot
```

出力:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

のブロックデバイスマッピングの指定の詳細についてはAMI、「Amazon EC2ユーザーガイド」の「[のブロックデバイスマッピングの指定AMI](#)」を参照してください。

例 3: 作成時に AMIとスナップショットにタグを付けるには

次のcreate-image例では、 を作成しAMI、 とスナップショットAMIに同じタグを付けます。
cost-center=cc123

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --tag-specifications "tag:cost-center=cc123"
```

```
--tag-specifications "ResourceType=image,Tags=[{Key=cost-center,Value=cc123}]" "ResourceType=snapshot,Tags=[{Key=cost-center,Value=cc123}]"
```

出力:

```
{
  "ImageId": "ami-abcdef01234567890"
}
```

作成時のリソースのタグ付けの詳細については、「Amazon EC2ユーザーガイド」の[「リソース作成時にタグを追加する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateImage](#)」の「」を参照してください。

create-instance-connect-endpoint

次のコード例は、create-instance-connect-endpoint を使用する方法を示しています。

AWS CLI

EC2 Instance Connect エンドポイントを作成するには

次のcreate-instance-connect-endpoint例では、指定されたサブネットに EC2 Instance Connect エンドポイントを作成します。

```
aws ec2 create-instance-connect-endpoint \
  --region us-east-1 \
  --subnet-id subnet-0123456789example
```

出力:

```
{
  "VpcId": "vpc-0123abcd",
  "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
  "AvailabilityZone": "us-east-1a",
  "NetworkInterfaceIds": [
    "eni-0123abcd"
  ],
  "PreserveClientIp": true,
  "Tags": [],
}
```

```
"FipsDnsName": "eice-0123456789example.0123abcd.fips.ec2-instance-connect-  
endpoint.us-east-1.amazonaws.com",  
"StateMessage": "",  
"State": "create-complete",  
"DnsName": "eice-0123456789example.0123abcd.ec2-instance-connect-endpoint.us-  
east-1.amazonaws.com",  
"SubnetId": "subnet-0123abcd",  
"OwnerId": "111111111111",  
"SecurityGroupIds": [  
    "sg-0123abcd"  
],  
"InstanceConnectEndpointId": "eice-0123456789example",  
"CreatedAt": "2023-04-07T15:43:53.000Z"  
}
```

詳細については、「Amazon ユーザーガイド」の[EC2「インスタンス接続エンドポイントの作成」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInstanceConnectEndpoint](#)」の「」を参照してください。

create-instance-event-window

次の例は、create-instance-event-window を使用する方法を説明しています。

AWS CLI

例 1: 時間範囲を持つイベントウィンドウを作成するには

次のcreate-instance-event-window例では、時間範囲を持つイベントウィンドウを作成します。また、cron-expression パラメータを指定することはできません。

```
aws ec2 create-instance-event-window \  
  --region us-east-1 \  
  --time-range StartWeekDay=monday, StartHour=2, EndWeekDay=wednesday, EndHour=8 \  
  --tag-specifications "ResourceType=instance-event-  
window, Tags=[{Key=K1, Value=V1}]" \  
  --name myEventWindowName
```

出力:

```
{
```

```
"InstanceEventWindow": {
  "InstanceEventWindowId": "iew-0abcdef1234567890",
  "TimeRanges": [
    {
      "StartWeekDay": "monday",
      "StartHour": 2,
      "EndWeekDay": "wednesday",
      "EndHour": 8
    }
  ],
  "Name": "myEventWindowName",
  "State": "creating",
  "Tags": [
    {
      "Key": "K1",
      "Value": "V1"
    }
  ]
}
```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの[「考慮事項」](#)を参照してください。

例 2: cron 式を使用してイベントウィンドウを作成するには

次のcreate-instance-event-window例では、cron 式を持つイベントウィンドウを作成します。また、time-range パラメータを指定することはできません。

```
aws ec2 create-instance-event-window \
  --region us-east-1 \
  --cron-expression "* 21-23 * * 2,3" \
  --tag-specifications "ResourceType=instance-event-  
window,Tags=[{Key=K1,Value=V1}]" \
  --name myEventWindowName
```

出力:

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
```



```
    "CronExpression": "* 21-23 * * 2,3",
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}
```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの「[考慮事項](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInstanceEventWindow](#)」の「」を参照してください。

create-instance-export-task

次の例は、create-instance-export-task を使用する方法を説明しています。

AWS CLI

インスタンスをエクスポートするには

このコマンド例では、インスタンス i-1234567890abcdef0 を Amazon S3 バケット myexportbucket にエクスポートするタスクを作成します。

コマンド:

```
aws ec2 create-instance-export-task --description "RHEL5 instance" --
instance-id i-1234567890abcdef0 --target-environment vmware --export-to-s3-
task DiskImageFormat=vmdk,ContainerFormat=ova,S3Bucket=myexportbucket,S3Prefix=RHEL5
```

出力:

```
{
  "ExportTask": {
    "State": "active",
    "InstanceExportDetails": {
      "InstanceId": "i-1234567890abcdef0",
      "TargetEnvironment": "vmware"
    }
  }
}
```

```

    },
    "ExportToS3Task": {
      "S3Bucket": "myexportbucket",
      "S3Key": "RHEL5export-i-fh8sjjsq.ova",
      "DiskImageFormat": "vmdk",
      "ContainerFormat": "ova"
    },
    "Description": "RHEL5 instance",
    "ExportTaskId": "export-i-fh8sjjsq"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateInstanceExportTask](#)」の「」を参照してください。

create-internet-gateway

次の例は、create-internet-gateway を使用する方法を説明しています。

AWS CLI

インターネットゲートウェイを作成するには

次のcreate-internet-gateway例では、タグを使用してインターネットゲートウェイを作成しますName=my-igw。

```

aws ec2 create-internet-gateway \
  --tag-specifications ResourceType=internet-gateway, Tags=[{Key=Name, Value=my-igw}]

```

出力:

```

{
  "InternetGateway": {
    "Attachments": [],
    "InternetGatewayId": "igw-0d0fb496b3994d755",
    "OwnerId": "123456789012",
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-igw"
      }
    ]
  }
}

```

```
    }  
  ]  
}  
}
```

詳細については、「[Amazon VPCユーザーガイド](#)」の「[インターネットゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInternetGateway](#)」の「」を参照してください。

create-ipam-pool

次の例は、create-ipam-pool を使用する方法を説明しています。

AWS CLI

IPAM プールを作成するには

次のcreate-ipam-pool例では、IPAMプールを作成します。

(Linux):

```
aws ec2 create-ipam-pool \  
  --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \  
  --address-family ipv4 \  
  --auto-import \  
  --allocation-min-netmask-length 16 \  
  --allocation-max-netmask-length 26 \  
  --allocation-default-netmask-length 24 \  
  --allocation-resource-tags "Key=Environment,Value=Preprod" \  
  --tag-specifications 'ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod pool"}]'
```

(Windows):

```
aws ec2 create-ipam-pool ^  
  --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^  
  --address-family ipv4 ^  
  --auto-import ^  
  --allocation-min-netmask-length 16 ^  
  --allocation-max-netmask-length 26 ^
```

```
--allocation-default-netmask-length 24 ^
--allocation-resource-tags "Key=Environment,Value=Preprod" ^
--tag-specifications ResourceType=ipam-pool,Tags=[{Key=Name,Value="Preprod
pool"}]
```

出力:

```
{
  "IpamPool": {
    "OwnerId": "123456789012",
    "IpamPoolId": "ipam-pool-0533048da7d823723",
    "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-0533048da7d823723",
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-02fc38cd4c48e7d38",
    "IpamScopeType": "private",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
    "IpamRegion": "us-east-1",
    "Locale": "None",
    "PoolDepth": 1,
    "State": "create-in-progress",
    "AutoImport": true,
    "AddressFamily": "ipv4",
    "AllocationMinNetmaskLength": 16,
    "AllocationMaxNetmaskLength": 26,
    "AllocationDefaultNetmaskLength": 24,
    "AllocationResourceTags": [
      {
        "Key": "Environment",
        "Value": "Preprod"
      }
    ],
    "Tags": [
      {
        "Key": "Name",
        "Value": "Preprod pool"
      }
    ]
  }
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の [「IP アドレスプロビジョニングの計画」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateIpamPool](#)」の「」を参照してください。

create-ipam-resource-discovery

次の例は、create-ipam-resource-discovery を使用する方法を説明しています。

AWS CLI

リソース検出を作成するには

この例では、委任されたIPAM管理者は、他の AWS 組織のIPAM管理者が組織内のリソースの IP アドレスを管理およびモニタリングできるように、リソース検出を作成して別の組織の管理者と共有します。

[重要]

この例では、`--region` オプションと `--operating-regions` オプションの両方が含まれています。これは、オプションはオプションですが、リソース検出を と正常に統合するために特定の方法で設定する必要がありますIPAM。* は、IPAM検出するリソースがあるリージョンと一致する `--operating-regions` 必要があります。IP アドレスIPAMを管理しないリージョン (コンプライアンス上の理由など) がある場合は、それらを含めないでください。* は、関連付けIPAM のホームリージョンと一致する `--region` 必要があります。リソース検出は、 がIPAM作成されたのと同じリージョンで作成する必要があります。例えば、IPAM関連付ける が us-east-1 で作成された場合は、 リクエスト `--region us-east-1` に を含めます。 `--region` と `--operating-regions` の両方のオプションは、コマンドを実行しているリージョンにデフォルト設定されています。

この例では、統合IPAMする の運用リージョンには、us-west-1、us-west-2、および が含まれますap-south-1。リソース検出を作成するときは、 us-west-1 および でリソース IP アドレスを検出IPAMしますus-west-2が、 では検出しませんap-south-1。そのため、リクエスト `--operating-regions RegionName='us-west-1' RegionName='us-west-2'` にのみ を含めます。

次のcreate-ipam-resource-discovery例では、IPAMリソース検出を作成します。

```
aws ec2 create-ipam-resource-discovery \
  --description 'Example-resource-discovery' \
  --tag-specifications 'ResourceType=ipam-resource-discovery,Tags=[{Key=cost-center,Value=cc123}]' \
```

```
--operating-regions RegionName='us-west-1' RegionName='us-west-2' \  
--region us-east-1
```

出力:

```
{  
  "IpamResourceDiscovery":{  
    "OwnerId": "149977607591",  
    "IpamResourceDiscoveryId": "ipam-res-disco-0257046d8aa78b8bc",  
    "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-  
discovery/ipam-res-disco-0257046d8aa78b8bc",  
    "IpamResourceDiscoveryRegion": "us-east-1",  
    "Description": "'Example-resource-discovery'",  
    "OperatingRegions":[  
      {"RegionName": "us-west-1"},  
      {"RegionName": "us-west-2"},  
      {"RegionName": "us-east-1"}  
    ],  
    "IsDefault": false,  
    "State": "create-in-progress",  
    "Tags": [  
      {  
        "Key": "cost-center",  
        "Value": "cc123"  
      }  
    ]  
  }  
}
```

リソース検出を作成したら、別のIPAM委任された管理者と共有できます。これは [create-resource-share](#) で行うことができます。詳細については、「Amazon VPC IPAM ユーザーガイド」の「[組織外のアカウント IPAM との統合](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateIpamResourceDiscovery](#)」の「」を参照してください。

create-ipam-scope

次の例は、create-ipam-scope を使用する方法を説明しています。

AWS CLI

IPAM スコープを作成するには

次のcreate-ipam-scope例では、IPAMスコープを作成します。

(Linux):

```
aws ec2 create-ipam-scope \  
  --ipam-id ipam-08440e7a3acde3908 \  
  --description "Example description" \  
  --tag-specifications 'ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example  
name value"}]'
```

(Windows):

```
aws ec2 create-ipam-scope ^  
  --ipam-id ipam-08440e7a3acde3908 ^  
  --description "Example description" ^  
  --tag-specifications ResourceType=ipam-scope,Tags=[{Key=Name,Value="Example name  
value"}]
```

出力:

```
{  
  "IpamScope": {  
    "OwnerId": "123456789012",  
    "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",  
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-01c1ebab2b63bd7e4",  
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
    "IpamRegion": "us-east-1",  
    "IpamScopeType": "private",  
    "IsDefault": false,  
    "Description": "Example description",  
    "PoolCount": 0,  
    "State": "create-in-progress",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "Example name value"  
      }  
    ]  
  }  
}
```

詳細については、「[Amazon VPC IPAM ユーザーガイド](#)」の「[追加のスコープを作成する](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス `create-ipam`](#)」の「」を参照してください。

create-ipam

次の例は、create-ipam を使用する方法を説明しています。

AWS CLI

を作成するには IPAM

次の create-ipam 例では、を作成します IPAM。

(Linux):

```
aws ec2 create-ipam \
  --description "Example description" \
  --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" \
  --tag-specifications 'ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]'
```

(Windows):

```
aws ec2 create-ipam ^
  --description "Example description" ^
  --operating-regions "RegionName=us-east-2" "RegionName=us-west-1" ^
  --tag-specifications ResourceType=ipam,Tags=[{Key=Name,Value=ExampleIPAM}]
```

出力:

```
{
  "Ipam": {
    "OwnerId": "123456789012",
    "IpamId": "ipam-036486dfa6af58ee0",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
    "IpamRegion": "us-east-1",
    "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
    "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
    "ScopeCount": 2,
    "OperatingRegions": [
```



```
{
  {
    "RegionName": "us-east-2"
  },
  {
    "RegionName": "us-west-1"
  },
  {
    "RegionName": "us-east-1"
  }
],
"State": "create-in-progress",
"Tags": [
  {
    "Key": "Name",
    "Value": "ExampleIPAM"
  }
]
}
```

詳細については、「[Amazon ユーザーガイド](#)」の「[の作成IPAM](#)」を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateIpam](#)」の「」を参照してください。

create-key-pair

次のコード例は、create-key-pair を使用する方法を示しています。

AWS CLI

キーペアを作成するには

この例では、MyKeyPair という名前のキーペアが作成されます。

コマンド:

```
aws ec2 create-key-pair --key-name MyKeyPair
```

出力はプライベートキーとキーフィンガープリントASCIIのバージョンです。キーはファイルに保存する必要があります。

詳細については、「AWS コマンドラインインターフェイスユーザーガイド」でキーペアの使用方法を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateKeyPair](#)」の「」を参照してください。

create-launch-template-version

次の例は、create-launch-template-version を使用する方法を説明しています。

AWS CLI

起動テンプレートバージョンを作成するには

この例では、起動テンプレートのバージョン 1 に基づいて新しい起動テンプレートバージョンを作成し、別の AMI ID を指定します。

コマンド:

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123
--version-description WebVersion2 --source-version 1 --launch-template-data
'{"ImageId": "ami-c998b6b2"}'
```

出力:

```
{
  "LaunchTemplateVersion": {
    "VersionDescription": "WebVersion2",
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "WebServers",
    "VersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "LaunchTemplateData": {
      "ImageId": "ami-c998b6b2",
      "InstanceType": "t2.micro",
      "NetworkInterfaces": [
        {
          "Ipv6Addresses": [
            {
              "Ipv6Address": "2001:db8:1234:1a00::123"
            }
          ]
        }
      ],
    }
  }
}
```

```

        "DeviceIndex": 0,
        "SubnetId": "subnet-7b16de0c",
        "AssociatePublicIpAddress": true
      }
    ]
  },
  "DefaultVersion": false,
  "CreateTime": "2017-12-01T13:35:46.000Z"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateLaunchTemplateVersion](#)」の「」を参照してください。

create-launch-template

次のコード例は、create-launch-template を使用する方法を示しています。

AWS CLI

例 1: 起動テンプレートを作成するには

次のcreate-launch-template例では、インスタンスを起動するサブネットを指定する起動テンプレートを作成し、インスタンスにパブリック IP アドレスとIPv6アドレスを割り当て、インスタンスのタグを作成します。

```

aws ec2 create-launch-template \
  --launch-template-name TemplateForWebServer \
  --version-description WebVersion1 \
  --launch-template-data '{"NetworkInterfaces":
[{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Ipv6AddressCount":1,"SubnetId":"subnet-7b16de0c"},
{"AssociatePublicIpAddress":false,"DeviceIndex":1,"Ipv6AddressCount":1,"SubnetId":"subnet-7b16de0c"}]}'

```

出力:

```

{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-01238c059e3466abc",
    "LaunchTemplateName": "TemplateForWebServer",
    "DefaultVersionNumber": 1,

```

```

    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-01-27T09:13:24.000Z"
  }
}

```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「起動テンプレートからのインスタンスの起動」を参照してください。JSON形式のパラメータの引用については、AWS コマンドラインインターフェイスユーザーガイドの「引用文字列」を参照してください。

例 2: Amazon EC2 Auto Scaling の起動テンプレートを作成するには

次のcreate-launch-template例では、複数のタグとブロックデバイスマッピングを含む起動テンプレートを作成し、インスタンスの起動時に追加のEBSボリュームを指定します。Auto Scaling グループGroupsがインスタンスを起動VPCする のセキュリティグループに対応する の値を指定します。Auto Scaling グループのプロパティとして VPCおよび サブネットを指定します。

```

aws ec2 create-launch-template \
  --launch-template-name TemplateForAutoScaling \
  --version-description AutoScalingVersion1 \
  --launch-template-data '{"NetworkInterfaces":
  [{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":
  [{"sg-7c227019,sg-903004f8"},"DeleteOnTermination":true}], "ImageId":"ami-
  b42209de","InstanceType":"m4.large","TagSpecifications":
  [{"ResourceType":"instance","Tags":[{"Key":"environment","Value":"production"},
  {"Key":"purpose","Value":"webserver"}]},{"ResourceType":"volume","Tags":
  [{"Key":"environment","Value":"production"}, {"Key":"cost-
  center","Value":"cc123"}]}],"BlockDeviceMappings":[{"DeviceName":"/dev/sda1","Ebs":
  {"VolumeSize":100}}]}' --region us-east-1

```

出力:

```

{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0123c79c33a54e0abc",
    "LaunchTemplateName": "TemplateForAutoScaling",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-04-30T18:16:06.000Z"
  }
}

```

```
}
```

詳細については、「Amazon Auto Scaling ユーザーガイド」の「Auto Scaling グループの起動テンプレートの作成」を参照してください。EC2 Auto Scaling JSON形式のパラメータの引用については、AWS コマンドラインインターフェイスユーザーガイドの「引用文字列」を参照してください。

例 3: EBSボリュームの暗号化を指定する起動テンプレートを作成するには

次のcreate-launch-template例では、暗号化されていないスナップショットから作成された暗号化されたEBSボリュームを含む起動テンプレートを作成します。また、作成時にボリュームにタグ付けしています。暗号化がデフォルトで無効になっている場合、次の例のように "Encrypted" オプションを指定する必要があります。"KmsKeyId" オプションを使用してカスタマーマネージドを指定する場合はCMK、デフォルトで暗号化が有効になっている場合でも、"Encrypted"オプションを指定する必要があります。

```
aws ec2 create-launch-template \  
  --launch-template-name TemplateForEncryption \  
  --launch-template-data file://config.json
```

config.json の内容:

```
{  
  "BlockDeviceMappings": [  
    {  
      "DeviceName": "/dev/sda1",  
      "Ebs": {  
        "VolumeType": "gp2",  
        "DeleteOnTermination": true,  
        "SnapshotId": "snap-066877671789bd71b",  
        "Encrypted": true,  
        "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/abcd1234-  
a123-456a-a12b-a123b4cd56ef"  
      }  
    }  
  ],  
  "ImageId": "ami-00068cd7555f543d5",  
  "InstanceType": "c5.large",  
  "TagSpecifications": [  
    {  
      "ResourceType": "volume",
```

```
    "Tags": [
      {
        "Key": "encrypted",
        "Value": "yes"
      }
    ]
  }
]
```

出力:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
    "LaunchTemplateName": "TemplateForEncryption",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2020-01-07T19:08:36.000Z"
  }
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「スナップショットからの Amazon EBS ボリュームの復元」および「デフォルトでの暗号化」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateLaunchTemplate](#)」の「」を参照してください。

create-local-gateway-route-table-virtual-interface-group-association

次の例は、create-local-gateway-route-table-virtual-interface-group-association を使用する方法を説明しています。

AWS CLI

ローカルゲートウェイルートテーブルを仮想インターフェイス (VIFs) グループに関連付けるには

次のcreate-local-gateway-route-table-virtual-interface-group-association例では、指定されたローカルゲートウェイルートテーブルとVIFグループ間の関連付けを作成します。

```
aws ec2 create-local-gateway-route-table-virtual-interface-group-association \  
--local-gateway-route-table-id lgw-rtb-exampleidabcd1234 \  
--local-gateway-virtual-interface-group-id lgw-vif-grp-exampleid0123abcd
```

出力:

```
{  
  "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {  
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-  
assoc-exampleid12345678",  
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",  
    "LocalGatewayId": "lgw-exampleid11223344",  
    "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",  
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-  
gateway-route-table/lgw-rtb-exampleidabcd1234",  
    "OwnerId": "111122223333",  
    "State": "pending",  
    "Tags": []  
  }  
}
```

詳細については、[VIF「Outposts ユーザーガイド」の「グループ関連付けAWS」](#)を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスCreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation](#)」の「」を参照してください。

create-local-gateway-route-table-vpc-association

次のコード例は、create-local-gateway-route-table-vpc-association を使用する方法を示しています。

AWS CLI

をルートテーブルVPCに関連付けるには

次のcreate-local-gateway-route-table-vpc-association例では、指定された を指定されたローカルゲートウェイルートテーブルVPCに関連付けます。

```
aws ec2 create-local-gateway-route-table-vpc-association \  

```

```
--local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \  
--vpc-id vpc-07ef66ac71EXAMPLE
```

出力:

```
{  
  "LocalGatewayRouteTableVpcAssociation": {  
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",  
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",  
    "VpcId": "vpc-07ef66ac71EXAMPLE",  
    "State": "associated"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateLocalGatewayRouteTableVpcAssociation](#)」の「」を参照してください。

create-local-gateway-route-table

次のコード例は、`create-local-gateway-route-table` を使用する方法を示しています。

AWS CLI

ローカルゲートウェイルートテーブルを作成するには

次の`create-local-gateway-route-table`例では、ダイレクトルーティングモードでローカルゲートウェイVPCルートテーブルを作成します。

```
aws ec2 create-local-gateway-route-table \  
--local-gateway-id lgw-1a2b3c4d5e6f7g8h9 \  
--mode direct-vpc-routing
```

出力:

```
{  
  "LocalGatewayRouteTable": {  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-gateway-route-table/lgw-rtb-abcdefg1234567890",  
  }  
}
```



```
    "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",
    "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-021345abcdef67890",
    "OwnerId": "111122223333",
    "State": "pending",
    "Tags": [],
    "Mode": "direct-vpc-routing"
  }
}
```

詳細については、「AWS Outposts ユーザーガイド」の「[ローカルゲートウェイテーブル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLocalGatewayRouteTable](#)」の「」を参照してください。

create-local-gateway-route

次の例は、create-local-gateway-route を使用する方法を説明しています。

AWS CLI

ローカルゲートウェイルートテーブルの静的ルートを作成するには

次のcreate-local-gateway-route例では、指定されたローカルゲートウェイルートテーブルに指定されたルートを作成します。

```
aws ec2 create-local-gateway-route \
  --destination-cidr-block 0.0.0.0/0 \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

出力:

```
{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
    "Type": "static",
    "State": "deleted",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
  }
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLocalGatewayRoute](#)」の「」を参照してください。

create-managed-prefix-list

次の例は、create-managed-prefix-list を使用する方法を説明しています。

AWS CLI

プレフィックスリストを作成するには

次のcreate-managed-prefix-list例では、最大 10 個のエントリを含むIPv4プレフィックスリストを作成し、プレフィックスリストに 2 個のエントリを作成します。

```
aws ec2 create-managed-prefix-list \  
  --address-family IPv4 \  
  --max-entries 10 \  
  --entries Cidr=10.0.0.0/16,Description=vpc-a Cidr=10.2.0.0/16,Description=vpc-b \  
 \  
  --prefix-list-name vpc-cidrs
```

出力:

```
{  
  "PrefixList": {  
    "PrefixListId": "pl-0123456abcabcabc1",  
    "AddressFamily": "IPv4",  
    "State": "create-in-progress",  
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-0123456abcabcabc1",  
    "PrefixListName": "vpc-cidrs",  
    "MaxEntries": 10,  
    "Version": 1,  
    "Tags": [],  
    "OwnerId": "123456789012"  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の「[マネージドプレフィックスリスト](#)」を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateManagedPrefixList](#)」の「」を参照してください。

create-nat-gateway

次の例は、create-nat-gateway を使用する方法を説明しています。

AWS CLI

例 1: パブリックNATゲートウェイを作成するには

次のcreate-nat-gateway例では、指定されたサブネットにパブリックNATゲートウェイを作成し、Elastic IP アドレスを指定された割り当て ID に関連付けます。パブリックNATゲートウェイを作成するときは、Elastic IP アドレスを関連付ける必要があります。

```
aws ec2 create-nat-gateway \  
  --subnet-id subnet-0250c25a1fEXAMPLE \  
  --allocation-id eipalloc-09ad461b0dEXAMPLE
```

出力:

```
{  
  "NatGateway": {  
    "CreateTime": "2021-12-01T22:22:38.000Z",  
    "NatGatewayAddresses": [  
      {  
        "AllocationId": "eipalloc-09ad461b0dEXAMPLE"  
      }  
    ],  
    "NatGatewayId": "nat-0c61bf8a12EXAMPLE",  
    "State": "pending",  
    "SubnetId": "subnet-0250c25a1fEXAMPLE",  
    "VpcId": "vpc-0a60eb65b4EXAMPLE",  
    "ConnectivityType": "public"  
  }  
}
```

詳細については、「Amazon VPCユーザーガイド[NAT](#)」の「[ゲートウェイ](#)」を参照してください。

例 2: プライベートNATゲートウェイを作成するには

次のcreate-nat-gateway例では、指定されたサブネットにプライベートNATゲートウェイを作成します。プライベートNATゲートウェイには、関連付けられた Elastic IP アドレスがありません。

```
aws ec2 create-nat-gateway \  
  --subnet-id subnet-0250c25a1fEXAMPLE \  
  --connectivity-type private
```

出力:

```
{  
  "NatGateway": {  
    "CreateTime": "2021-12-01T22:26:00.000Z",  
    "NatGatewayAddresses": [  
      {}  
    ],  
    "NatGatewayId": "nat-011b568379EXAMPLE",  
    "State": "pending",  
    "SubnetId": "subnet-0250c25a1fEXAMPLE",  
    "VpcId": "vpc-0a60eb65b4EXAMPLE",  
    "ConnectivityType": "private"  
  }  
}
```

詳細については、「Amazon VPCユーザーガイド [NAT](#)」の「[ゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateNatGateway](#)」の「」を参照してください。

create-network-acl-entry

次の例は、create-network-acl-entry を使用する方法を説明しています。

AWS CLI

ネットワークACLエントリを作成するには

この例では、指定されたネットワークのエントリを作成しますACL。このルールでは、UDPポート 53 () 上の任意のIPv4アドレス (0.0.0.0/0DNS) から、関連するサブネットへの進入トラフィックを許可します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 0.0.0.0/0 --rule-action allow
```

この例では、ポート TCP80 () 上の任意のIPv6アドレス (::/0) からの進入トラフィックACLを許可する、指定されたネットワークのルールを作成しますHTTP。

コマンド:

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 120 --protocol tcp --port-range From=80,To=80 --ipv6-cidr-block ::/0 --rule-action allow
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNetworkAclEntry](#)」の「」を参照してください。

create-network-acl

次の例は、create-network-acl を使用する方法を説明しています。

AWS CLI

ネットワークを作成するには ACL

この例では、指定された ACLのネットワークを作成しますVPC。

コマンド:

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

出力:

```
{
  "NetworkAcl": {
    "Associations": [],
    "NetworkAclId": "acl-5fb85d36",
    "VpcId": "vpc-a01106c2",
    "Tags": [],
    "Entries": [
```

```
{
  "CidrBlock": "0.0.0.0/0",
  "RuleNumber": 32767,
  "Protocol": "-1",
  "Egress": true,
  "RuleAction": "deny"
},
{
  "CidrBlock": "0.0.0.0/0",
  "RuleNumber": 32767,
  "Protocol": "-1",
  "Egress": false,
  "RuleAction": "deny"
}
],
"IsDefault": false
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNetworkAcl](#)」の「」を参照してください。

create-network-insights-access-scope

次の例は、create-network-insights-access-scope を使用する方法を説明しています。

AWS CLI

ネットワークアクセススコープを作成するには

次のcreate-network-insights-access-scope例では、ネットワークアクセススコープを作成します。

```
aws ec2 create-network-insights-access-scope \
  --cli-input-json file://access-scope-file.json
```

access-scope-file.json の内容:

```
{
  "MatchPaths": [
    {
      "Source": {
```

```

        "ResourceStatement": {
            "Resources": [
                "vpc-abcd12e3"
            ]
        }
    ],
    "ExcludePaths": [
        {
            "Source": {
                "ResourceStatement": {
                    "ResourceTypes": [
                        "AWS::EC2::InternetGateway"
                    ]
                }
            }
        }
    ]
}

```

出力:

```

{
  "NetworkInsightsAccessScope": {
    "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
    "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789abc01234",
    "CreateDate": "2022-01-25T19:20:28.796000+00:00",
    "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
  },
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "Resources": [
              "vpc-abcd12e3"
            ]
          }
        }
      }
    ]
  }
}

```

```
    ],
    "ExcludePaths": [
      {
        "Source": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      }
    ]
  }
}
```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateNetworkInsightsAccessScope](#)」の「」を参照してください。

create-network-insights-path

次のコード例は、create-network-insights-path を使用する方法を示しています。

AWS CLI

パスを作成するには

次のcreate-network-insights-path例では、パスを作成します。送信元は指定されたインターネットゲートウェイで、送信先は指定されたEC2インスタンスです。指定されたプロトコルとポートを使用して送信先に到達できるかどうかを確認するには、start-network-insights-analysis コマンドを使用してパスを分析します。

```
aws ec2 create-network-insights-path \
  --source igw-0797cccdc9d73b0e5 \
  --destination i-0495d385ad28331c7 \
  --destination-port 22 \
  --protocol TCP
```

出力:


```
{
  "NetworkInsightsPaths": {
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
    "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
    "CreateDate": "2021-01-20T22:43:46.933Z",
    "Source": "igw-0797cccdc9d73b0e5",
    "Destination": "i-0495d385ad28331c7",
    "Protocol": "tcp"
  }
}
```

詳細については、「Reachability Analyzer ガイド」の「[の使用開始 AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNetworkInsightsPath](#)」の「」を参照してください。

create-network-interface-permission

次のコード例は、create-network-interface-permission を使用する方法を示しています。

AWS CLI

ネットワークインターフェイスのアクセス許可を作成するには

この例では、ネットワークインターフェイスをインスタンスにアタッチ123456789012するアクセス許可をアカウントに付与eni-1a2b3c4dします。

コマンド:

```
aws ec2 create-network-interface-permission --network-interface-id eni-1a2b3c4d --
aws-account-id 123456789012 --permission INSTANCE-ATTACH
```

出力:

```
{
  "InterfacePermission": {
    "PermissionState": {
      "State": "GRANTED"
    },
    "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
```

```
"NetworkInterfaceId": "eni-1a2b3c4d",  
"Permission": "INSTANCE-ATTACH",  
"AwsAccountId": "123456789012"  
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNetworkInterfacePermission](#)」の「」を参照してください。

create-network-interface

次のコード例は、create-network-interface を使用する方法を示しています。

AWS CLI

例 1: ネットワークインターフェイスのIPv4アドレスを指定するには

次のcreate-network-interface例では、指定したプライマリIPv4アドレスを持つ、指定したサブネットのネットワークインターフェイスを作成します。

```
aws ec2 create-network-interface \  
  --subnet-id subnet-00a24d0d67acf6333 \  
  --description "my network interface" \  
  --groups sg-09dfba7ed20cda78b \  
  --private-ip-address 10.0.8.17
```

出力:

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "Description": "my network interface",  
    "Groups": [  
      {  
        "GroupName": "my-security-group",  
        "GroupId": "sg-09dfba7ed20cda78b"  
      }  
    ],  
    "InterfaceType": "interface",  
    "Ipv6Addresses": [],  
    "MacAddress": "06:6a:0f:9a:49:37",  
    "NetworkInterfaceId": "eni-0492b355f0cf3b3f8",
```

```

    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
    "PrivateIpAddress": "10.0.8.17",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-8-17.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.17"
      }
    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeeb9d57b"
  }
}

```

例 2: IPv4 アドレスとIPv6アドレスを持つネットワークインターフェイスを作成するには

次のcreate-network-interface例では、IPv4アドレスと Amazon によって選択されたIPv6アドレスを使用して、指定されたサブネットのネットワークインターフェイスを作成します EC2。

```

aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my dual stack network interface" \
  --ipv6-address-count 1 \
  --groups sg-09dfba7ed20cda78b

```

出力:

```

{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "Description": "my dual stack network interface",
    "Groups": [
      {
        "GroupName": "my-security-group",
        "GroupId": "sg-09dfba7ed20cda78b"
      }
    ]
  }
}

```

```

    }
  ],
  "InterfaceType": "interface",
  "Ipv6Addresses": [
    {
      "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7",
      "IsPrimaryIpv6": false
    }
  ],
  "MacAddress": "06:b8:68:d2:b2:2d",
  "NetworkInterfaceId": "eni-05da417453f9a84bf",
  "OwnerId": "123456789012",
  "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
  "PrivateIpAddress": "10.0.8.18",
  "PrivateIpAddresses": [
    {
      "Primary": true,
      "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
      "PrivateIpAddress": "10.0.8.18"
    }
  ],
  "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
  "RequesterManaged": false,
  "SourceDestCheck": true,
  "Status": "pending",
  "SubnetId": "subnet-00a24d0d67acf6333",
  "TagSet": [],
  "VpcId": "vpc-02723a0feeb9d57b",
  "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7"
}
}

```

例 3: 接続追跡設定オプションを使用してネットワークインターフェイスを作成するには

次のcreate-network-interface例では、ネットワークインターフェイスを作成し、アイドル状態の接続追跡タイムアウトを設定します。

```

aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --groups sg-02e57dbcfe0331c1b \
  --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60

```

出力:

```
{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "ConnectionTrackingConfiguration": {
      "TcpEstablishedTimeout": 86400,
      "UdpTimeout": 60
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "my-security-group",
        "GroupId": "sg-02e57dbcfe0331c1b"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [],
    "MacAddress": "06:4c:53:de:6d:91",
    "NetworkInterfaceId": "eni-0c133586e08903d0b",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
    "PrivateIpAddress": "10.0.8.94",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.94"
      }
    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeeb9d57b"
  }
}
```

例 4: Elastic Fabric Adapter を作成するには

次の `create-network-interface` 例では、を作成しますEFA。

```
aws ec2 create-network-interface \
```

```
--interface-type efa \  
--subnet-id subnet-00a24d0d67acf6333 \  
--description "my efa" \  
--groups sg-02e57dbcfe0331c1b
```

出力:

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "Description": "my efa",  
    "Groups": [  
      {  
        "GroupName": "my-efa-sg",  
        "GroupId": "sg-02e57dbcfe0331c1b"  
      }  
    ],  
    "InterfaceType": "efa",  
    "Ipv6Addresses": [],  
    "MacAddress": "06:d7:a4:f7:4d:57",  
    "NetworkInterfaceId": "eni-034acc2885e862b65",  
    "OwnerId": "123456789012",  
    "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",  
    "PrivateIpAddress": "10.0.8.180",  
    "PrivateIpAddresses": [  
      {  
        "Primary": true,  
        "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",  
        "PrivateIpAddress": "10.0.8.180"  
      }  
    ],  
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",  
    "RequesterManaged": false,  
    "SourceDestCheck": true,  
    "Status": "pending",  
    "SubnetId": "subnet-00a24d0d67acf6333",  
    "TagSet": [],  
    "VpcId": "vpc-02723a0feeeb9d57b"  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の「[Elastic Network Interface](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNetworkInterface](#)」の「」を参照してください。

create-placement-group

次の例は、create-placement-group を使用する方法を説明しています。

AWS CLI

プレースメントグループを作成するには

この例では、指定した名前のプレースメントグループを作成します。

コマンド:

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

パーティションプレースメントグループを作成するには

この例では、という名前のパーティション配置グループを 5 つのパーティションHDFS-Group-Aで作成します。

コマンド:

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --partition-count 5
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePlacementGroup](#)」の「」を参照してください。

create-replace-root-volume-task

次のコード例は、create-replace-root-volume-task を使用する方法を示しています。

AWS CLI

例 1: ルートボリュームを初期起動状態に復元するには

次のcreate-replace-root-volume-task例では、インスタンス i-0123456789abcdefa のルートボリュームを初期起動状態に復元します。

```
aws ec2 create-replace-root-volume-task \  
--instance-id i-0123456789abcdefa
```

出力:

```
{  
  "ReplaceRootVolumeTask":  
  {  
    "InstanceId": "i-0123456789abcdefa",  
    "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",  
    "TaskState": "pending",  
    "StartTime": "2022-03-14T15:06:38Z",  
    "Tags": []  
  }  
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[ルートボリュームを置き換える](#)」を参照してください。

例 2: ルートボリュームを特定のスナップショットに復元するには

次のcreate-replace-root-volume-task例では、インスタンス i-0123456789abcdefa のルートボリュームをスナップショット snap-0abcdef1234567890 に復元します。

```
aws ec2 create-replace-root-volume-task \  
--instance-id i-0123456789abcdefa \  
--snapshot-id snap-0abcdef1234567890
```

出力:

```
{  
  "ReplaceRootVolumeTask":  
  {  
    "InstanceId": "i-0123456789abcdefa",  
    "ReplaceRootVolumeTaskId": "replacevol-0555566667777abcd",  
    "TaskState": "pending",  
    "StartTime": "2022-03-14T15:16:28Z",  
    "Tags": []  
  }  
}
```


詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[ルートボリュームを置き換える](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateReplaceRootVolumeTask](#)」の「」を参照してください。

create-reserved-instances-listing

次のコード例は、create-reserved-instances-listing を使用する方法を示しています。

AWS CLI

リザーブドインスタンスをリザーブドインスタンスマーケットプレイスに一覧表示するには

次のcreate-reserved-instances-listing例では、Reserved Instance Marketplace で指定されたリザーブドインスタンスのリストを作成します。

```
aws ec2 create-reserved-instances-listing \  
  --reserved-instances-id 5ec28771-05ff-4b9b-aa31-9e57dexample \  
  --instance-count 3 \  
  --price-schedules CurrencyCode=USD,Price=25.50 \  
  --client-token 550e8400-e29b-41d4-a716-446655440000
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateReservedInstancesListing](#)」の「」を参照してください。

create-restore-image-task

次のコード例は、create-restore-image-task を使用する方法を示しています。

AWS CLI

S3 バケットAMIから を復元するには

次のcreate-restore-image-task例では、S3 バケットAMIから を復元します。describe-store-image-tasks 出力S3objectKey `` and ``Bucketからの値を使用し、のオブジェクトキーAMIと、がコピーAMIされた S3 バケットの名前を指定し、復元された の名前を指定しますAMI。この名前は、このアカウントのリージョンAMIsで に一意である必要があります。復元された AMIは新しい AMI ID を受け取ります。

```
aws ec2 create-restore-image-task \  
  --source-image-id ami-0123456789abcdef0 \  
  --target-image-id ami-0123456789abcdef0 \  
  --source-image-key source-image-key \  
  --source-image-bucket source-image-bucket \  
  --target-image-key target-image-key \  
  --target-image-bucket target-image-bucket \  
  --client-token client-token
```

```
--object-key ami-1234567890abcdef0.bin \  
--bucket my-ami-bucket \  
--name "New AMI Name"
```

出力:

```
{  
  "ImageId": "ami-0eab20fe36f83e1a8"  
}
```

S3 AMIを使用した の保存と復元の詳細については、「Amazon EC2ユーザーガイド」のS3 <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-store-restore.html>> AMIを使用した の保存と復元」を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[CreateRestoreImageTask](#)」の「」を参照してください。

create-route-table

次の例は、create-route-table を使用する方法を説明しています。

AWS CLI

ルートテーブルを作成するには

この例では、指定された のルートテーブルを作成しますVPC。

コマンド:

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

出力:

```
{  
  "RouteTable": {  
    "Associations": [],  
    "RouteTableId": "rtb-22574640",  
    "VpcId": "vpc-a01106c2",  
    "PropagatingVgws": [],  
    "Tags": [],  
    "Routes": [  
      {
```

```
        "GatewayId": "local",
        "DestinationCidrBlock": "10.0.0.0/16",
        "State": "active"
    }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateRouteTable](#)」の「」を参照してください。

create-route

次の例は、create-route を使用する方法を説明しています。

AWS CLI

ルートを作成するには

この例では、指定されたルートテーブルのルートを作成します。ルートはすべてのIPv4トラフィック (0.0.0.0/0) に一致し、指定されたインターネットゲートウェイにルーティングされます。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

この例では、ルートテーブル `rtb-g8ff4ea2` にルートを作成します。ルートはIPv4CIDRブロック `10.0.0.0/16` のトラフィックに一致し、VPCピアリング接続 `pcx-111aaa22` にルーティングされます。このルートにより、ピアリング接続VPCでVPCトラフィックをピアに誘導できます。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

この例では、すべてのIPv6トラフィック (`::/0`) に一致するルートを指定されたルートテーブルに作成し、指定されたエグレス専用インターネットゲートウェイにルーティングします。

コマンド:

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-block ::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRoute](#)」の「」を参照してください。

create-security-group

次の例は、create-security-group を使用する方法を説明しています。

AWS CLI

EC2-Classic のセキュリティグループを作成するには

この例では、MySecurityGroup という名前のセキュリティグループが作成されます。

コマンド:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
```

出力:

```
{
  "GroupId": "sg-903004f8"
}
```

EC2- のセキュリティグループを作成するにはVPC

この例では、指定された MySecurityGroup に という名前のセキュリティグループを作成しますVPC。

コマンド:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group" --vpc-id vpc-1a2b3c4d
```

出力:

```
{
  "GroupId": "sg-903004f8"
}
```

詳細については、「AWS コマンドラインインターフェイスユーザーガイド」でセキュリティグループの使用方法を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSecurityGroup](#)」の「」を参照してください。

create-snapshot

次の例は、create-snapshot を使用する方法を説明しています。

AWS CLI

スナップショットを作成するには

このコマンド例では、ボリューム ID が vol-1234567890abcdef0 で、スナップショットを識別するための簡単な説明を持つボリュームのスナップショットを作成します。

コマンド:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my root volume snapshot"
```

出力:

```
{
  "Description": "This is my root volume snapshot",
  "Tags": [],
  "Encrypted": false,
  "VolumeId": "vol-1234567890abcdef0",
  "State": "pending",
  "VolumeSize": 8,
  "StartTime": "2018-02-28T21:06:01.000Z",
  "Progress": "",
  "OwnerId": "012345678910",
  "SnapshotId": "snap-066877671789bd71b"
}
```

タグを使用してスナップショットを作成するには

この例では、スナップショットを作成し、 `purpose=prod` と `costcenter=123` の 2 つのタグを適用します。

コマンド:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description 'Prod backup' --tag-specifications 'ResourceType=snapshot,Tags=[{Key=purpose,Value=prod},{Key=costcenter,Value=123}]'
```

出力:

```
{
  "Description": "Prod backup",
  "Tags": [
    {
      "Value": "prod",
      "Key": "purpose"
    },
    {
      "Value": "123",
      "Key": "costcenter"
    }
  ],
  "Encrypted": false,
  "VolumeId": "vol-1234567890abcdef0",
  "State": "pending",
  "VolumeSize": 8,
  "StartTime": "2018-02-28T21:06:06.000Z",
  "Progress": "",
  "OwnerId": "012345678910",
  "SnapshotId": "snap-09ed24a70bc19bbe4"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSnapshot](#)」の「」を参照してください。

create-snapshots

次のコード例は、`create-snapshots` を使用する方法を示しています。

AWS CLI

例 1: マルチボリュームスナップショットを作成するには

次のcreate-snapshots例では、指定されたインスタンスにアタッチされたすべてのボリュームのスナップショットを作成します。

```
aws ec2 create-snapshots \  
  --instance-specification InstanceId=i-1234567890abcdef0 \  
  --description "This is snapshot of a volume from my-instance"
```

出力:

```
{  
  "Snapshots": [  
    {  
      "Description": "This is a snapshot of a volume from my-instance",  
      "Tags": [],  
      "Encrypted": false,  
      "VolumeId": "vol-0a01d2d5a34697479",  
      "State": "pending",  
      "VolumeSize": 16,  
      "StartTime": "2019-08-05T16:58:19.000Z",  
      "Progress": "",  
      "OwnerId": "123456789012",  
      "SnapshotId": "snap-07f30e3909aa0045e"  
    },  
    {  
      "Description": "This is a snapshot of a volume from my-instance",  
      "Tags": [],  
      "Encrypted": false,  
      "VolumeId": "vol-02d0d4947008cb1a2",  
      "State": "pending",  
      "VolumeSize": 20,  
      "StartTime": "2019-08-05T16:58:19.000Z",  
      "Progress": "",  
      "OwnerId": "123456789012",  
      "SnapshotId": "snap-0ec20b602264aad48"  
    },  
    ...  
  ]  
}
```

例 2: ソースボリュームからタグを使用してマルチボリュームスナップショットを作成するには

次のcreate-snapshots例では、指定されたインスタンスにアタッチされたすべてのボリュームのスナップショットを作成し、各ボリュームから対応するスナップショットにタグをコピーします。

```
aws ec2 create-snapshots \  
  --instance-specification InstanceId=i-1234567890abcdef0 \  
  --copy-tags-from-source volume \  
  --description "This is snapshot of a volume from my-instance"
```

出力:

```
{  
  "Snapshots": [  
    {  
      "Description": "This is a snapshot of a volume from my-instance",  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "my-volume"  
        }  
      ],  
      "Encrypted": false,  
      "VolumeId": "vol-02d0d4947008cb1a2",  
      "State": "pending",  
      "VolumeSize": 20,  
      "StartTime": "2019-08-05T16:53:04.000Z",  
      "Progress": "",  
      "OwnerId": "123456789012",  
      "SnapshotId": "snap-053bfaeb821a458dd"  
    }  
    ...  
  ]  
}
```

例 3: ルートボリュームを含まないマルチボリュームスナップショットを作成するには

次のcreate-snapshots例では、ルートボリュームを除く、指定されたインスタンスにアタッチされたすべてのボリュームのスナップショットを作成します。

```
aws ec2 create-snapshots \  
  --instance-specification InstanceId=i-1234567890abcdef0 \  
  --copy-tags-from-source volume \  
  --description "This is snapshot of a volume from my-instance"
```



```
--instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true
```

出力例については、例 1 を参照してください。

例 4: マルチボリュームスナップショットを作成し、タグを追加するには

次のcreate-snapshots例では、指定されたインスタンスにアタッチされたすべてのボリュームのスナップショットを作成し、各スナップショットに 2 つのタグを追加します。

```
aws ec2 create-snapshots \  
  --instance-specification InstanceId=i-1234567890abcdef0 \  
  --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=backup},  
{Key=costcenter,Value=123}]'
```

出力例については、例 1 を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSnapshots](#)」の「」を参照してください。

create-spot-datafeed-subscription

次の例は、create-spot-datafeed-subscription を使用する方法を説明しています。

AWS CLI

スポットインスタンスのデータフィードを作成するには

次のcreate-spot-datafeed-subscription例では、スポットインスタンスデータフィードを作成します。

```
aws ec2 create-spot-datafeed-subscription \  
  --bucket my-bucket \  
  --prefix spot-data-feed
```

出力:

```
{  
  "SpotDatafeedSubscription": {  
    "Bucket": "my-bucket",  
    "OwnerId": "123456789012",  
    "Prefix": "spot-data-feed",
```

```
    "State": "Active"
  }
}
```

データフィードは、指定した Amazon S3 バケットに保存されます。このデータフィードのファイル名の形式は次のとおりです。

```
my-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-HH.n.abcd1234.gz
```

詳細については、「Linux [インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド](#)」の「[スポットインスタンスデータフィード](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateSpotDatafeedSubscription](#)」の「」を参照してください。

create-store-image-task

次の例は、create-store-image-task を使用する方法を説明しています。

AWS CLI

を S3 AMI バケットに保存するには

次のcreate-store-image-task例では、を S3 バケットAMIに保存します。の ID AMIと、を保存する S3 バケットの名前を指定しますAMI。

```
aws ec2 create-store-image-task \  
  --image-id ami-1234567890abcdef0 \  
  --bucket my-ami-bucket
```

出力:

```
{  
  "ObjectKey": "ami-1234567890abcdef0.bin"  
}
```

詳細については、「Amazon EC2ユーザーガイド」の[S3 AMIを使用して を保存および復元する](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateStoreImageTask](#)」の「」を参照してください。

create-subnet-cidr-reservation

次の例は、create-subnet-cidr-reservation を使用する方法を説明しています。

AWS CLI

サブネットCIDR予約を作成するには

次のcreate-subnet-cidr-reservation例では、指定されたサブネットとCIDR範囲のサブネットCIDR予約を作成します。

```
aws ec2 create-subnet-cidr-reservation \
  --subnet-id subnet-03c51e2eEXAMPLE \
  --reservation-type prefix \
  --cidr 10.1.0.20/26
```

出力:

```
{
  "SubnetCidrReservation": {
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",
    "Cidr": "10.1.0.16/28",
    "ReservationType": "prefix",
    "OwnerId": "123456789012"
  }
}
```

詳細については、「Amazon ユーザーガイド」の [「サブネットCIDR予約」](#) を参照してください。

VPC

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSubnetCidrReservation](#)」の「」を参照してください。

create-subnet

次のコード例は、create-subnet を使用する方法を示しています。

AWS CLI

例 1: IPv4CIDRブロックのみを使用してサブネットを作成するには

次のcreate-subnet例では、指定されたIPv4CIDRブロックVPCで指定された にサブネットを作成します。

```
aws ec2 create-subnet \  
  --vpc-id vpc-081ec835f3EXAMPLE \  
  --cidr-block 10.0.0.0/24 \  
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-subnet}]
```

出力:

```
{  
  "Subnet": {  
    "AvailabilityZone": "us-west-2a",  
    "AvailabilityZoneId": "usw2-az2",  
    "AvailableIpAddressCount": 251,  
    "CidrBlock": "10.0.0.0/24",  
    "DefaultForAz": false,  
    "MapPublicIpOnLaunch": false,  
    "State": "available",  
    "SubnetId": "subnet-0e99b93155EXAMPLE",  
    "VpcId": "vpc-081ec835f3EXAMPLE",  
    "OwnerId": "123456789012",  
    "AssignIpv6AddressOnCreation": false,  
    "Ipv6CidrBlockAssociationSet": [],  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-ipv4-only-subnet"  
      }  
    ],  
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0e99b93155EXAMPLE"  
  }  
}
```

例 2: IPv4と IPv6CIDRブロックの両方を使用してサブネットを作成するには

次のcreate-subnet例では、指定された IPv4 および IPv6CIDRブロックVPCを使用して、指定された にサブネットを作成します。

```
aws ec2 create-subnet \  
  --vpc-id vpc-081ec835f3EXAMPLE \  
  --cidr-block 10.0.0.0/24 \  
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-subnet}]
```

```
--vpc-id vpc-081ec835f3EXAMPLE \  
--cidr-block 10.0.0.0/24 \  
--ipv6-cidr-block 2600:1f16:cfe:3660::/64 \  
--tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-ipv6-  
subnet}]
```

出力:

```
{  
  "Subnet": {  
    "AvailabilityZone": "us-west-2a",  
    "AvailabilityZoneId": "usw2-az2",  
    "AvailableIpAddressCount": 251,  
    "CidrBlock": "10.0.0.0/24",  
    "DefaultForAz": false,  
    "MapPublicIpOnLaunch": false,  
    "State": "available",  
    "SubnetId": "subnet-0736441d38EXAMPLE",  
    "VpcId": "vpc-081ec835f3EXAMPLE",  
    "OwnerId": "123456789012",  
    "AssignIpv6AddressOnCreation": false,  
    "Ipv6CidrBlockAssociationSet": [  
      {  
        "AssociationId": "subnet-cidr-assoc-06c5f904499fcc623",  
        "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",  
        "Ipv6CidrBlockState": {  
          "State": "associating"  
        }  
      }  
    ],  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-ipv4-ipv6-subnet"  
      }  
    ],  
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0736441d38EXAMPLE"  
  }  
}
```

例 3: IPv6CIDRブロックのみを使用してサブネットを作成するには

次のcreate-subnet例では、指定されたIPv6CIDRブロックVPCで指定された にサブネットを作成します。

```
aws ec2 create-subnet \  
  --vpc-id vpc-081ec835f3EXAMPLE \  
  --ipv6-native \  
  --ipv6-cidr-block 2600:1f16:115:200::/64 \  
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv6-only-  
subnet}]
```

出力:

```
{  
  "Subnet": {  
    "AvailabilityZone": "us-west-2a",  
    "AvailabilityZoneId": "usw2-az2",  
    "AvailableIpAddressCount": 0,  
    "DefaultForAz": false,  
    "MapPublicIpOnLaunch": false,  
    "State": "available",  
    "SubnetId": "subnet-03f720e7deEXAMPLE",  
    "VpcId": "vpc-081ec835f3EXAMPLE",  
    "OwnerId": "123456789012",  
    "AssignIpv6AddressOnCreation": true,  
    "Ipv6CidrBlockAssociationSet": [  
      {  
        "AssociationId": "subnet-cidr-assoc-01ef639edde556709",  
        "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",  
        "Ipv6CidrBlockState": {  
          "State": "associating"  
        }  
      }  
    ],  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-ipv6-only-subnet"  
      }  
    ],  
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-03f720e7deEXAMPLE"  
  }  
}
```

```
}
```

詳細については、「Amazon VPCユーザーガイド」の[VPCs「とサブネット」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateSubnet](#)」の「」を参照してください。

create-tags

次の例は、create-tags を使用する方法を説明しています。

AWS CLI

例 1: リソースにタグを追加するには

次のcreate-tags例では、指定されたイメージStack=productionに タグを追加するか、タグキーAMIが である の既存のタグを上書きしますStack。

```
aws ec2 create-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags Key=Stack,Value=production
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の[「これはトピックタイトルです」](#)を参照してください。

例 2: 複数のリソースにタグを追加するには

次のcreate-tags例では、AMIと インスタンスに 2 つのタグを追加 (または上書き) します。一方のタグでは、キー (webserver) はありますが値はありません (値は空文字列に設定されています)。もう一方のタグにはキー (stack) と値 (Production) があります。

```
aws ec2 create-tags \  
  --resources ami-1a2b3c4d i-1234567890abcdef0 \  
  --tags Key=webserver,Value= Key=stack,Value=Production
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の[「これはトピックタイトルです」](#)を参照してください。

例 3: 特殊文字を含むタグを追加するには

次の `create-tags` の例では、インスタンスにタグ `[Group]=test` を追加します。角括弧 (`[]`) は特殊文字であり、エスケープする必要があります。以下の例でも、各環境に適した行継続文字を使用しています。

Windows を使用している場合、特殊文字を含む要素を二重引用符 (`"`) で囲み、各二重引用符の前にバックスラッシュ (`\`) を付けます。

```
aws ec2 create-tags ^
  --resources i-1234567890abcdef0 ^
  --tags Key="\[Group]",Value=test
```

Windows を使用している場合は PowerShell、次のように、要素に特殊文字を含む値を二重引用符 (`"`) で囲み、各二重引用符の前にバックスラッシュ (`\`) を付けてから、キーと値の構造全体を一重引用符 (`'`) で囲みます。

```
aws ec2 create-tags `
  --resources i-1234567890abcdef0 `
  --tags 'Key="\[Group]",Value=test'
```

Linux または OS X を使用している場合は、次のように特殊文字を含む要素を二重引用符 (`"`) で囲んだ後、キーと値の構造全体を一重引用符 (`'`) で囲みます。

```
aws ec2 create-tags \
  --resources i-1234567890abcdef0 \
  --tags 'Key="[Group]",Value=test'
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[これはトピックタイトルです](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateTags](#)」の「」を参照してください。

create-traffic-mirror-filter-rule

次の例は、`create-traffic-mirror-filter-rule` を使用する方法を説明しています。

AWS CLI

受信TCPトラフィックのフィルタールールを作成するには

次のcreate-traffic-mirror-filter-rule例では、すべての受信TCPトラフィックをミラーリングするために使用できるルールを作成します。このコマンドを実行する前に、create-traffic-mirror-filterを使用して Traffic Mirror フィルターを作成します。

```
aws ec2 create-traffic-mirror-filter-rule \  
  --description "TCP Rule" \  
  --destination-cidr-block 0.0.0.0/0 \  
  --protocol 6 \  
  --rule-action accept \  
  --rule-number 1 \  
  --source-cidr-block 0.0.0.0/0 \  
  --traffic-direction ingress \  
  --traffic-mirror-filter-id tmf-04812ff784b25ae67
```

出力:

```
{  
  "TrafficMirrorFilterRule": {  
    "DestinationCidrBlock": "0.0.0.0/0",  
    "TrafficMirrorFilterId": "tmf-04812ff784b25ae67",  
    "TrafficMirrorFilterRuleId": "tmfr-02d20d996673f3732",  
    "SourceCidrBlock": "0.0.0.0/0",  
    "TrafficDirection": "ingress",  
    "Description": "TCP Rule",  
    "RuleNumber": 1,  
    "RuleAction": "accept",  
    "Protocol": 6  
  },  
  "ClientToken": "4752b573-40a6-4eac-a8a4-a72058761219"  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーフィルターの作成AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTrafficMirrorFilterRule](#)」の「」を参照してください。

create-traffic-mirror-filter

次のコード例は、create-traffic-mirror-filter を使用する方法を示しています。

AWS CLI

Traffic Mirror Filter を作成するには

次のcreate-traffic-mirror-filter例では、Traffic Mirror フィルターを作成します。フィルターを作成したら、create-traffic-mirror-filter-ruleを使用してフィルターにルールを追加します。

```
aws ec2 create-traffic-mirror-filter \  
  --description "TCP Filter"
```

出力:

```
{  
  "ClientToken": "28908518-100b-4987-8233-8c744EXAMPLE",  
  "TrafficMirrorFilter": {  
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",  
    "Description": "TCP Filter",  
    "EgressFilterRules": [],  
    "IngressFilterRules": [],  
    "Tags": [],  
    "NetworkServices": []  
  }  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーフィルターの作成AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTrafficMirrorFilter](#)」の「」を参照してください。

create-traffic-mirror-session

次のコード例は、create-traffic-mirror-session を使用する方法を示しています。

AWS CLI

トラフィックミラーセッションを作成するには

次のcreate-traffic-mirror-sessionコマンドは、指定されたソースとターゲットのトラフィックミラーセッションを 25 バイトのパケットに対して作成します。

```
aws ec2 create-traffic-mirror-session \  
  --description "example session" \  
  --traffic-mirror-target-id tmt-07f75d8feeEXAMPLE \  
  --network-interface-id eni-070203f901EXAMPLE \  
  --session-number 1 \  
  --packet-length 25 \  
  --traffic-mirror-filter-id tmf-04812ff784EXAMPLE
```

出力:

```
{  
  "TrafficMirrorSession": {  
    "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",  
    "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",  
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",  
    "NetworkInterfaceId": "eni-070203f901EXAMPLE",  
    "OwnerId": "111122223333",  
    "PacketLength": 25,  
    "SessionNumber": 1,  
    "VirtualNetworkId": 7159709,  
    "Description": "example session",  
    "Tags": []  
  },  
  "ClientToken": "5236cffc-ee13-4a32-bb5b-388d9da09d96"  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーセッションの作成AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTrafficMirrorSession](#)」の「」を参照してください。

create-traffic-mirror-target

次のコード例は、create-traffic-mirror-target を使用する方法を示しています。

AWS CLI

Network Load Balancer トラフィックミラーターゲットを作成するには

次のcreate-traffic-mirror-target例では、Network Load Balancer トラフィックミラーターゲットを作成します。

```
aws ec2 create-traffic-mirror-target \  
  --description "Example Network Load Balancer Target" \  
  --network-load-balancer-arn arn:aws:elasticloadbalancing:us-  
east-1:111122223333:loadbalancer/net/NLB/7cdec873EXAMPLE
```

出力:

```
{  
  "TrafficMirrorTarget": {  
    "Type": "network-load-balancer",  
    "Tags": [],  
    "Description": "Example Network Load Balancer Target",  
    "OwnerId": "111122223333",  
    "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-  
east-1:724145273726:loadbalancer/net/NLB/7cdec873EXAMPLE",  
    "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE"  
  },  
  "ClientToken": "d5c090f5-8a0f-49c7-8281-72c796a21f72"  
}
```

ネットワーク Traffic Mirror ターゲットを作成するには

次のcreate-traffic-mirror-target例では、ネットワークインターフェイスの Traffic Mirror ターゲットを作成します。

```
aws ec2 create-traffic-mirror-target --description 「ネットワークインターフェイスターゲット」 --  
network-interface-id eni-eni-01f6f631eEXAMPLE
```

出力:

```
{  
  "ClientToken": "5289a345-0358-4e62-93d5-47ef3061d65e",  
  "TrafficMirrorTarget": {  
    "Description": "Network interface target",  
    "NetworkInterfaceId": "eni-01f6f631eEXAMPLE",  
    "TrafficMirrorTargetId": "tmt-02dcdb2abEXAMPLE",  
    "OwnerId": "111122223333",  
    "Type": "network-interface",  
    "Tags": []  
  }  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーターゲットの作成AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTrafficMirrorTarget](#)」の「」を参照してください。

create-transit-gateway-connect-peer

次の例は、create-transit-gateway-connect-peer を使用する方法を説明しています。

AWS CLI

Transit Gateway Connect ピアを作成するには

次のcreate-transit-gateway-connect-peer例では、Connect ピアを作成します。

```
aws ec2 create-transit-gateway-connect-peer \  
  --transit-gateway-attachment-id tgw-attach-0f0927767cEXAMPLE \  
  --peer-address 172.31.1.11 \  
  --inside-cidr-blocks 169.254.6.0/29
```

出力:

```
{  
  "TransitGatewayConnectPeer": {  
    "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",  
    "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",  
    "State": "pending",  
    "CreationTime": "2021-10-13T03:35:17.000Z",  
    "ConnectPeerConfiguration": {  
      "TransitGatewayAddress": "10.0.0.234",  
      "PeerAddress": "172.31.1.11",  
      "InsideCidrBlocks": [  
        "169.254.6.0/29"  
      ],  
      "Protocol": "gre",  
      "BgpConfigurations": [  
        {  
          "TransitGatewayAsn": 64512,  
          "PeerAsn": 64512,  
          "TransitGatewayAddress": "169.254.6.2",  
          "PeerAddress": "169.254.6.1",  
          "BgpStatus": "down"  
        }  
      ]  
    }  
  }  
}
```

```

    },
    {
      "TransitGatewayAsn": 64512,
      "PeerAsn": 64512,
      "TransitGatewayAddress": "169.254.6.3",
      "PeerAddress": "169.254.6.1",
      "BgpStatus": "down"
    }
  ]
}
}
}
}
}

```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway Connect アタッチメント」と「Transit Gateway Connect ピア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitGatewayConnectPeer](#)」の「」を参照してください。

create-transit-gateway-connect

次のコード例は、create-transit-gateway-connect を使用する方法を示しています。

AWS CLI

Transit Gateway Connect アタッチメントを作成するには

次のcreate-transit-gateway-connect例では、指定されたアタッチメントの「gre」プロトコルを使用して Connect アタッチメントを作成します。

```

aws ec2 create-transit-gateway-connect \
  --transport-transit-gateway-attachment-id tgw-attach-0a89069f57EXAMPLE \
  --options "Protocol=gre"

```

出力:

```

{
  "TransitGatewayConnect": {
    "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
    "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "State": "pending",
  }
}

```

```
    "CreationTime": "2021-03-09T19:59:17+00:00",
    "Options": {
      "Protocol": "gre"
    }
  }
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway Connect アタッチメント」と「Transit Gateway Connect ピア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitGatewayConnect](#)」の「」を参照してください。

create-transit-gateway-multicast-domain

次の例は、create-transit-gateway-multicast-domain を使用する方法を説明しています。

AWS CLI

例 1: IGMP マルチキャストドメインを作成するには

次のcreate-transit-gateway-multicast-domain例では、指定されたトランジットゲートウェイのマルチキャストドメインを作成します。静的ソースが無効になっている場合、マルチキャストドメインに関連付けられたサブネット内のインスタンスは、マルチキャストトラフィックを送信できます。少なくとも 1 人のメンバーがIGMPプロトコルを使用する場合は、IGMPv2サポートを有効にする必要があります。

```
aws ec2 create-transit-gateway-multicast-domain \
  --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
  --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

出力:

```
{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c9e29e2a7EXAMPLE",
    "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
    "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-domain-0c9e29e2a7EXAMPLE",
    "OwnerId": "123456789012",
```

```
    "Options": {
      "Icmpv2Support": "enable",
      "StaticSourcesSupport": "disable",
      "AutoAcceptSharedAssociations": "disable"
    },
    "State": "pending",
    "CreationTime": "2021-09-29T22:17:13.000Z"
  }
}
```

例 2: 静的マルチキャストドメインを作成するには

次の `create-transit-gateway-multicast-domain` 例では、指定されたトランジットゲートウェイのマルチキャストドメインを作成します。静的ソースが有効になっている場合は、ソースを静的に追加する必要があります。

```
aws ec2 create-transit-gateway-multicast-domain \
  --transit-gateway-id tgw-0bf0bfffefEXAMPLE \
  --options StaticSourcesSupport=enable,Icmpv2Support=disable
```

出力:

```
{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
    "TransitGatewayId": "tgw-0bf0bfffefEXAMPLE",
    "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-domain-000fb24d04EXAMPLE",
    "OwnerId": "123456789012",
    "Options": {
      "Icmpv2Support": "disable",
      "StaticSourcesSupport": "enable",
      "AutoAcceptSharedAssociations": "disable"
    },
    "State": "pending",
    "CreationTime": "2021-09-29T22:20:19.000Z"
  }
}
```

詳細については、「Transit Gateways ガイド」の [「マルチキャストドメインの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateTransitGatewayMulticastDomain](#)」の「」を参照してください。

create-transit-gateway-peering-attachment

次の例は、create-transit-gateway-peering-attachment を使用する方法を説明しています。

AWS CLI

Transit Gateway ピアリングアタッチメントを作成するには

次のcreate-transit-gateway-peering-attachment例では、指定された2つのトランジットゲートウェイ間にピアリングアタッチメントリクエストを作成します。

```
aws ec2 create-transit-gateway-peering-attachment \  
  --transit-gateway-id tgw-123abc05e04123abc \  
  --peer-transit-gateway-id tgw-11223344aabbcc112 \  
  --peer-account-id 123456789012 \  
  --peer-region us-east-2
```

出力:

```
{  
  "TransitGatewayPeeringAttachment": {  
    "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbcccd",  
    "RequesterTgwInfo": {  
      "TransitGatewayId": "tgw-123abc05e04123abc",  
      "OwnerId": "123456789012",  
      "Region": "us-west-2"  
    },  
    "AcceptorTgwInfo": {  
      "TransitGatewayId": "tgw-11223344aabbcc112",  
      "OwnerId": "123456789012",  
      "Region": "us-east-2"  
    },  
    "State": "initiatingRequest",  
    "CreationTime": "2019-12-09T11:38:05.000Z"  
  }  
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ピアリングアタッチメント」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitGatewayPeeringAttachment](#)」の「」を参照してください。

create-transit-gateway-policy-table

次の例は、create-transit-gateway-policy-table を使用する方法を説明しています。

AWS CLI

トランジットゲートウェイポリシーテーブルを作成するには

次のcreate-transit-gateway-policy-table例では、指定されたトランジットゲートウェイのトランジットゲートウェイポリシーテーブルを作成します。

```
aws ec2 create-transit-gateway-policy-table \  
--transit-gateway-id tgw-067f8505c18f0bd6e
```

出力:

```
{  
  "TransitGatewayPolicyTable": {  
    "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",  
    "TransitGatewayId": "tgw-067f8505c18f0bd6e",  
    "State": "pending",  
    "CreationTime": "2023-11-28T16:36:43+00:00"  
  }  
}
```

詳細については、[「Transit Gateway ユーザーガイド」の「Transit Gateway ポリシーテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitGatewayPolicyTable](#)」の「」を参照してください。

create-transit-gateway-prefix-list-reference

次の例は、create-transit-gateway-prefix-list-reference を使用する方法を説明しています。

AWS CLI

プレフィックスリストへの参照を作成するには

次のcreate-transit-gateway-prefix-list-reference例では、指定された Transit Gateway ルートテーブルに指定されたプレフィックスリストへの参照を作成します。

```
aws ec2 create-transit-gateway-prefix-list-reference \  
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \  
  --prefix-list-id pl-1111112222222333 \  
  --transit-gateway-attachment-id tgw-attach-aaaaabbbbb11111
```

出力:

```
{  
  "TransitGatewayPrefixListReference": {  
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",  
    "PrefixListId": "pl-1111112222222333",  
    "PrefixListOwnerId": "123456789012",  
    "State": "pending",  
    "Blackhole": false,  
    "TransitGatewayAttachment": {  
      "TransitGatewayAttachmentId": "tgw-attach-aaaaabbbbb11111",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-112233445566aabbcc"  
    }  
  }  
}
```

詳細については、「Transit Gateways ガイド」の「[プレフィックスリストリファレンス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateTransitGatewayPrefixListReference](#)」の「」を参照してください。

create-transit-gateway-route-table

次のコード例は、create-transit-gateway-route-table を使用する方法を示しています。

AWS CLI

Transit Gateway ルートテーブルを作成するには

次のcreate-transit-gateway-route-table例では、指定されたトランジットゲートウェイのルートテーブルを作成します。

```
aws ec2 create-transit-gateway-route-table \  
  --transit-gateway-id tgw-0262a0e521EXAMPLE
```

出力:

```
{  
  "TransitGatewayRouteTable": {  
    "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",  
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",  
    "State": "pending",  
    "DefaultAssociationRouteTable": false,  
    "DefaultPropagationRouteTable": false,  
    "CreationTime": "2019-07-10T19:01:46.000Z"  
  }  
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブルを作成する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitGatewayRouteTable](#)」の「」を参照してください。

create-transit-gateway-route

次のコード例は、create-transit-gateway-route を使用する方法を示しています。

AWS CLI

トランジットゲートウェイルートを作成するには

次のcreate-transit-gateway-route例では、指定されたルートテーブルに対して、指定された宛先を持つルートを作成します。

```
aws ec2 create-transit-gateway-route \  
  --destination-cidr-block 10.0.2.0/24 \  
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

出力:

```
{
  "Route": {
    "DestinationCidrBlock": "10.0.2.0/24",
    "TransitGatewayAttachments": [
      {
        "ResourceId": "vpc-0065acced4EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
        "ResourceType": "vpc"
      }
    ],
    "Type": "static",
    "State": "active"
  }
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitGatewayRoute](#)」の「」を参照してください。

create-transit-gateway-vpc-attachment

次のコード例は、create-transit-gateway-vpc-attachment を使用する方法を示しています。

AWS CLI

例 1: トランジットゲートウェイを に関連付けるには VPC

次のcreate-transit-gateway-vpc-attachment例では、指定された へのトランジットゲートウェイアタッチメントを作成しますVPC。

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-0262a0e521EXAMPLE \
  --vpc-id vpc-07e8ffd50f49335df \
  --subnet-id subnet-0752213d59EXAMPLE
```

出力:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "VpcId": "vpc-07e8ffd50fEXAMPLE",
    "VpcOwnerId": "111122223333",
    "State": "pending",
    "SubnetIds": [
      "subnet-0752213d59EXAMPLE"
    ],
    "CreationTime": "2019-07-10T17:33:46.000Z",
    "Options": {
      "DnsSupport": "enable",
      "Ipv6Support": "disable"
    }
  }
}
```

詳細については、[「Transit Gateways ガイド」の「へのトランジットゲートウェイアタッチメントの作成VPC」](#)を参照してください。

例 2: トランジットゲートウェイを 内の複数のサブネットに関連付けるには VPC

次のcreate-transit-gateway-vpc-attachment例では、指定された VPC および サブネットへの Transit Gateway アタッチメントを作成します。

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-02f776b1a7EXAMPLE \
  --vpc-id vpc-3EXAMPLE \
  --subnet-ids "subnet-dEXAMPLE" "subnet-6EXAMPLE"
```

出力:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0e141e0bebEXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "VpcId": "vpc-3EXAMPLE",
    "VpcOwnerId": "111122223333",
    "State": "pending",
    "SubnetIds": [
      "subnet-6EXAMPLE",
    ]
  }
}
```

```

        "subnet-dEXAMPLE"
    ],
    "CreationTime": "2019-12-17T20:07:52.000Z",
    "Options": {
        "DnsSupport": "enable",
        "Ipv6Support": "disable"
    }
}
}
}

```

詳細については、[「トランジットゲートウェイガイド」の「へのトランジットゲートウェイアタッチメントの作成VPC」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateTransitGatewayVpcAttachment](#)」の「」を参照してください。

create-transit-gateway

次のコード例は、create-transit-gateway を使用する方法を示しています。

AWS CLI

トランジットゲートウェイを作成するには

次のcreate-transit-gateway例では、トランジットゲートウェイを作成します。

```

aws ec2 create-transit-gateway \
  --description MyTGW \
  --
options AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=

```

出力:

```

{
  "TransitGateway": {
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-gateway/tgw-0262a0e521EXAMPLE",
    "State": "pending",
    "OwnerId": "111122223333",
    "Description": "MyTGW",
    "CreationTime": "2019-07-10T14:02:12.000Z",

```

```

    "Options": {
      "AmazonSideAsn": 64516,
      "AutoAcceptSharedAttachments": "enable",
      "DefaultRouteTableAssociation": "enable",
      "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
      "DefaultRouteTablePropagation": "enable",
      "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
      "VpnEcmpSupport": "enable",
      "DnsSupport": "enable"
    }
  }
}

```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway を作成する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTransitGateway](#)」の「」を参照してください。

create-verified-access-endpoint

次のコード例は、create-verified-access-endpoint を使用する方法を示しています。

AWS CLI

Verified Access エンドポイントを作成するには

次のcreate-verified-access-endpoint例では、指定された Verified Access グループの Verified Access エンドポイントを作成します。指定されたネットワークインターフェイスとセキュリティグループは、同じに属している必要がありますVPC。

```

aws ec2 create-verified-access-endpoint \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --endpoint-type network-interface \
  --attachment-type vpc \
  --domain-certificate-arn arn:aws:acm:us-east-2:123456789012:certificate/  

eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE \
  --application-domain example.com \
  --endpoint-domain-prefix my-ava-app \
  --security-group-ids sg-004915970c4c8f13a \
  --network-interface-  

options NetworkInterfaceId=eni-0aec70418c8d87a0f,Protocol=https,Port=443 \

```



```
--tag-specifications ResourceType=verified-access-  
endpoint,Tags=[{Key=Name,Value=my-va-endpoint}]
```

出力:

```
{  
  "VerifiedAccessEndpoint": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",  
    "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",  
    "ApplicationDomain": "example.com",  
    "EndpointType": "network-interface",  
    "AttachmentType": "vpc",  
    "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/  
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",  
    "EndpointDomain": "my-ava-  
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-  
east-2.amazonaws.com",  
    "SecurityGroupIds": [  
      "sg-004915970c4c8f13a"  
    ],  
    "NetworkInterfaceOptions": {  
      "NetworkInterfaceId": "eni-0aec70418c8d87a0f",  
      "Protocol": "https",  
      "Port": 443  
    },  
    "Status": {  
      "Code": "pending"  
    },  
    "Description": "",  
    "CreationTime": "2023-08-25T20:54:43",  
    "LastUpdatedTime": "2023-08-25T20:54:43",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-va-endpoint"  
      }  
    ]  
  }  
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access エンドポイント」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVerifiedAccessEndpoint](#)」の「」を参照してください。

create-verified-access-group

次の例は、create-verified-access-group を使用する方法を説明しています。

AWS CLI

Verified Access グループを作成するには

次のcreate-verified-access-group例では、指定された Verified Access インスタンス用の Verified Access グループを作成します。

```
aws ec2 create-verified-access-group \  
  --verified-access-instance-id vai-0ce000c0b7643abea \  
  --tag-specifications ResourceType=verified-access-  
group, Tags=[{Key=Name, Value=my-va-group}]
```

出力:

```
{  
  "VerifiedAccessGroup": {  
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "",  
    "Owner": "123456789012",  
    "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-  
access-group/vagr-0dbe967baf14b7235",  
    "CreationTime": "2023-08-25T19:55:19",  
    "LastUpdatedTime": "2023-08-25T19:55:19",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-va-group"  
      }  
    ]  
  }  
}
```

詳細については、[「Verified Access ユーザーガイド」](#)の「[Verified Access グループ](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVerifiedAccessGroup](#)」の「」を参照してください。

create-verified-access-instance

次のコード例は、create-verified-access-instance を使用する方法を示しています。

AWS CLI

Verified Access インスタンスを作成するには

次のcreate-verified-access-instance例では、名前タグを使用して Verified Access インスタンスを作成します。

```
aws ec2 create-verified-access-instance \  
  --tag-specifications ResourceType=verified-access-  
instance, Tags=[{Key=Name, Value=my-va-instance}]
```

出力:

```
{  
  "VerifiedAccessInstance": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "",  
    "VerifiedAccessTrustProviders": [],  
    "CreationTime": "2023-08-25T18:27:56",  
    "LastUpdatedTime": "2023-08-25T18:27:56",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-va-instance"  
      }  
    ]  
  }  
}
```

詳細については、「[Verified Access ユーザーガイド](#)」の「[Verified Access インスタンス](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVerifiedAccessInstance](#)」の「」を参照してください。

create-verified-access-trust-provider

次のコード例は、`create-verified-access-trust-provider` を使用する方法を示しています。

AWS CLI

Verified Access 信頼プロバイダーを作成するには

次の`create-verified-access-trust-provider`例では、AWS Identity Center を使用して Verified Access 信頼プロバイダーを設定します。

```
aws ec2 create-verified-access-trust-provider \  
  --trust-provider-type user \  
  --user-trust-provider-type iam-identity-center \  
  --policy-reference-name idc \  
  --tag-specifications ResourceType=verified-access-trust-  
provider,Tags=[{Key=Name,Value=my-va-trust-provider}]
```

出力:

```
{  
  "VerifiedAccessTrustProvider": {  
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
    "Description": "",  
    "TrustProviderType": "user",  
    "UserTrustProviderType": "iam-identity-center",  
    "PolicyReferenceName": "idc",  
    "CreationTime": "2023-08-25T18:40:36",  
    "LastUpdatedTime": "2023-08-25T18:40:36",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "my-va-trust-provider"  
      }  
    ]  
  }  
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access の信頼プロバイダー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateVerifiedAccessTrustProvider](#)」の「」を参照してください。

create-volume

次のコード例は、create-volume を使用する方法を示しています。

AWS CLI

空の汎用 SSD (gp2) ボリュームを作成するには

次のcreate-volume例では、指定されたアベイラビリティゾーンに 80 GiB 汎用 SSD (gp2) ボリュームを作成します。現在のリージョンは である必要があります。またはus-east-1、 --regionパラメータを追加してコマンドのリージョンを指定することもできます。

```
aws ec2 create-volume \  
  --volume-type gp2 \  
  --size 80 \  
  --availability-zone us-east-1a
```

出力:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": false,  
  "VolumeType": "gp2",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
  "Iops": 240,  
  "SnapshotId": "",  
  "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
  "Size": 80  
}
```

ボリュームタイプを指定しない場合、デフォルトのボリュームタイプは `gp2` です。

```
aws ec2 create-volume \  
  --size 80 \  
  --availability-zone us-east-1a
```

例 2: スナップショットからプロビジョンド IOPS SSD (io1) ボリュームを作成するには

次のcreate-volume例では、指定されたスナップショットを使用して、指定されたアベイラビリティゾーンIOPSにプロビジョニングされた 1000 のプロビジョニング済み IOPS SSD (io1) ボリュームを作成します。

```
aws ec2 create-volume \  
  --volume-type io1 \  
  --iops 1000 \  
  --snapshot-id snap-066877671789bd71b \  
  --availability-zone us-east-1a
```

出力:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": false,  
  "VolumeType": "io1",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
  "Iops": 1000,  
  "SnapshotId": "snap-066877671789bd71b",  
  "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
  "Size": 500  
}
```

例 3: 暗号化されたボリュームを作成するには

次のcreate-volume例では、暗号化CMKのデフォルトを使用してEBS暗号化されたボリュームを作成します。デフォルトでは暗号化が無効になっている場合は、次のように --encryptedパラメータを指定する必要があります。

```
aws ec2 create-volume \  
  --size 80 \  
  --encrypted \  
  --availability-zone us-east-1a
```

出力:

```
{
```

```
"AvailabilityZone": "us-east-1a",
"Tags": [],
"Encrypted": true,
"VolumeType": "gp2",
"VolumeId": "vol-1234567890abcdef0",
"State": "creating",
"Iops": 240,
"SnapshotId": "",
"CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",
"Size": 80
}
```

暗号化がデフォルトで有効になっている場合、次のコマンド例では、`--encrypted`パラメータがなくても暗号化されたボリュームを作成します。

```
aws ec2 create-volume \
  --size 80 \
  --availability-zone us-east-1a
```

`--kms-key-id` パラメータを使用してカスタマー管理の を指定する場合はCMK、デフォルトで暗号化が有効になっている場合でも、`--encrypted`パラメータを指定する必要があります。

```
aws ec2 create-volume \
  --volume-type gp2 \
  --size 80 \
  --encrypted \
  --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \
  --availability-zone us-east-1a
```

例 4: タグを使用してボリュームを作成するには

次のcreate-volume例では、ボリュームを作成し、2つのタグを追加します。

```
aws ec2 create-volume \
  --availability-zone us-east-1a \
  --volume-type gp2 \
  --size 80 \
  --tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},
{Key=cost-center,Value=cc123}]'
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateVolume](#)」の「」を参照してください。

create-vpc-endpoint-connection-notification

次のコード例は、`create-vpc-endpoint-connection-notification` を使用する方法を示しています。

AWS CLI

エンドポイント接続通知を作成するには

この例では、インターフェイスエンドポイントがサービスに接続され、エンドポイントがサービスに受け入れられたときに警告する、特定のエンドポイントサービスの通知を作成します。

コマンド:

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-arn arn:aws:sns:us-east-2:123456789012:VpceNotification --connection-events Connect Accept --service-id vpce-svc-1237881c0d25a3abc
```

出力:

```
{
  "ConnectionNotification": {
    "ConnectionNotificationState": "Enabled",
    "ConnectionNotificationType": "Topic",
    "ServiceId": "vpce-svc-1237881c0d25a3abc",
    "ConnectionEvents": [
      "Accept",
      "Connect"
    ],
    "ConnectionNotificationId": "vpce-nfn-008776de7e03f5abc",
    "ConnectionNotificationArn": "arn:aws:sns:us-east-2:123456789012:VpceNotification"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateVpcEndpointConnectionNotification](#)」の「」を参照してください。

create-vpc-endpoint-service-configuration

次の例は、`create-vpc-endpoint-service-configuration` を使用する方法を説明しています。

AWS CLI

例 1: インターフェイスエンドポイントのエンドポイントサービス設定を作成するには

次の`create-vpc-endpoint-service-configuration`例では、Network Load Balancer を使用してVPCエンドポイントサービス設定を作成します`nlb-vpce`。この例では、インターフェイスエンドポイントを介してサービスに接続するリクエストを受け入れる必要があることも指定します。

```
aws ec2 create-vpc-endpoint-service-configuration \  
  --network-load-balancer-arns arn:aws:elasticloadbalancing:us-  
east-1:123456789012:loadbalancer/net/nlb-vpce/e94221227f1ba532 \  
  --acceptance-required
```

出力:

```
{  
  "ServiceConfiguration": {  
    "ServiceType": [  
      {  
        "ServiceType": "Interface"  
      }  
    ],  
    "NetworkLoadBalancerArns": [  
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/  
nlb-vpce/e94221227f1ba532"  
    ],  
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-03d5ebb7d9579a2b3",  
    "ServiceState": "Available",  
    "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",  
    "AcceptanceRequired": true,  
    "AvailabilityZones": [  
      "us-east-1d"  
    ],  
    "BaseEndpointDnsNames": [  
      "vpce-svc-03d5ebb7d9579a2b3.us-east-1.vpce.amazonaws.com"  
    ]  
  }  
}
```

例 2: Gateway Load Balancer エンドポイントのエンドポイントサービス設定を作成するには

次のcreate-vpc-endpoint-service-configuration例では、Gateway Load Balancer を使用してVPCエンドポイントサービス設定を作成しますGWLBService。Gateway Load Balancer エンドポイントを介してサービスに接続するリクエストは自動的に受け入れられます。

```
aws ec2 create-vpc-endpoint-service-configuration \  
  --gateway-load-balancer-arns arn:aws:elasticloadbalancing:us-  
east-1:123456789012:loadbalancer/gwy/GWLBService/123123123123abcc \  
  --no-acceptance-required
```

出力:

```
{  
  "ServiceConfiguration": {  
    "ServiceType": [  
      {  
        "ServiceType": "GatewayLoadBalancer"  
      }  
    ],  
    "ServiceId": "vpce-svc-123123a1c43abc123",  
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",  
    "ServiceState": "Available",  
    "AvailabilityZones": [  
      "us-east-1d"  
    ],  
    "AcceptanceRequired": false,  
    "ManagesVpcEndpoints": false,  
    "GatewayLoadBalancerArns": [  
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/gwy/  
GWLBService/123123123123abcc"  
    ]  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の[VPC「エンドポイントサービス」](#)を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpcEndpointServiceConfiguration](#)」の「」を参照してください。

create-vpc-endpoint

次のコード例は、create-vpc-endpoint を使用する方法を示しています。

AWS CLI

例 1: ゲートウェイエンドポイントを作成するには

次のcreate-vpc-endpoint例では、us-east-1 リージョンで VPC vpc-1a2b3c4d と Amazon S3 の間にゲートウェイVPCエンドポイントを作成し、ルートテーブルをエンドポイント rtb-11aa22bbに関連付けます。

```
aws ec2 create-vpc-endpoint \  
  --vpc-id vpc-1a2b3c4d \  
  --service-name com.amazonaws.us-east-1.s3 \  
  --route-table-ids rtb-11aa22bb
```

出力:

```
{  
  "VpcEndpoint": {  
    "PolicyDocument": "{\n\"Version\": \"2008-10-17\",  
\"Statement\": [{\n\"Sid\": \"\",  
\"Effect\": \"Allow\",  
\"Principal\": \"*\",  
\"Action\": \"*\",  
\"Resource\": \"*\"}]}",  
    "VpcId": "vpc-1a2b3c4d",  
    "State": "available",  
    "ServiceName": "com.amazonaws.us-east-1.s3",  
    "RouteTableIds": [  
      "rtb-11aa22bb"  
    ],  
    "VpcEndpointId": "vpc-1a2b3c4d",  
    "CreationTimestamp": "2015-05-15T09:40:50Z"  
  }  
}
```

詳細については、「ガイド」の [「ゲートウェイエンドポイントの作成」](#) を参照してください。

AWS PrivateLink

例 2: インターフェイスエンドポイントを作成するには

次のcreate-vpc-endpoint例では、us-east-1 リージョンで VPC vpc-1a2b3c4d と Amazon S3 の間にインターフェイスVPCエンドポイントを作成します。コマンドは、サブネットにエンドポイントを作成し subnet-1a2b3c4d、セキュリティグループに関連付け sg-1a2b3c4d、「Service」のキーと「S3」の値を持つタグを追加します。

```
aws ec2 create-vpc-endpoint \  
  --vpc-id vpc-1a2b3c4d \  
  --interface-vpc-id vpc-1a2b3c4d \  
  --security-groups sg-1a2b3c4d \  
  --tags Key=Service,Value=S3
```

```
--vpc-id vpc-1a2b3c4d \  
--vpc-endpoint-type Interface \  
--service-name com.amazonaws.us-east-1.s3 \  
--subnet-ids subnet-7b16de0c \  
--security-group-id sg-1a2b3c4d \  
--tag-specifications ResourceType=vpc-endpoint,Tags=[{Key=service,Value=S3}]
```

出力:

```
{  
  "VpcEndpoint": {  
    "VpcEndpointId": "vpce-1a2b3c4d5e6f1a2b3",  
    "VpcEndpointType": "Interface",  
    "VpcId": "vpc-1a2b3c4d",  
    "ServiceName": "com.amazonaws.us-east-1.s3",  
    "State": "pending",  
    "RouteTableIds": [],  
    "SubnetIds": [  
      "subnet-1a2b3c4d"  
    ],  
    "Groups": [  
      {  
        "GroupId": "sg-1a2b3c4d",  
        "GroupName": "default"  
      }  
    ],  
    "PrivateDnsEnabled": false,  
    "RequesterManaged": false,  
    "NetworkInterfaceIds": [  
      "eni-0b16f0581c8ac6877"  
    ],  
    "DnsEntries": [  
      {  
        "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg.s3.us-east-1.vpce.amazonaws.com",  
        "HostedZoneId": "Z7HUB22UULQXV"  
      },  
      {  
        "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg-us-east-1c.s3.us-east-1.vpce.amazonaws.com",  
        "HostedZoneId": "Z7HUB22UULQXV"  
      }  
    ],  
  },  
}
```

```
    "CreationTimestamp": "2021-03-05T14:46:16.030000+00:00",
    "Tags": [
      {
        "Key": "service",
        "Value": "S3"
      }
    ],
    "OwnerId": "123456789012"
  }
}
```

詳細については、「ユーザーガイド」の[「インターフェイスエンドポイントの作成」](#)を参照してください。 AWS PrivateLink

例 3: Gateway Load Balancer エンドポイントを作成するには

次のcreate-vpc-endpoint例では、VPC vpc-111122223333aabbcc と の間に Gateway Load Balancer エンドポイントを作成し、Gateway Load Balancer を使用して設定されたサービスを作成します。

```
aws ec2 create-vpc-endpoint \
  --service-name com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123 \
  --vpc-endpoint-type GatewayLoadBalancer \
  --vpc-id vpc-111122223333aabbcc \
  --subnet-ids subnet-0011aabbcc2233445
```

出力:

```
{
  "VpcEndpoint": {
    "VpcEndpointId": "vpce-aabbaabbaabbaabba",
    "VpcEndpointType": "GatewayLoadBalancer",
    "VpcId": "vpc-111122223333aabbcc",
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",
    "State": "pending",
    "SubnetIds": [
      "subnet-0011aabbcc2233445"
    ],
    "RequesterManaged": false,
    "NetworkInterfaceIds": [
      "eni-01010120203030405"
    ],
  },
}
```

```
    "CreationTimestamp": "2020-11-11T08:06:03.522Z",
    "OwnerId": "123456789012"
  }
}
```

詳細については、「ユーザーガイド」の「[Gateway Load Balancer エンドポイント](#)」を参照してください。 AWS PrivateLink

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpcEndpoint](#)」の「」を参照してください。

create-vpc-peering-connection

次のコード例は、create-vpc-peering-connection を使用する方法を示しています。

AWS CLI

間でVPCピアリング接続を作成するには VPCs

この例では、vpc-1a2b3c4d と VPCs vpc-11122233 間のピアリング接続をリクエストします。

コマンド:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-  
id vpc-11122233
```

出力:

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Initiating Request to 444455556666",
      "Code": "initiating-request"
    },
    "Tags": [],
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-1a2b3c4d",
      "CidrBlock": "10.0.0.0/28"
    },
    "VpcPeeringConnectionId": "pcx-111aaa111",
```

```
    "ExpirationTime": "2014-04-02T16:13:36.000Z",
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-11122233"
    }
  }
}
```

別のアカウントVPCでとのVPCピアリング接続を作成するには

この例では、VPC (vpc-1a2b3c4d) と AWS アカウント 123456789012 に属する VPC (vpc-11122233) 間のピアリング接続をリクエストします。

コマンド:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-owner-id 123456789012
```

別のリージョンVPCでとのVPCピアリング接続を作成するには

この例では、現在のリージョン (vpc-1a2b3c4d) VPCの とus-west-2、リージョンのアカウント VPC (vpc-11122233) 間のピアリング接続をリクエストします。

コマンド:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-region us-west-2
```

この例では、現在のリージョン (vpc-1a2b3c4d) VPCの とus-west-2、リージョンにある AWS アカウント 123456789012 に属する VPC (vpc-11122233) 間のピアリング接続をリクエストします。

コマンド:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-owner-id 123456789012 --peer-region us-west-2
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpcPeeringConnection](#)」の「」を参照してください。

create-vpc

次の例は、create-vpc を使用する方法を説明しています。

AWS CLI

例 1: を作成するには VPC

次のcreate-vpc例では、指定されたIPv4CIDRブロックと名前タグVPCを使用して を作成します。

```
aws ec2 create-vpc \  
  --cidr-block 10.0.0.0/16 \  
  --tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=MyVpc}]
```

出力:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.0.0/16",  
    "DhcpOptionsId": "dopt-5EXAMPLE",  
    "State": "pending",  
    "VpcId": "vpc-0a60eb65b4EXAMPLE",  
    "OwnerId": "123456789012",  
    "InstanceTenancy": "default",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-07501b79ecEXAMPLE",  
        "CidrBlock": "10.0.0.0/16",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false,  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "MyVpc"  
      }  
    ]  
  }  
}
```



```
}
```

例 2: 専用テナンシーVPCで を作成するには

次のcreate-vpc例では、指定されたIPv4CIDRブロックと専用テナンシーVPCを使用して を作成します。

```
aws ec2 create-vpc \  
  --cidr-block 10.0.0.0/16 \  
  --instance-tenancy dedicated
```

出力:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.0.0/16",  
    "DhcpOptionsId": "dopt-19edf471",  
    "State": "pending",  
    "VpcId": "vpc-0a53287fa4EXAMPLE",  
    "OwnerId": "111122223333",  
    "InstanceTenancy": "dedicated",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-00b24cc1c2EXAMPLE",  
        "CidrBlock": "10.0.0.0/16",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false  
  }  
}
```

例 3: IPv6CIDRブロックVPCを使用して を作成するには

次のcreate-vpc例では、Amazon が提供するIPv6CIDRブロックVPCを使用して を作成します。

```
aws ec2 create-vpc \  
  --cidr-block 2001:db8:1:1::/64
```

```
--cidr-block 10.0.0.0/16 \  
--amazon-provided-ipv6-cidr-block
```

出力:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.0.0/16",  
    "DhcpOptionsId": "dopt-dEXAMPLE",  
    "State": "pending",  
    "VpcId": "vpc-0fc5e3406bEXAMPLE",  
    "OwnerId": "123456789012",  
    "InstanceTenancy": "default",  
    "Ipv6CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-068432c60bEXAMPLE",  
        "Ipv6CidrBlock": "",  
        "Ipv6CidrBlockState": {  
          "State": "associating"  
        },  
        "Ipv6Pool": "Amazon",  
        "NetworkBorderGroup": "us-west-2"  
      }  
    ],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",  
        "CidrBlock": "10.0.0.0/16",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false  
  }  
}
```

例 4: IPAM プールCIDRから VPCを使用して を作成するには

次のcreate-vpc例では、Amazon VPC IP Address Manager (IPAM) プールCIDRから VPCを使用して を作成します。

Linux および macOS:

```
aws ec2 create-vpc \  
  --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 \  
  --tag-specifications ResourceType=vpc,Tags='[{Key=Environment,Value="Preprod"}],  
{Key=Owner,Value="Build Team"}]'
```

Windows :

```
aws ec2 create-vpc ^  
  --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --tag-specifications ResourceType=vpc,Tags=[{Key=Environment,Value="Preprod"},  
{Key=Owner,Value="Build Team"}]
```

出力:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.1.0/24",  
    "DhcpOptionsId": "dopt-2afccf50",  
    "State": "pending",  
    "VpcId": "vpc-010e1791024eb0af9",  
    "OwnerId": "123456789012",  
    "InstanceTenancy": "default",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-0a77de1d803226d4b",  
        "CidrBlock": "10.0.1.0/24",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false,  
    "Tags": [  
      {  
        "Key": "Environment",  
        "Value": "Preprod"  
      },  
      {  
        "Key": "Owner",  
        "Value": "Build Team"  
      }  
    ]  
  }  
}
```

```
    ]  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の[VPC IPAM「プールを使用するを作成する CIDR」](#)を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpc](#)」の「」を参照してください。

create-vpn-connection-route

次の例は、create-vpn-connection-route を使用する方法を説明しています。

AWS CLI

VPN接続の静的ルートを作成するには

この例では、指定されたVPN接続の静的ルートを作成します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpnConnectionRoute](#)」の「」を参照してください。

create-vpn-connection

次の例は、create-vpn-connection を使用する方法を説明しています。

AWS CLI

例 1: 動的ルーティングを使用してVPN接続を作成するには

次のcreate-vpn-connection例では、指定された仮想プライベートゲートウェイと指定されたカスタマーゲートウェイ間のVPN接続を作成し、VPN接続にタグを適用します。出力には、カスタマーゲートウェイデバイスの設定情報が XML形式で含まれます。

```
aws ec2 create-vpn-connection \  
  --type ipsec.1 \  
  --customer-gateway-id cgw-001122334455aabbcc \  
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \  
  --tag-specification 'ResourceType=vpn-connection,Tags=[{Key=Name,Value=BGP-  
VPN}]'
```

出力:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "...configuration information...",  
    "CustomerGatewayId": "cgw-001122334455aabbcc",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-123123123123abccab",  
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",  
    "Options": {  
      "EnableAcceleration": false,  
      "StaticRoutesOnly": false,  
      "LocalIpv4NetworkCidr": "0.0.0.0/0",  
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
      "TunnelInsideIpVersion": "ipv4",  
      "TunnelOptions": [  
        {},  
        {}  
      ]  
    },  
    "Routes": [],  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "BGP-VPN"  
      }  
    ]  
  }  
}
```

詳細については、「ユーザーガイド」の「[AWS Site-to-Site VPNの仕組み](#)」を参照してください。AWS Site-to-Site VPN

例 2: 静的ルーティングを使用してVPN接続を作成するには

次のcreate-vpn-connection例では、指定された仮想プライベートゲートウェイと指定されたカスタマーゲートウェイ間のVPN接続を作成します。オプションは静的ルーティングを指定します。出力には、カスタマーゲートウェイデバイスの設定情報が XML 形式で含まれます。

```
aws ec2 create-vpn-connection \  
  --type ipsec.1 \  
  --customer-gateway-id cgw-001122334455aabbc \  
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \  
  --options "{\"StaticRoutesOnly\":true}"
```

出力:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "..configuration information...",  
    "CustomerGatewayId": "cgw-001122334455aabbc",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-123123123123abcab",  
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",  
    "Options": {  
      "EnableAcceleration": false,  
      "StaticRoutesOnly": true,  
      "LocalIpv4NetworkCidr": "0.0.0.0/0",  
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
      "TunnelInsideIpVersion": "ipv4",  
      "TunnelOptions": [  
        {},  
        {}  
      ]  
    },  
    "Routes": [],  
    "Tags": []  
  }  
}
```

詳細については、「ユーザーガイド」の「[AWS Site-to-Site VPNの仕組み](#)」を参照してください。AWS Site-to-Site VPN

例 3: VPN接続を作成し、独自の内部CIDRキーと事前共有キーを指定するには

次のcreate-`vpn-connection`例では、VPN接続を作成し、内部 IP アドレス CIDRブロックと各トンネルのカスタム事前共有キーを指定します。指定された値はCustomerGatewayConfiguration情報で返されます。

```
aws ec2 create-vpn-connection \  
  --type ipsec.1 \  
  --customer-gateway-id cgw-001122334455aabbc \  
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \  
  --options  
  TunnelOptions='[{TunnelInsideCidr=169.254.12.0/30,PreSharedKey=ExamplePreSharedKey1},  
{TunnelInsideCidr=169.254.13.0/30,PreSharedKey=ExamplePreSharedKey2}]'
```

出力:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "..configuration information..",  
    "CustomerGatewayId": "cgw-001122334455aabbc",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-123123123123abcab",  
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",  
    "Options": {  
      "EnableAcceleration": false,  
      "StaticRoutesOnly": false,  
      "LocalIpv4NetworkCidr": "0.0.0.0/0",  
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
      "TunnelInsideIpVersion": "ipv4",  
      "TunnelOptions": [  
        {  
          "OutsideIpAddress": "203.0.113.3",  
          "TunnelInsideCidr": "169.254.12.0/30",  
          "PreSharedKey": "ExamplePreSharedKey1"  
        },  
        {  
          "OutsideIpAddress": "203.0.113.5",  
          "TunnelInsideCidr": "169.254.13.0/30",  
          "PreSharedKey": "ExamplePreSharedKey2"  
        }  
      ]  
    }  
  },  
  "Routes": [],  
  "Tags": []  
}
```

```
}
}
```

詳細については、「ユーザーガイド」の「[AWS Site-to-Site VPNの仕組み](#)」を参照してください。AWS Site-to-Site VPN

例 4: IPv6トラフィックをサポートするVPN接続を作成するには

次のcreate-vpn-connection例では、指定されたトランジットゲートウェイと指定されたカスタマーゲートウェイ間のIPv6トラフィックをサポートするVPN接続を作成します。両方のトンネルのトンネルオプションでは、IKEネゴシエーションを開始 AWS する必要がある を指定します。

```
aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --transit-gateway-id tgw-12312312312312312 \
  --customer-gateway-id cgw-001122334455aabbc \
  --options TunnelInsideIpVersion=ipv6,TunnelOptions=[{StartupAction=start},
{StartupAction=start}]
```

出力:

```
{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "..configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbc",
    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-11111111122222222",
    "TransitGatewayId": "tgw-12312312312312312",
    "Options": {
      "EnableAcceleration": false,
      "StaticRoutesOnly": false,
      "LocalIpv6NetworkCidr": "::/0",
      "RemoteIpv6NetworkCidr": "::/0",
      "TunnelInsideIpVersion": "ipv6",
      "TunnelOptions": [
        {
          "OutsideIpAddress": "203.0.113.3",
          "StartupAction": "start"
        }
      ]
    }
  }
}
```



```
        "OutsideIpAddress": "203.0.113.5",
        "StartupAction": "start"
      }
    ]
  },
  "Routes": [],
  "Tags": []
}
}
```

詳細については、「ユーザーガイド」の「[AWS Site-to-Site VPNの仕組み](#)」を参照してください。AWS Site-to-Site VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpnConnection](#)」の「」を参照してください。

create-vpn-gateway

次の例は、create-vpn-gateway を使用する方法を説明しています。

AWS CLI

仮想プライベートゲートウェイを作成するには

この例では、仮想プライベートゲートウェイを作成します。

コマンド:

```
aws ec2 create-vpn-gateway --type ipsec.1
```

出力:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 64512,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

特定の Amazon 側で仮想プライベートゲートウェイを作成するには ASN

この例では、仮想プライベートゲートウェイを作成し、BGPセッションの Amazon 側の自律システム番号 (ASN) を指定します。

コマンド:

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

出力:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 65001,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpnGateway](#)」の「」を参照してください。

delete-carrier-gateway

次のコード例は、delete-carrier-gateway を使用する方法を示しています。

AWS CLI

キャリアゲートウェイを削除するには

次のdelete-carrier-gateway例では、指定されたキャリアゲートウェイを削除します。

```
aws ec2 delete-carrier-gateway \  
  --carrier-gateway-id cagw-0465cdEXAMPLE1111
```

出力:

```
{
```

```
"CarrierGateway": {
  "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
  "VpcId": "vpc-0c529aEXAMPLE1111",
  "State": "deleting",
  "OwnerId": "123456789012"
}
```

詳細については、[「Amazon Virtual Private Cloud ユーザーガイド」](#)の「[キャリアゲートウェイ](#)」を参照してください。 Amazon Virtual Private Cloud

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteCarrierGateway](#)」の「」を参照してください。

delete-client-vpn-endpoint

次のコード例は、delete-client-vpn-endpoint を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントを削除するには

次のdelete-client-vpn-endpoint例では、指定されたクライアントVPNエンドポイントを削除します。

```
aws ec2 delete-client-vpn-endpoint \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

出力:

```
{
  "Status": {
    "Code": "deleting"
  }
}
```

詳細については、[「クライアント管理者ガイド」](#)の「[クライアントVPNエンドポイント](#)」を参照してください。 AWS VPN

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteClientVpnEndpoint](#)」の「」を参照してください。

delete-client-vpn-route

次の例は、delete-client-vpn-route を使用する方法を説明しています。

AWS CLI

クライアントVPNエンドポイントのルートを削除するには

次のdelete-client-vpn-route例では、クライアントVPNエンドポイントの指定されたサブネットの0.0.0.0/0ルートを削除します。

```
aws ec2 delete-client-vpn-route \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --destination-cidr-block 0.0.0.0/0 \  
  --target-vpc-subnet-id subnet-0123456789abcabca
```

出力:

```
{  
  "Status": {  
    "Code": "deleting"  
  }  
}
```

詳細については、AWS 「クライアントVPN管理者ガイド」の「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteClientVpnRoute](#)」の「」を参照してください。

delete-coip-cidr

次のコード例は、delete-coip-cidr を使用する方法を示しています。

AWS CLI

カスタマー所有 IP (CoIP) アドレスの範囲を削除するには

次のdelete-coip-cidr例では、指定された CoIP プール内の指定された範囲の CoIP アドレスを削除します。

```
aws ec2 delete-coip-cidr \  
  --coip-pool-id coip-pool-0123456789abcabca \  
  --cidr-block 10.0.0.0/24
```

```
--cidr 14.0.0.0/24 \  
--coip-pool-id ipv4pool-coip-1234567890abcdefg
```

出力:

```
{  
  "CoipCidr": {  
    "Cidr": "14.0.0.0/24",  
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"  
  }  
}
```

CoIP の詳細については、AWS Outposts ユーザーガイドの「[カスタマー所有の IP アドレス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCoipCidr](#)」の「」を参照してください。

delete-coip-pool

次の例は、delete-coip-pool を使用する方法を説明しています。

AWS CLI

カスタマー所有 IP (CoIP) アドレスのプールを削除するには

次のdelete-coip-pool例では、CoIP アドレスの CoIP プールを削除します。

```
aws ec2 delete-coip-pool \  
--coip-pool-id ipv4pool-coip-1234567890abcdefg
```

出力:

```
{  
  "CoipPool": {  
    "PoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-coip-1234567890abcdefg"  
  }  
}
```

```
}
```

CoIP の詳細については、AWS Outposts ユーザーガイドの「[カスタマー所有の IP アドレス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCoipPool](#)」の「」を参照してください。

delete-customer-gateway

次の例は、delete-customer-gateway を使用する方法を説明しています。

AWS CLI

カスタマーゲートウェイを削除するには

この例では、指定されたカスタマーゲートウェイを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCustomerGateway](#)」の「」を参照してください。

delete-dhcp-options

次のコード例は、delete-dhcp-options を使用する方法を示しています。

AWS CLI

DHCPオプションセットを削除するには

この例では、指定されたDHCPオプションセットを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDhcpOptions](#)」の「」を参照してください。

delete-egress-only-internet-gateway

次のコード例は、delete-egress-only-internet-gateway を使用する方法を示しています。

AWS CLI

Egress-Only インターネットゲートウェイを削除するには

この例では、指定されたエグレス専用インターネットゲートウェイを削除します。

コマンド:

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

出力:

```
{
  "ReturnCode": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEgressOnlyInternetGateway](#)」の「」を参照してください。

delete-fleets

次のコード例は、delete-fleets を使用する方法を示しています。

AWS CLI

例 1: EC2フリートを削除して関連するインスタンスを終了するには

次のdelete-fleets例では、指定されたEC2フリートを削除し、関連付けられたオンデマンドインスタンスとスポットインスタンスを終了します。

```
aws ec2 delete-fleets \
```

```
--fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
--terminate-instances
```

出力:

```
{  
  "SuccessfulFleetDeletions": [  
    {  
      "CurrentFleetState": "deleted_terminating",  
      "PreviousFleetState": "active",  
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
    }  
  ],  
  "UnsuccessfulFleetDeletions": []  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の[EC2「フリートの削除」](#)を参照してください。

例 2: 関連付けられたインスタンスを終了せずに EC2 フリートを削除するには

次の delete-fleets 例では、関連付けられたオンデマンドインスタンスとスポットインスタンスを終了せずに、指定された EC2 フリートを削除します。

```
aws ec2 delete-fleets \  
--fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
--no-terminate-instances
```

出力:

```
{  
  "SuccessfulFleetDeletions": [  
    {  
      "CurrentFleetState": "deleted_running",  
      "PreviousFleetState": "active",  
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
    }  
  ],  
  "UnsuccessfulFleetDeletions": []  
}
```


詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の[EC2「フリートの削除」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteFleets](#)」の「」を参照してください。

delete-flow-logs

次のコード例は、delete-flow-logs を使用する方法を示しています。

AWS CLI

フローログを削除するには

次のdelete-flow-logs例では、指定されたフローログを削除します。

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

出力:

```
{
  "Unsuccessful": []
}
```

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteFlowLogs](#)」の「」を参照してください。

delete-fpga-image

次のコード例は、delete-fpga-image を使用する方法を示しています。

AWS CLI

Amazon FPGAイメージを削除するには

この例では、指定された を削除しますAFI。

コマンド:

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFpgaImage](#)」の「」を参照してください。

delete-instance-connect-endpoint

次のコード例は、delete-instance-connect-endpoint を使用する方法を示しています。

AWS CLI

EC2 Instance Connect エンドポイントを削除するには

次のdelete-instance-connect-endpoint例では、指定された EC2 Instance Connect エンドポイントを削除します。

```
aws ec2 delete-instance-connect-endpoint \
  --instance-connect-endpoint-id eice-03f5e49b83924bbc7
```

出力:

```
{
  "InstanceConnectEndpoint": {
    "OwnerId": "111111111111",
    "InstanceConnectEndpointId": "eice-0123456789example",
    "InstanceConnectEndpointArn": "arn:aws:ec2:us-east-1:111111111111:instance-connect-endpoint/eice-0123456789example",
    "State": "delete-in-progress",
    "StateMessage": "",
    "NetworkInterfaceIds": [],
    "VpcId": "vpc-0123abcd",
    "AvailabilityZone": "us-east-1d",
    "CreatedAt": "2023-02-07T12:05:37+00:00",
    "SubnetId": "subnet-0123abcd"
  }
}
```

詳細については、「Amazon ユーザーガイド」の[EC2「インスタンス接続エンドポイントの削除」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteInstanceConnectEndpoint](#)」の「」を参照してください。

delete-instance-event-window

次のコード例は、delete-instance-event-window を使用する方法を示しています。

AWS CLI

例 1: イベントウィンドウを削除するには

次のdelete-instance-event-window例では、イベントウィンドウを削除します。

```
aws ec2 delete-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890
```

出力:

```
{  
  "InstanceEventWindowState": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "State": "deleting"  
  }  
}
```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの[「考慮事項」](#)を参照してください。

例 2: イベントウィンドウを強制的に削除するには

次のdelete-instance-event-window例では、イベントウィンドウが現在ターゲットに関連付けられている場合、強制的にイベントウィンドウを削除します。

```
aws ec2 delete-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --force-delete
```

出力:

```
{
  "InstanceEventWindowState": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "State": "deleting"
  }
}
```

イベントウィンドウの制約については、「Amazon EC2ユーザーガイド」の「スケジュールされたイベント」セクションの「[考慮事項](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInstanceEventWindow](#)」の「」を参照してください。

delete-internet-gateway

次の例は、delete-internet-gateway を使用する方法を説明しています。

AWS CLI

インターネットゲートウェイを削除するには

次のdelete-internet-gateway例では、指定されたインターネットゲートウェイを削除します。

```
aws ec2 delete-internet-gateway \
  --internet-gateway-id igw-0d0fb496b3EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「Amazon VPCユーザーガイド」の「[インターネットゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInternetGateway](#)」の「」を参照してください。

delete-ipam-pool

次のコード例は、delete-ipam-pool を使用する方法を示しています。

AWS CLI

IPAM プールを削除するには

この例では、不要になったIPAMプールを削除したいが、プールにCIDRプロビジョニングされているIPAM委任された管理者です。--cascade オプションを使用しないと、プールがCIDRsプロビジョニングされている場合は削除できません。そのため、を使用します--cascade。

このリクエストを完了するには：

で取得できるIPAMプール ID が必要です[describe-ipam-pools](#)。はIPAMホームリージョン--regionである必要があります。

次のdelete-ipam-pool例では、AWS アカウントのIPAMプールを削除します。

```
aws ec2 delete-ipam-pool \  
  --ipam-pool-id ipam-pool-050c886a3ca41cd5b \  
  --cascade \  
  --region us-east-1
```

出力:

```
{  
  "IpamPool": {  
    "OwnerId": "320805250157",  
    "IpamPoolId": "ipam-pool-050c886a3ca41cd5b",  
    "IpamPoolArn": "arn:aws:ec2::320805250157:ipam-pool/ipam-pool-050c886a3ca41cd5b",  
    "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-scope-0a158dde35c51107b",  
    "IpamScopeType": "private",  
    "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",  
    "IpamRegion": "us-east-1",  
    "Locale": "None",  
    "PoolDepth": 1,  
    "State": "delete-in-progress",  
    "Description": "example",  
    "AutoImport": false,  
    "AddressFamily": "ipv4",  
    "AllocationMinNetmaskLength": 0,  
    "AllocationMaxNetmaskLength": 32  
  }  
}
```

```
}
```

詳細については、「Amazon ユーザーガイド」の「[プールの削除](#)」を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIpamPool](#)」の「」を参照してください。

delete-ipam-resource-discovery

次のコード例は、delete-ipam-resource-discovery を使用する方法を示しています。

AWS CLI

リソース検出を削除するには

この例では、組織外のアカウントIPAMと統合するプロセス中に、別の管理者と共有するために作成したデフォルト以外のリソース検出を削除したいIPAM委任されたIPAM管理者です。

このリクエストを完了するには：

は、リソース検出を作成したリージョン--regionである必要があります。 の場合、デフォルトのリソース検出を削除することはできません"IsDefault": true。デフォルトのリソース検出は、 を作成するアカウントで自動的に作成されるものですIPAM。デフォルトのリソース検出を削除するには、 を削除する必要がありますIPAM。

次のdelete-ipam-resource-discovery例では、リソース検出を削除します。

```
aws ec2 delete-ipam-resource-discovery \  
  --ipam-resource-discovery-id ipam-res-disco-0e39761475298ee0f \  
  --region us-east-1
```

出力:

```
{  
  "IpamResourceDiscovery": {  
    "OwnerId": "149977607591",  
    "IpamResourceDiscoveryId": "ipam-res-disco-0e39761475298ee0f",  
    "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-  
discovery/ipam-res-disco-0e39761475298ee0f",  
    "IpamResourceDiscoveryRegion": "us-east-1",  
    "OperatingRegions": [  

```

```
    {
      "RegionName": "us-east-1"
    }
  ],
  "IsDefault": false,
  "State": "delete-in-progress"
}
```

リソース検出の詳細については、「Amazon VPCIPAMユーザーガイド」の[「リソース検出の操作」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIpamResourceDiscovery](#)」の「」を参照してください。

delete-ipam-scope

次のコード例は、delete-ipam-scope を使用する方法を示しています。

AWS CLI

IPAMスコープを削除するには

次のdelete-ipam-scope例では、 を削除しますIPAM。

```
aws ec2 delete-ipam-scope \
  --ipam-scope-id ipam-scope-01c1ebab2b63bd7e4
```

出力:

```
{
  "IpamScope": {
    "OwnerId": "123456789012",
    "IpamScopeId": "ipam-scope-01c1ebab2b63bd7e4",
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-
scope-01c1ebab2b63bd7e4",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
    "IpamRegion": "us-east-1",
    "IpamScopeType": "private",
    "IsDefault": false,
    "Description": "Example description",
    "PoolCount": 0,
  }
}
```

```
    "State": "delete-in-progress"
  }
}
```

詳細については、「[Amazon VPC IPAM ユーザーガイド](#)」の「[スコープの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス DeleteIpamScope](#)」の「」を参照してください。

delete-ipam

次の例は、delete-ipam を使用する方法を説明しています。

AWS CLI

を削除するには IPAM

次のdelete-ipam例では、 を削除しますIPAM。

```
aws ec2 delete-ipam \
  --ipam-id ipam-036486dfa6af58ee0
```

出力:

```
{
  "Ipam": {
    "OwnerId": "123456789012",
    "IpamId": "ipam-036486dfa6af58ee0",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-036486dfa6af58ee0",
    "IpamRegion": "us-east-1",
    "PublicDefaultScopeId": "ipam-scope-071b8042b0195c183",
    "PrivateDefaultScopeId": "ipam-scope-0807405dece705a30",
    "ScopeCount": 2,
    "OperatingRegions": [
      {
        "RegionName": "us-east-1"
      },
      {
        "RegionName": "us-east-2"
      },
      {
```



```
        "RegionName": "us-west-1"
      }
    ],
    "State": "delete-in-progress"
  }
}
```

詳細については、「[Amazon ユーザーガイド](#)」の「[を削除するIPAM](#)」を参照してください。

VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIpam](#)」の「」を参照してください。

delete-key-pair

次のコード例は、delete-key-pair を使用する方法を示しています。

AWS CLI

キーペアを削除するには

次のdelete-key-pair例では、指定されたキーペアを削除します。

```
aws ec2 delete-key-pair \  
  --key-name my-key-pair
```

出力:

```
{  
  "Return": true,  
  "KeyId": "key-03c8d3aceb53b507"  
}
```

詳細については、AWS 「[コマンドラインインターフェイスユーザーガイド](#)」の「[キーペアの作成と削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteKeyPair](#)」の「」を参照してください。

delete-launch-template-versions

次のコード例は、delete-launch-template-versions を使用する方法を示しています。

AWS CLI

起動テンプレートのバージョンを削除するには

この例では、指定された起動テンプレートバージョンを削除します。

コマンド:

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --versions 1
```

出力:

```
{
  "UnsuccessfullyDeletedLaunchTemplateVersions": [],
  "SuccessfullyDeletedLaunchTemplateVersions": [
    {
      "LaunchTemplateName": "TestVersion",
      "VersionNumber": 1,
      "LaunchTemplateId": "lt-0abcd290751193123"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLaunchTemplateVersions](#)」の「」を参照してください。

delete-launch-template

次の例は、delete-launch-template を使用する方法を説明しています。

AWS CLI

起動テンプレートを削除するには

次の例では、指定した起動テンプレートを削除しています。

コマンド:

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

出力:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "TestTemplate",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLaunchTemplate](#)」の「」を参照してください。

delete-local-gateway-route-table-virtual-interface-group-association

次の例は、delete-local-gateway-route-table-virtual-interface-group-association を使用する方法を説明しています。

AWS CLI

仮想インターフェイス (VIFs) グループからローカルゲートウェイルートテーブルの関連付けを解除するには

次のdelete-local-gateway-route-table-virtual-interface-group-association例では、指定されたローカルゲートウェイルートテーブルとVIFグループ間の関連付けを削除します。

```
aws ec2 delete-local-gateway-route-table-virtual-interface-group-association \
  --local-gateway-route-table-virtual-interface-group-association-id lgw-vif-grp-  
assoc-exampleid12345678
```

出力:

```
{
  "LocalGatewayRouteTableVirtualInterfaceGroupAssociation": {
    "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-  
assoc-exampleid12345678",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-exampleid0123abcd",
    "LocalGatewayId": "lgw-exampleid11223344",
    "LocalGatewayRouteTableId": "lgw-rtb-exampleidabcd1234",
```

```
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-  
gateway-route-table/lgw-rtb-exampleidabcd1234",  
    "OwnerId": "111122223333",  
    "State": "disassociating",  
    "Tags": []  
  }  
}
```

詳細については、[VIF「Outposts ユーザーガイド」の「グループ関連付けAWS」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociation](#)」の「」を参照してください。

delete-local-gateway-route-table-vpc-association

次の例は、delete-local-gateway-route-table-vpc-association を使用する方法を説明しています。

AWS CLI

ローカルゲートウェイルートテーブルと の関連付けを解除するには VPC

次のdelete-local-gateway-route-table-vpc-association例では、指定されたローカルゲートウェイルートテーブルと の関連付けを削除しますVPC。

```
aws ec2 delete-local-gateway-route-table-vpc-association \  
--local-gateway-route-table-vpc-association-id vpc-example0123456789
```

出力:

```
{  
  "LocalGatewayRouteTableVpcAssociation": {  
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-abcd1234wxyz56789",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:555555555555:local-  
gateway-route-table/lgw-rtb-abcdefg1234567890",  
    "LocalGatewayId": "lgw-exampleid01234567",  
    "VpcId": "vpc-example0123456789",  
    "OwnerId": "555555555555",  
    "State": "disassociating"  
  }  
}
```

```
}  
}
```

詳細については、AWS 「Outposts ユーザーガイド」の[VPC「関連付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLocalGatewayRouteTableVpcAssociation](#)」の「」を参照してください。

delete-local-gateway-route-table

次のコード例は、delete-local-gateway-route-table を使用する方法を示しています。

AWS CLI

ローカルゲートウェイルートテーブルを削除するには

次のdelete-local-gateway-route-table例では、ダイレクトルーティングモードでローカルゲートウェイVPCルートテーブルを作成します。

```
aws ec2 delete-local-gateway-route-table \  
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

出力:

```
{  
  "LocalGatewayRouteTable": {  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-gateway-route-table/lgw-rtb-abcdefg1234567890",  
    "LocalGatewayId": "lgw-1a2b3c4d5e6f7g8h9",  
    "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/op-021345abcdef67890",  
    "OwnerId": "111122223333",  
    "State": "deleting",  
    "Tags": [],  
    "Mode": "direct-vpc-routing"  
  }  
}
```

詳細については、「AWS Outposts ユーザーガイド」の「[ローカルゲートウェイテーブル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLocalGatewayRouteTable](#)」の「」を参照してください。

delete-local-gateway-route

次の例は、delete-local-gateway-route を使用する方法を説明しています。

AWS CLI

ローカルゲートウェイルートテーブルからルートを削除するには

次のdelete-local-gateway-route例では、指定されたローカルゲートウェイルートテーブルから指定されたルートを削除します。

```
aws ec2 delete-local-gateway-route \  
  --destination-cidr-block 0.0.0.0/0 \  
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

出力:

```
{  
  "Route": {  
    "DestinationCidrBlock": "0.0.0.0/0",  
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",  
    "Type": "static",  
    "State": "deleted",  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLocalGatewayRoute](#)」の「」を参照してください。

delete-managed-prefix-list

次の例は、delete-managed-prefix-list を使用する方法を説明しています。

AWS CLI

プレフィックスリストを削除するには

次のdelete-managed-prefix-list例では、指定されたプレフィックスリストを削除します。

```
aws ec2 delete-managed-prefix-list \  
  --prefix-list-id pl-0123456abcabcabc1
```

出力:

```
{  
  "PrefixList": {  
    "PrefixListId": "pl-0123456abcabcabc1",  
    "AddressFamily": "IPv4",  
    "State": "delete-in-progress",  
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-0123456abcabcabc1",  
    "PrefixListName": "test",  
    "MaxEntries": 10,  
    "Version": 1,  
    "OwnerId": "123456789012"  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の「[マネージドプレフィックスリスト](#)」を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteManagedPrefixList](#)」の「」を参照してください。

delete-nat-gateway

次のコード例は、delete-nat-gateway を使用する方法を示しています。

AWS CLI

NATゲートウェイを削除するには

この例では、NATゲートウェイ を削除しますnat-04ae55e711cec5680。

コマンド:

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

出力:

```
{
  "NatGatewayId": "nat-04ae55e711cec5680"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNatGateway](#)」の「」を参照してください。

delete-network-acl-entry

次のコード例は、delete-network-acl-entry を使用する方法を示しています。

AWS CLI

ネットワークACLエントリを削除するには

この例では、指定されたネットワーク から進入ルール番号 100 を削除しますACL。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNetworkAclEntry](#)」の「」を参照してください。

delete-network-acl

次の例は、delete-network-acl を使用する方法を説明しています。

AWS CLI

ネットワークを削除するには ACL

この例では、指定されたネットワーク を削除しますACL。コマンドが成功した場合、出力は返りません。

コマンド:


```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNetworkAcl](#)」の「」を参照してください。

delete-network-insights-access-scope-analysis

次の例は、delete-network-insights-access-scope-analysis を使用する方法を説明しています。

AWS CLI

ネットワークアクセススコープ分析を削除するには

次のdelete-network-insights-access-scope-analysis例では、指定されたネットワークアクセススコープ分析を削除します。

```
aws ec2 delete-network-insights-access-scope-analysis \  
--network-insights-access-scope-analysis-id nisa-01234567891abcdef
```

出力:

```
{  
  "NetworkInsightsAccessScopeAnalysisId": "nisa-01234567891abcdef"  
}
```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNetworkInsightsAccessScopeAnalysis](#)」の「」を参照してください。

delete-network-insights-access-scope

次のコード例は、delete-network-insights-access-scope を使用する方法を示しています。

AWS CLI

ネットワークアクセススコープを削除するには

次のdelete-network-insights-access-scope例では、指定されたネットワークアクセススコープを削除します。

```
aws ec2 delete-network-insights-access-scope \  
  --network-insights-access-scope-id nis-123456789abc01234
```

出力:

```
{  
  "NetworkInsightsAccessScopeId": "nis-123456789abc01234"  
}
```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンスDeleteNetworkInsightsAccessScope」の「」を参照してください。

delete-network-insights-analysis

次の例は、delete-network-insights-analysis を使用する方法を説明しています。

AWS CLI

パス分析を削除するには

次のdelete-network-insights-analysis例では、指定された分析を削除します。

```
aws ec2 delete-network-insights-analysis \  
  --network-insights-analysis-id nia-02207aa13eb480c7a
```

出力:

```
{  
  "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"  
}
```

詳細については、「Reachability Analyzer ガイド」の「[の使用開始 AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteNetworkInsightsAnalysis](#)」の「」を参照してください。

delete-network-insights-path

次の例は、delete-network-insights-path を使用する方法を説明しています。

AWS CLI

パスを削除するには

次のdelete-network-insights-path例では、指定されたパスを削除します。パスを削除する前に、delete-network-insights-analysis コマンドを使用してすべての分析を削除する必要があります。

```
aws ec2 delete-network-insights-path \  
  --network-insights-path-id nip-0b26f224f1d131fa8
```

出力:

```
{  
  "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"  
}
```

詳細については、「Reachability Analyzer ガイド」の「[の使用を開始する AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteNetworkInsightsPath](#)」の「」を参照してください。

delete-network-interface-permission

次の例は、delete-network-interface-permission を使用する方法を説明しています。

AWS CLI

ネットワークインターフェイスのアクセス許可を削除するには

この例では、指定されたネットワークインターフェイスのアクセス許可を削除します。

コマンド:

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-perm-06fd19020ede149ea
```

出力:

```
{  
  "Return": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNetworkInterfacePermission](#)」の「」を参照してください。

delete-network-interface

次の例は、delete-network-interface を使用する方法を説明しています。

AWS CLI

ネットワークインターフェイスを削除するには

この例では、指定されたネットワークインターフェイスを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNetworkInterface](#)」の「」を参照してください。

delete-placement-group

次の例は、delete-placement-group を使用する方法を説明しています。

AWS CLI

プレイズメントグループを削除するには

この例では、指定されたプレースメントグループを削除します。

コマンド:

```
aws ec2 delete-placement-group --group-name my-cluster
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePlacementGroup](#)」の「」を参照してください。

delete-queued-reserved-instances

次のコード例は、delete-queued-reserved-instances を使用する方法を示しています。

AWS CLI

キューに入れられた購入を削除するには

次のdelete-queued-reserved-instances例では、指定されたリザーブドインスタンスを削除しています。このリザーブドインスタンスは、購入のためにキューに入れられています。

```
aws ec2 delete-queued-reserved-instances \  
  --reserved-instances-ids af9f760e-6f91-4559-85f7-4980eexample
```

出力:

```
{  
  "SuccessfulQueuedPurchaseDeletions": [  
    {  
      "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"  
    }  
  ],  
  "FailedQueuedPurchaseDeletions": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteQueuedReservedInstances](#)」の「」を参照してください。

delete-route-table

次のコード例は、delete-route-table を使用する方法を示しています。

AWS CLI

ルートテーブルを削除するには

この例では、指定されたルートテーブルを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRouteTable](#)」の「」を参照してください。

delete-route

次の例は、delete-route を使用する方法を説明しています。

AWS CLI

ルートを削除するには

この例では、指定されたルートテーブルから指定されたルートを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRoute](#)」の「」を参照してください。

delete-security-group

次の例は、delete-security-group を使用する方法を説明しています。

AWS CLI

[EC2-Classic] セキュリティグループを削除するには

この例では、MySecurityGroup という名前のセキュリティグループを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

〔EC2-VPC〕セキュリティグループを削除するには

この例では、sg-903004f8 という ID のセキュリティグループを削除します。EC2-VPC のセキュリティグループを名前で参照することはできません。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-security-group --group-id sg-903004f8
```

詳細については、「AWS コマンドラインインターフェイスユーザーガイド」でセキュリティグループの使用方法を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSecurityGroup](#)」の「」を参照してください。

delete-snapshot

次のコード例は、delete-snapshot を使用する方法を示しています。

AWS CLI

スナップショットを削除するには

このコマンド例は、スナップショット ID が snap-1234567890abcdef0 のスナップショットを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSnapshot](#)」の「」を参照してください。

delete-spot-datafeed-subscription

次のコード例は、delete-spot-datafeed-subscription を使用する方法を示しています。

AWS CLI

スポットインスタンスのデータフィードサブスクリプションをキャンセルするには

この例では、アカウントのスポットデータフィードサブスクリプションを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-spot-datafeed-subscription
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteSpotDatafeedSubscription](#)」の「」を参照してください。

delete-subnet-cidr-reservation

次の例は、delete-subnet-cidr-reservation を使用する方法を説明しています。

AWS CLI

サブネットCIDR予約を削除するには

次のdelete-subnet-cidr-reservation例では、指定されたサブネットCIDR予約を削除します。

```
aws ec2 delete-subnet-cidr-reservation \  
--subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

出力:

```
{  
  "DeletedSubnetCidrReservation": {  
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",  
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",  
    "Cidr": "10.1.0.16/28",  
    "ReservationType": "prefix",
```



```
    "OwnerId": "123456789012"  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の「[サブネットCIDR予約](#)」を参照してください。
VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSubnetCidrReservation](#)」の「」を参照してください。

delete-subnet

次のコード例は、delete-subnet を使用する方法を示しています。

AWS CLI

サブネットを削除するには

この例では、指定されたサブネットを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSubnet](#)」の「」を参照してください。

delete-tags

次の例は、delete-tags を使用する方法を説明しています。

AWS CLI

例 1: リソースからタグを削除するには

次のdelete-tags例では、指定されたイメージStack=Testから タグを削除します。値とキー名の両方を指定すると、タグの値が指定された値と一致する場合にのみ、タグは削除されます。

```
aws ec2 delete-tags \  

```

```
--resources ami-1234567890abcdef0 \  
--tags Key=Stack,Value=Test
```

タグの値を指定することはオプションです。次のdelete-tags例では、タグのタグ値に関係なく、指定されたインスタンスpurposeからキー名のタグを削除します。

```
aws ec2 delete-tags \  
--resources i-1234567890abcdef0 \  
--tags Key=purpose
```

空の文字列をタグ値として指定すると、タグの値が空の文字列である場合にのみタグが削除されます。次のdelete-tags例では、空の文字列を、削除するタグのタグ値として指定します。

```
aws ec2 delete-tags \  
--resources i-1234567890abcdef0 \  
--tags Key=Name,Value=
```

例 2: 複数のリソースからタグを削除するには

次のdelete-tags例では、インスタンスとAMIの両方からタグ「Purpose=Test」を削除します。前の例に示すように、コマンドからタグ値を省略できます。

```
aws ec2 delete-tags \  
--resources i-1234567890abcdef0 ami-1234567890abcdef0 \  
--tags Key=Purpose
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTags](#)」の「」を参照してください。

delete-traffic-mirror-filter-rule

次のコード例は、delete-traffic-mirror-filter-rule を使用する方法を示しています。

AWS CLI

トラフィックミラーフィルタールールを削除するには

次のdelete-traffic-mirror-filter-rule例では、指定されたトラフィックミラーフィルタールールを削除します。

```
aws ec2 delete-traffic-mirror-filter-rule \  
  --traffic-mirror-filter-rule-id tmfr-081f71283bEXAMPLE
```

出力:

```
{  
  "TrafficMirrorFilterRuleId": "tmfr-081f71283bEXAMPLE"  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーフィルターの変更AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTrafficMirrorFilterRule](#)」の「」を参照してください。

delete-traffic-mirror-filter

次のコード例は、delete-traffic-mirror-filter を使用する方法を示しています。

AWS CLI

トラフィックミラーフィルターを削除するには

次のdelete-traffic-mirror-filter例では、指定されたトラフィックミラーフィルターを削除します。

```
aws ec2 delete-traffic-mirror-filter \  
  --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

出力:

```
{  
  "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーフィルターの削除AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTrafficMirrorFilter](#)」の「」を参照してください。

delete-traffic-mirror-session

次の例は、delete-traffic-mirror-session を使用する方法を説明しています。

AWS CLI

トラフィックミラーセッションを削除するには

次のdelete-traffic-mirror-session例では、指定されたトラフィックミラーセッションを削除します。

```
aws ec2 delete-traffic-mirror-session \  
  --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

出力:

```
{  
  "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーセッションの削除AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTrafficMirrorSession](#)」の「」を参照してください。

delete-traffic-mirror-target

次の例は、delete-traffic-mirror-target を使用する方法を説明しています。

AWS CLI

トラフィックミラーターゲットを削除するには

次のdelete-traffic-mirror-target例では、指定されたトラフィックミラーターゲットを削除します。

```
aws ec2 delete-traffic-mirror-target \  
  --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

出力:

```
{
  "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーターゲットの削除AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTrafficMirrorTarget](#)」の「」を参照してください。

delete-transit-gateway-connect-peer

次のコード例は、delete-transit-gateway-connect-peer を使用する方法を示しています。

AWS CLI

Transit Gateway Connect ピアを削除するには

次のdelete-transit-gateway-connect-peer例では、指定された Connect ピアを削除します。

```
aws ec2 delete-transit-gateway-connect-peer \
  --transit-gateway-connect-peer-id tgw-connect-peer-0666adbac4EXAMPLE
```

出力:

```
{
  "TransitGatewayConnectPeer": {
    "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",
    "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",
    "State": "deleting",
    "CreationTime": "2021-10-13T03:35:17.000Z",
    "ConnectPeerConfiguration": {
      "TransitGatewayAddress": "10.0.0.234",
      "PeerAddress": "172.31.1.11",
      "InsideCidrBlocks": [
        "169.254.6.0/29"
      ],
      "Protocol": "gre",
      "BgpConfigurations": [
        {
```

```
        "TransitGatewayAsn": 64512,
        "PeerAsn": 64512,
        "TransitGatewayAddress": "169.254.6.2",
        "PeerAddress": "169.254.6.1",
        "BgpStatus": "down"
    },
    {
        "TransitGatewayAsn": 64512,
        "PeerAsn": 64512,
        "TransitGatewayAddress": "169.254.6.3",
        "PeerAddress": "169.254.6.1",
        "BgpStatus": "down"
    }
]
}
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway Connect アタッチメント」と「Transit Gateway Connect ピア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTransitGatewayConnectPeer](#)」の「」を参照してください。

delete-transit-gateway-connect

次の例は、delete-transit-gateway-connect を使用する方法を説明しています。

AWS CLI

Transit Gateway Connect アタッチメントを削除するには

次のdelete-transit-gateway-connect例では、指定された Connect アタッチメントを削除します。

```
aws ec2 delete-transit-gateway-connect \
  --transit-gateway-attachment-id tgw-attach-037012e5dcEXAMPLE
```

出力:

```
{
```

```
"TransitGatewayConnect": {
  "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
  "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
  "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
  "State": "deleting",
  "CreationTime": "2021-03-09T19:59:17+00:00",
  "Options": {
    "Protocol": "gre"
  }
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway Connect アタッチメント」と「Transit Gateway Connect ピア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTransitGatewayConnect](#)」の「」を参照してください。

delete-transit-gateway-multicast-domain

次の例は、delete-transit-gateway-multicast-domain を使用する方法を説明しています。

AWS CLI

Transit Gateway マルチキャストドメインを削除するには

次のdelete-transit-gateway-multicast-domain例では、指定されたマルチキャストドメインを削除します。

```
aws ec2 delete-transit-gateway-multicast-domain \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

出力:

```
{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-02bb79002bEXAMPLE",
    "TransitGatewayId": "tgw-0d88d2d0d5EXAMPLE",
    "State": "deleting",
    "CreationTime": "2019-11-20T22:02:03.000Z"
  }
}
```

```
}
```

詳細については、「Transit Gateways ガイド」の「[マルチキャストドメインの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteTransitGatewayMulticastDomain](#)」の「」を参照してください。

delete-transit-gateway-peering-attachment

次のコード例は、delete-transit-gateway-peering-attachment を使用する方法を示しています。

AWS CLI

Transit Gateway ピアリングアタッチメントを削除するには

次のdelete-transit-gateway-peering-attachment例では、指定された Transit Gateway ピアリングアタッチメントを削除します。

```
aws ec2 delete-transit-gateway-peering-attachment \  
--transit-gateway-attachment-id tgw-attach-4455667788aabbccd
```

出力:

```
{  
  "TransitGatewayPeeringAttachment": {  
    "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",  
    "RequesterTgwInfo": {  
      "TransitGatewayId": "tgw-123abc05e04123abc",  
      "OwnerId": "123456789012",  
      "Region": "us-west-2"  
    },  
    "AcceptorTgwInfo": {  
      "TransitGatewayId": "tgw-11223344aabbcc112",  
      "OwnerId": "123456789012",  
      "Region": "us-east-2"  
    },  
    "State": "deleting",  
    "CreationTime": "2019-12-09T11:38:31.000Z"  
  }  
}
```



```
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ピアリングアタッチメント」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteTransitGatewayPeeringAttachment](#)」の「」を参照してください。

delete-transit-gateway-policy-table

次の例は、delete-transit-gateway-policy-table を使用する方法を説明しています。

AWS CLI

トランジットゲートウェイポリシーテーブルを削除するには

次のdelete-transit-gateway-policy-table例では、指定された Transit Gateway ポリシーテーブルを削除します。

```
aws ec2 delete-transit-gateway-policy-table \  
  --transit-gateway-policy-table-id tgw-ptb-0a16f134b78668a81
```

出力:

```
{  
  "TransitGatewayPolicyTables": [  
    {  
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",  
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",  
      "State": "deleting",  
      "CreationTime": "2023-11-28T16:36:43+00:00",  
      "Tags": []  
    }  
  ]  
}
```

詳細については、[「Transit Gateway ユーザーガイド」の「Transit Gateway ポリシーテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteTransitGatewayPolicyTable](#)」の「」を参照してください。

delete-transit-gateway-prefix-list-reference

次の例は、delete-transit-gateway-prefix-list-reference を使用する方法を説明しています。

AWS CLI

プレフィックスリストリファレンスを削除するには

次のdelete-transit-gateway-prefix-list-reference例では、指定されたプレフィックスリストリファレンスを削除します。

```
aws ec2 delete-transit-gateway-prefix-list-reference \  
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \  
  --prefix-list-id pl-1111112222222333
```

出力:

```
{  
  "TransitGatewayPrefixListReference": {  
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",  
    "PrefixListId": "pl-1111112222222333",  
    "PrefixListOwnerId": "123456789012",  
    "State": "deleting",  
    "Blackhole": false,  
    "TransitGatewayAttachment": {  
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-112233445566aabbcc"  
    }  
  }  
}
```

詳細については、「Transit Gateways ガイド」の「[プレフィックスリストリファレンス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteTransitGatewayPrefixListReference](#)」の「」を参照してください。

delete-transit-gateway-route-table

次のコード例は、delete-transit-gateway-route-table を使用する方法を示しています。

AWS CLI

Transit Gateway ルートテーブルを削除するには

次のdelete-transit-gateway-route-table例では、指定された Transit Gateway ルートテーブルを削除します。

```
aws ec2 delete-transit-gateway-route-table \  
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE
```

出力:

```
{  
  "TransitGatewayRouteTable": {  
    "TransitGatewayRouteTableId": "tgw-rtb-0b6f6aaa01EXAMPLE",  
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",  
    "State": "deleting",  
    "DefaultAssociationRouteTable": false,  
    "DefaultPropagationRouteTable": false,  
    "CreationTime": "2019-07-17T20:27:26.000Z"  
  }  
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブルの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTransitGatewayRouteTable](#)」の「」を参照してください。

delete-transit-gateway-route

次の例は、delete-transit-gateway-route を使用する方法を説明しています。

AWS CLI

ルートテーブルからCIDRブロックを削除するには

次のdelete-transit-gateway-route例では、指定された Transit Gateway ルートテーブルから CIDR ブロックを削除します。

```
aws ec2 delete-transit-gateway-route \  
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \  
  --cidr-block 10.0.0.0/24
```

```
--destination-cidr-block 10.0.2.0/24
```

出力:

```
{
  "Route": {
    "DestinationCidrBlock": "10.0.2.0/24",
    "TransitGatewayAttachments": [
      {
        "ResourceId": "vpc-0065acced4EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
        "ResourceType": "vpc"
      }
    ],
    "Type": "static",
    "State": "deleted"
  }
}
```

詳細については、「Transit Gateways ガイド」の「[静的ルートの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTransitGatewayRoute](#)」の「」を参照してください。

delete-transit-gateway-vpc-attachment

次の例は、delete-transit-gateway-vpc-attachment を使用する方法を説明しています。

AWS CLI

Transit Gateway アタッチメントを削除するには VPC

次のdelete-transit-gateway-vpc-attachment例では、指定されたVPC添付ファイルを削除します。

```
aws ec2 delete-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0d2c54bdbEXAMPLE
```

出力:

```
{
  "TransitGatewayVpcAttachment": {
```

```
    "TransitGatewayAttachmentId": "tgw-attach-0d2c54bdb3EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "VpcId": "vpc-0065acced4f61c651",
    "VpcOwnerId": "111122223333",
    "State": "deleting",
    "CreationTime": "2019-07-17T16:04:27.000Z"
  }
}
```

詳細については、「Transit Gateways ガイド」の[VPC「添付ファイルの削除」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteTransitGatewayVpcAttachment](#)」の「」を参照してください。

delete-transit-gateway

次の例は、delete-transit-gateway を使用する方法を説明しています。

AWS CLI

トランジットゲートウェイを削除するには

次のdelete-transit-gateway例では、指定されたトランジットゲートウェイを削除します。

```
aws ec2 delete-transit-gateway \  
  --transit-gateway-id tgw-01f04542b2EXAMPLE
```

出力:

```
{  
  "TransitGateway": {  
    "TransitGatewayId": "tgw-01f04542b2EXAMPLE",  
    "State": "deleting",  
    "OwnerId": "123456789012",  
    "Description": "Example Transit Gateway",  
    "CreationTime": "2019-08-27T15:04:35.000Z",  
    "Options": {  
      "AmazonSideAsn": 64515,  
      "AutoAcceptSharedAttachments": "disable",  
      "DefaultRouteTableAssociation": "enable",  
      "AssociationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",  
      "DefaultRouteTablePropagation": "enable",
```

```

        "PropagationDefaultRouteTableId": "tgw-rtb-0ce7a6948fEXAMPLE",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
    }
}
}

```

詳細については、[「トランジットゲートウェイガイド」の「トランジットゲートウェイの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTransitGateway](#)」の「」を参照してください。

delete-verified-access-endpoint

次のコード例は、delete-verified-access-endpoint を使用する方法を示しています。

AWS CLI

Verified Access エンドポイントを削除するには

次のdelete-verified-access-endpoint例では、指定された Verified Access エンドポイントを削除します。

```

aws ec2 delete-verified-access-endpoint \
  --verified-access-endpoint-id vae-066fac616d4d546f2

```

出力:

```

{
  "VerifiedAccessEndpoint": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
    "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
    "ApplicationDomain": "example.com",
    "EndpointType": "network-interface",
    "AttachmentType": "vpc",
    "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
    "EndpointDomain": "my-ava-app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-east-2.amazonaws.com",
  }
}

```

```
    "SecurityGroupIds": [
      "sg-004915970c4c8f13a"
    ],
    "NetworkInterfaceOptions": {
      "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
      "Protocol": "https",
      "Port": 443
    },
    "Status": {
      "Code": "deleting"
    },
    "Description": "Testing Verified Access",
    "CreationTime": "2023-08-25T20:54:43",
    "LastUpdatedTime": "2023-08-25T22:46:32"
  }
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access エンドポイント」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVerifiedAccessEndpoint](#)」の「」を参照してください。

delete-verified-access-group

次の例は、delete-verified-access-group を使用する方法を説明しています。

AWS CLI

Verified Access グループを削除するには

次のdelete-verified-access-group例では、指定された Verified Access グループを削除します。

```
aws ec2 delete-verified-access-group \
  --verified-access-group-id vagr-0dbe967baf14b7235
```

出力:

```
{
  "VerifiedAccessGroup": {
```

```
"VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
"VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
"Description": "Testing Verified Access",
"Owner": "123456789012",
"VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
"CreationTime": "2023-08-25T19:55:19",
"LastUpdatedTime": "2023-08-25T22:49:03",
"DeletionTime": "2023-08-26T00:58:31"
}
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access グループ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVerifiedAccessGroup](#)」の「」を参照してください。

delete-verified-access-instance

次のコード例は、delete-verified-access-instance を使用する方法を示しています。

AWS CLI

Verified Access インスタンスを削除するには

次のdelete-verified-access-instance例では、指定された Verified Access インスタンスを削除します。

```
aws ec2 delete-verified-access-instance \
  --verified-access-instance-id vai-0ce000c0b7643abea
```

出力:

```
{
  "VerifiedAccessInstance": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "Testing Verified Access",
    "VerifiedAccessTrustProviders": [],
    "CreationTime": "2023-08-25T18:27:56",
    "LastUpdatedTime": "2023-08-26T01:00:18"
  }
}
```



```
}
```

詳細については、[「Verified Access ユーザーガイド」](#)の「[Verified Access インスタンス](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVerifiedAccessInstance](#)」の「」を参照してください。

delete-verified-access-trust-provider

次の例は、delete-verified-access-trust-provider を使用する方法を説明しています。

AWS CLI

Verified Access 信頼プロバイダーを削除するには

次のdelete-verified-access-trust-provider例では、指定された Verified Access 信頼プロバイダーを削除します。

```
aws ec2 delete-verified-access-trust-provider \  
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

出力:

```
{  
  "VerifiedAccessTrustProvider": {  
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
    "Description": "Testing Verified Access",  
    "TrustProviderType": "user",  
    "UserTrustProviderType": "iam-identity-center",  
    "PolicyReferenceName": "idc",  
    "CreationTime": "2023-08-25T18:40:36",  
    "LastUpdatedTime": "2023-08-25T18:40:36"  
  }  
}
```

詳細については、[「Verified Access ユーザーガイド」](#)の「[Verified Access の信頼プロバイダー](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVerifiedAccessTrustProvider](#)」の「」を参照してください。

delete-volume

次の例は、delete-volume を使用する方法を説明しています。

AWS CLI

ボリュームを削除するには

この例では、ボリューム ID が の使用可能なボリュームを削除します vol-049df61146c4d7901。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteVolume](#)」の「」を参照してください。

delete-vpc-endpoint-connection-notifications

次の例は、delete-vpc-endpoint-connection-notifications を使用する方法を説明しています。

AWS CLI

エンドポイント接続通知を削除するには

この例では、指定されたエンドポイント接続通知を削除します。

コマンド:

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-ids vpce-nfn-008776de7e03f5abc
```

出力:

```
{  
  "Unsuccessful": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteVpcEndpointConnectionNotifications](#)」の「」を参照してください。

delete-vpc-endpoint-service-configurations

次の例は、delete-vpc-endpoint-service-configurations を使用する方法を説明しています。

AWS CLI

エンドポイントサービス設定を削除するには

この例では、指定されたエンドポイントサービス設定を削除します。

コマンド:

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-svc-03d5ebb7d9579a2b3
```

出力:

```
{
  "Unsuccessful": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteVpcEndpointServiceConfigurations](#)」の「」を参照してください。

delete-vpc-endpoints

次の例は、delete-vpc-endpoints を使用する方法を説明しています。

AWS CLI

エンドポイントを削除するには

この例では、エンドポイント vpce-aa22bb33 と vpce-1a2b3c4d を削除します。コマンドが部分的に成功または失敗した場合、失敗した項目のリストが返されます。コマンドが成功すると、返されたリストは空になります。

コマンド:

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

出力:

```
{
  "Unsuccessful": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVpcEndpoints](#)」の「」を参照してください。

delete-vpc-peering-connection

次の例は、delete-vpc-peering-connection を使用する方法を説明しています。

AWS CLI

VPCピアリング接続を削除するには

この例では、指定されたVPCピアリング接続を削除します。

コマンド:

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVpcPeeringConnection](#)」の「」を参照してください。

delete-vpc

次のコード例は、delete-vpc を使用する方法を示しています。

AWS CLI

を削除するには VPC

この例では、指定された を削除しますVPC。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVpc](#)」の「」を参照してください。

delete-vpn-connection-route

次のコード例は、delete-vpn-connection-route を使用する方法を示しています。

AWS CLI

VPN接続から静的ルートを削除するには

この例では、指定された静的ルートを指定されたVPN接続から削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVpnConnectionRoute](#)」の「」を参照してください。

delete-vpn-connection

次の例は、delete-vpn-connection を使用する方法を説明しています。

AWS CLI

VPN接続を削除するには

この例では、指定されたVPN接続を削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVpnConnection](#)」の「」を参照してください。

delete-vpn-gateway

次のコード例は、delete-vpn-gateway を使用する方法を示しています。

AWS CLI

仮想プライベートゲートウェイを削除するには

この例では、指定された仮想プライベートゲートウェイを削除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVpnGateway](#)」の「」を参照してください。

deprovision-byoip-cidr

次のコード例は、deprovision-byoip-cidr を使用する方法を示しています。

AWS CLI

IP アドレス範囲の使用を削除するには

次の例では、指定されたアドレス範囲をでの使用から削除します AWS。

```
aws ec2 deprovision-byoip-cidr \  
--cidr 203.0.113.25/24
```

出力:

```
{  
  "ByoipCidr": {  
    "Cidr": "203.0.113.25/24",  
    "State": "pending-deprovision"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeprovisionByoipCidr](#)」の「」を参照してください。

deprovision-ipam-pool-cidr

次の例は、deprovision-ipam-pool-cidr を使用する方法を説明しています。

AWS CLI

IPAM プールのプロビジョニングを解除するには CIDR

次のdeprovision-ipam-pool-cidr例では、IPAMプールにCIDRプロビジョニングされた をプロビジョニング解除します。

(Linux):

```
aws ec2 deprovision-ipam-pool-cidr \  
  --ipam-pool-id ipam-pool-02ec043a19bbe5d08 \  
  --cidr 11.0.0.0/16
```

(Windows):

```
aws ec2 deprovision-ipam-pool-cidr ^  
  --ipam-pool-id ipam-pool-02ec043a19bbe5d08 ^  
  --cidr 11.0.0.0/16
```

出力:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "11.0.0.0/16",  
    "State": "pending-deprovision"  
  }  
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の「[プールのプロビジョニング解除 CIDRs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeprovisionIpamPoolCidr](#)」の「」を参照してください。

deregister-image

次のコード例は、deregister-image を使用する方法を示しています。

AWS CLI

の登録を解除するには AMI

この例では、指定された を登録解除しますAMI。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterImage](#)」の「」を参照してください。

deregister-instance-event-notification-attributes

次のコード例は、deregister-instance-event-notification-attributes を使用する方法を示しています。

AWS CLI

例 1: イベント通知からすべてのタグを削除するには

次のderegister-instance-event-notification-attributes例ではIncludeAllTagsOfInstance=true、 を削除します。これは、IncludeAllTagsOfInstance を に設定する効果がありますfalse。

```
aws ec2 deregister-instance-event-notification-attributes \  
--instance-tag-attribute IncludeAllTagsOfInstance=true
```

出力:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [],  
    "IncludeAllTagsOfInstance": true  
  }  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスの[スケジュールされたイベント](#)」を参照してください。

例 2: イベント通知から特定のタグを削除するには

次のderegister-instance-event-notification-attributes例では、イベント通知に含まれるタグから指定されたタグを削除します。イベント通知に含まれる残りのタグを記述するには、を使用しますdescribe-instance-event-notification-attributes。

```
aws ec2 deregister-instance-event-notification-attributes \  
  --instance-tag-attribute InstanceTagKeys="tag-key2"
```

出力:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [  
      "tag-key2"  
    ],  
    "IncludeAllTagsOfInstance": false  
  }  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスの[スケジュールされたイベント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterInstanceEventNotificationAttributes](#)」の「」を参照してください。

deregister-transit-gateway-multicast-group-members

次の例は、deregister-transit-gateway-multicast-group-members を使用する方法を説明しています。

AWS CLI

マルチキャストグループからグループメンバーの登録を解除するには

この例では、Transit Gateway マルチキャストグループから指定されたネットワークインターフェイスグループメンバーを登録解除します。

```
aws ec2 deregister-transit-gateway-multicast-group-members \  
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE \  
  --group-ip-address 224.0.1.0 \  
  --network-interface-ids eni-0e246d3269EXAMPLE
```

出力:

```
{
  "DeregisteredMulticastGroupMembers": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
    "RegisteredNetworkInterfaceIds": [
      "eni-0e246d3269EXAMPLE"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

詳細については、「Transit Gateways ユーザーガイド」の [「マルチキャストグループからメンバーを登録する」](#) を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterTransitGatewayMulticastGroupMembers](#)」の「」を参照してください。

deregister-transit-gateway-multicast-group-source

次の例は、`deregister-transit-gateway-multicast-group-source` を使用する方法を説明しています。

AWS CLI

Transit Gateway マルチキャストグループからソースの登録を解除するには

この例では、マルチキャストグループから指定されたネットワークインターフェイスグループソースを登録解除します。

```
aws ec2 register-transit-gateway-multicast-group-sources \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \
  --group-ip-address 224.0.1.0 \
  --network-interface-ids eni-07f290fc3c090cbae
```

出力:

```
{
  "DeregisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "DeregisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ]
  }
}
```

```
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

詳細については、「Transit [Gateways ユーザーガイド](#)」の「[マルチキャストグループからのソースの登録解除](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスDeregisterTransitGatewayMulticastGroupSource](#)」の「」を参照してください。

describe-account-attributes

次のコード例は、describe-account-attributes を使用する方法を示しています。

AWS CLI

AWS アカウントのすべての属性を記述するには

この例では、AWS アカウントの属性について説明します。

コマンド:

```
aws ec2 describe-account-attributes
```

出力:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "vpc-max-security-groups-per-interface",
      "AttributeValues": [
        {
          "AttributeValue": "5"
        }
      ]
    },
    {
      "AttributeName": "max-instances",
      "AttributeValues": [
        {
          "AttributeValue": "20"
        }
      ]
    }
  ]
}
```

```
    }
  ]
},
{
  "AttributeName": "supported-platforms",
  "AttributeValues": [
    {
      "AttributeValue": "EC2"
    },
    {
      "AttributeValue": "VPC"
    }
  ]
},
{
  "AttributeName": "default-vpc",
  "AttributeValues": [
    {
      "AttributeValue": "none"
    }
  ]
},
{
  "AttributeName": "max-elastic-ips",
  "AttributeValues": [
    {
      "AttributeValue": "5"
    }
  ]
},
{
  "AttributeName": "vpc-max-elastic-ips",
  "AttributeValues": [
    {
      "AttributeValue": "5"
    }
  ]
}
]
```

AWS アカウントの単一の属性を記述するには

この例では、AWS アカウントの `supported-platforms` 属性について説明します。

コマンド:

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

出力:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "supported-platforms",
      "AttributeValues": [
        {
          "AttributeValue": "EC2"
        },
        {
          "AttributeValue": "VPC"
        }
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountAttributes](#)」の「」を参照してください。

describe-address-transfers

次のコード例は、describe-address-transfers を使用する方法を示しています。

AWS CLI

Elastic IP アドレス転送を記述するには

次のdescribe-address-transfers例では、指定された Elastic IP アドレスの Elastic IP アドレス転送について説明します。

```
aws ec2 describe-address-transfers \
  --allocation-ids eipalloc-09ad461b0d03f6aaf
```

出力:

```
{
  "AddressTransfers": [
    {
      "PublicIp": "100.21.184.216",
      "AllocationId": "eipalloc-09ad461b0d03f6aaf",
      "TransferAccountId": "123456789012",
      "TransferOfferExpirationTimestamp": "2023-02-22T22:51:01.000Z",
      "AddressTransferStatus": "pending"
    }
  ]
}
```

詳細については、「Amazon VPCユーザーガイド」の「[Elastic IP アドレスを転送する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAddressTransfers](#)」の「」を参照してください。

describe-addresses-attribute

次の例は、describe-addresses-attribute を使用する方法を説明しています。

AWS CLI

Elastic IP アドレスに関連付けられたドメイン名の属性を表示するには

次のdescribe-addresses-attribute例では、Elastic IP アドレスに関連付けられたドメイン名の属性を返します。

Linux :

```
aws ec2 describe-addresses-attribute \
  --allocation-ids eipalloc-abcdef01234567890 \
  --attribute domain-name
```

Windows :

```
aws ec2 describe-addresses-attribute ^
  --allocation-ids eipalloc-abcdef01234567890 ^
  --attribute domain-name
```

出力:

```
{
  "Addresses": [
    {
      "PublicIp": "192.0.2.0",
      "AllocationId": "eipalloc-abcdef01234567890",
      "PtrRecord": "example.com."
    }
  ]
}
```

Elastic IP アドレスの属性を表示するには、まずドメイン名をElastic IP アドレスに関連付ける必要があります。詳細については、「Amazon EC2ユーザーガイド」またはAWS CLI「コマンドリファレンス[modify-address-attribute](#)」の「[E メールアプリケーションDNSにリバースを使用する](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeAddressesAttribute](#)」の「」を参照してください。

describe-addresses

次の例は、describe-addresses を使用する方法を説明しています。

AWS CLI

例 1: すべての Elastic IP アドレスに関する詳細を取得するには

次の describe addresses の例では、Elastic IP アドレスに関する詳細が表示されます。

```
aws ec2 describe-addresses
```

出力:

```
{
  "Addresses": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "PublicIp": "198.51.100.0",
      "PublicIpv4Pool": "amazon",
      "Domain": "standard"
    },
    {
      "Domain": "vpc",

```

```
    "PublicIpv4Pool": "amazon",
    "InstanceId": "i-1234567890abcdef0",
    "NetworkInterfaceId": "eni-12345678",
    "AssociationId": "eipassoc-12345678",
    "NetworkInterfaceOwnerId": "123456789012",
    "PublicIp": "203.0.113.0",
    "AllocationId": "eipalloc-12345678",
    "PrivateIpAddress": "10.0.1.241"
  }
]
```

例 2: EC2- の Elastic IP アドレスの詳細を取得するにはVPC

次のdescribe-addresses例では、 のインスタンスで使用する Elastic IP アドレスの詳細を表示しますVPC。

```
aws ec2 describe-addresses \
  --filters "Name=domain,Values=vpc"
```

出力:

```
{
  "Addresses": [
    {
      "Domain": "vpc",
      "PublicIpv4Pool": "amazon",
      "InstanceId": "i-1234567890abcdef0",
      "NetworkInterfaceId": "eni-12345678",
      "AssociationId": "eipassoc-12345678",
      "NetworkInterfaceOwnerId": "123456789012",
      "PublicIp": "203.0.113.0",
      "AllocationId": "eipalloc-12345678",
      "PrivateIpAddress": "10.0.1.241"
    }
  ]
}
```

例 3: 割り当て ID で指定された Elastic IP アドレスに関する詳細を取得するには

次のdescribe-addresses例では、EC2- のインスタンスに関連付けられている指定された割り当て ID を持つ Elastic IP アドレスの詳細を表示しますVPC。


```
aws ec2 describe-addresses \  
  --allocation-ids eipalloc-282d9641
```

出力:

```
{  
  "Addresses": [  
    {  
      "Domain": "vpc",  
      "PublicIpv4Pool": "amazon",  
      "InstanceId": "i-1234567890abcdef0",  
      "NetworkInterfaceId": "eni-1a2b3c4d",  
      "AssociationId": "eipassoc-123abc12",  
      "NetworkInterfaceOwnerId": "1234567891012",  
      "PublicIp": "203.0.113.25",  
      "AllocationId": "eipalloc-282d9641",  
      "PrivateIpAddress": "10.251.50.12"  
    }  
  ]  
}
```

例 4: VPCプライベート IP アドレスで指定された Elastic IP アドレスの詳細を取得するには

次のdescribe-addresses例では、EC2- の特定のプライベート IP アドレスに関連付けられた Elastic IP アドレスの詳細を表示しますVPC。

```
aws ec2 describe-addresses \  
  --filters "Name=private-ip-address,Values=10.251.50.12"
```

例 5: EC2-Classic の Elastic IP アドレスの詳細を取得するには

The 次のdescribe-addresses例では、EC2-Classic で使用する Elastic IP アドレスの詳細を表示します。

```
aws ec2 describe-addresses \  
  --filters "Name=domain,Values=standard"
```

出力:

```
{
```

```
"Addresses": [  
  {  
    "InstanceId": "i-1234567890abcdef0",  
    "PublicIp": "203.0.110.25",  
    "PublicIpv4Pool": "amazon",  
    "Domain": "standard"  
  }  
]  
}
```

例 6: パブリック IP アドレスで指定された Elastic IP アドレスに関する詳細を取得するには

次のdescribe-addresses例では203.0.110.25、EC2-Classic のインスタンスに関連付けられている値を持つ Elastic IP アドレスの詳細を表示します。

```
aws ec2 describe-addresses \  
  --public-ips 203.0.110.25
```

出力:

```
{  
  "Addresses": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "PublicIp": "203.0.110.25",  
      "PublicIpv4Pool": "amazon",  
      "Domain": "standard"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAddresses](#)」の「」を参照してください。

describe-aggregate-id-format

次のコード例は、describe-aggregate-id-format を使用する方法を示しています。

AWS CLI

リージョン内のすべてのリソースタイプの長い ID 形式設定を記述するには

次のdescribe-aggregate-id-format例では、現在のリージョンの全体的なロング ID 形式ステータスについて説明します。Deadline 値は、これらのリソースがショート ID 形式からロング ID 形式に完全に切り替わる期限が期限切れであることを示します。このUseLongIdsAggregated値は、すべてのIAMユーザーとIAMロールがすべてのリソースタイプに長い ID 形式を使用するように設定されていることを示します。

```
aws ec2 describe-aggregate-id-format
```

出力:

```
{
  "UseLongIdsAggregated": true,
  "Statuses": [
    {
      "Deadline": "2018-08-13T02:00:00.000Z",
      "Resource": "network-interface-attachment",
      "UseLongIds": true
    },
    {
      "Deadline": "2016-12-13T02:00:00.000Z",
      "Resource": "instance",
      "UseLongIds": true
    },
    {
      "Deadline": "2018-08-13T02:00:00.000Z",
      "Resource": "elastic-ip-association",
      "UseLongIds": true
    },
    ...
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAggregateIdFormat](#)」の「」を参照してください。

describe-availability-zones

次の例は、describe-availability-zones を使用する方法を説明しています。

AWS CLI

アベイラビリティゾーンを説明するには

次の `describe-availability-zones` の例では、利用可能なアベイラビリティゾーンの詳細が表示されます。レスポンスには、現在のリージョンのアベイラビリティゾーンのみが含まれます。この例では、デフォルトの `us-west-2` (オレゴン) リージョンのプロファイルを使用しています。

```
aws ec2 describe-availability-zones
```

出力:

```
{
  "AvailabilityZones": [
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2a",
      "ZoneId": "usw2-az1",
      "GroupName": "us-west-2",
      "NetworkBorderGroup": "us-west-2"
    },
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2b",
      "ZoneId": "usw2-az2",
      "GroupName": "us-west-2",
      "NetworkBorderGroup": "us-west-2"
    },
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2c",
      "ZoneId": "usw2-az3",
```

```
    "GroupName": "us-west-2",
    "NetworkBorderGroup": "us-west-2"
  },
  {
    "State": "available",
    "OptInStatus": "opt-in-not-required",
    "Messages": [],
    "RegionName": "us-west-2",
    "ZoneName": "us-west-2d",
    "ZoneId": "usw2-az4",
    "GroupName": "us-west-2",
    "NetworkBorderGroup": "us-west-2"
  },
  {
    "State": "available",
    "OptInStatus": "opted-in",
    "Messages": [],
    "RegionName": "us-west-2",
    "ZoneName": "us-west-2-lax-1a",
    "ZoneId": "usw2-lax1-az1",
    "GroupName": "us-west-2-lax-1",
    "NetworkBorderGroup": "us-west-2-lax-1"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAvailabilityZones](#)」の「」を参照してください。

describe-aws-network-performance-metric-subscription

次のコード例は、describe-aws-network-performance-metric-subscription を使用する
方法を示しています。

AWS CLI

メトリクスサブスクリプションを記述するには

次のdescribe-aws-network-performance-metric-subscriptions例では、メトリクス
サブスクリプションについて説明します。

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

出力:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes"
    }
  ]
}
```

詳細については、「Infrastructure Performance ユーザーガイド」の[「サブスクリプションの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAwsNetworkPerformanceMetricSubscription](#)」の「」を参照してください。

describe-aws-network-performance-metric-subscriptions

次の例は、describe-aws-network-performance-metric-subscriptions を使用する方法を説明しています。

AWS CLI

メトリクスサブスクリプションを記述するには

次のdescribe-aws-network-performance-metric-subscriptions例では、メトリクスサブスクリプションについて説明します。

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

出力:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
```

```
        "Metric": "aggregate-latency",
        "Statistic": "p50",
        "Period": "five-minutes"
    }
]
}
```

詳細については、「Infrastructure Performance ユーザーガイド」の[「サブスクリプションの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAwsNetworkPerformanceMetricSubscriptions](#)」の「」を参照してください。

describe-bundle-tasks

次の例は、describe-bundle-tasks を使用する方法を説明しています。

AWS CLI

バンドルタスクを記述するには

この例では、すべてのバンドルタスクについて説明します。

コマンド:

```
aws ec2 describe-bundle-tasks
```

出力:

```
{
  "BundleTasks": [
    {
      "UpdateTime": "2015-09-15T13:26:54.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "Storage": {
        "S3": {
          "Prefix": "winami",
          "Bucket": "bundletasks"
        }
      },
      "State": "bundling",
      "StartTime": "2015-09-15T13:24:35.000Z",
```

```
    "Progress": "3%",
    "BundleId": "bun-2a4e041c"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeBundleTasks](#)」の「」を参照してください。

describe-byoip-cidrs

次のコード例は、describe-byoip-cidrs を使用する方法を示しています。

AWS CLI

プロビジョニングされたアドレス範囲を記述するには

次のdescribe-byoip-cidrs例では、が使用するためにプロビジョニングしたパブリックIPv4アドレス範囲の詳細を表示します AWS。

```
aws ec2 describe-byoip-cidrs
```

出力:

```
{
  "ByoipCidrs": [
    {
      "Cidr": "203.0.113.25/24",
      "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
      "State": "provisioned"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeByoipCidrs](#)」の「」を参照してください。

describe-capacity-reservation-fleets

次の例は、describe-capacity-reservation-fleets を使用する方法を説明しています。

AWS CLI

キャパシティ予約フリートを表示するには

次のdescribe-capacity-reservation-fleets例では、指定されたキャパシティ予約フリートの設定とキャパシティ情報を一覧表示します。また、フリート内の個々のキャパシティ予約の詳細も一覧表示されます。

```
aws ec2 describe-capacity-reservation-fleets \
  --capacity-reservation-fleet-ids crf-abcdef01234567890
```

出力:

```
{
  "CapacityReservationFleets": [
    {
      "Status": "active",
      "EndDate": "2022-12-31T23:59:59.000Z",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "CapacityReservationFleetId": "crf-abcdef01234567890",
      "Tenancy": "default",
      "InstanceTypeSpecifications": [
        {
          "CapacityReservationId": "cr-1234567890abcdef0",
          "AvailabilityZone": "us-east-1a",
          "FulfilledCapacity": 5.0,
          "Weight": 1.0,
          "CreateDate": "2022-07-02T08:34:33.398Z",
          "InstancePlatform": "Linux/UNIX",
          "TotalInstanceCount": 5,
          "Priority": 1,
          "EbsOptimized": true,
          "InstanceType": "m5.xlarge"
        }
      ],
      "TotalTargetCapacity": 5,
      "TotalFulfilledCapacity": 5.0,
      "CreateTime": "2022-07-02T08:34:33.397Z",
      "AllocationStrategy": "prioritized"
    }
  ]
}
```

```
}
```

キャパシティ予約フリートの詳細については、Amazon EC2ユーザーガイドの「[キャパシティ予約フリート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCapacityReservationFleets](#)」の「」を参照してください。

describe-capacity-reservations

次の例は、describe-capacity-reservations を使用する方法を説明しています。

AWS CLI

例 1: 1 つ以上のキャパシティ予約を記述するには

次のdescribe-capacity-reservations例では、現在の AWS リージョンのすべてのキャパシティ予約の詳細を表示します。

```
aws ec2 describe-capacity-reservations
```

出力:

```
{
  "CapacityReservations": [
    {
      "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
      "EndDateType": "unlimited",
      "AvailabilityZone": "eu-west-1a",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "EphemeralStorage": false,
      "CreateDate": "2019-08-16T09:03:18.000Z",
      "AvailableInstanceCount": 1,
      "InstancePlatform": "Linux/UNIX",
      "TotalInstanceCount": 1,
      "State": "active",
      "Tenancy": "default",
      "EbsOptimized": true,
      "InstanceType": "a1.medium"
    },
    {
```

```

    "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
    "EndDateType": "unlimited",
    "AvailabilityZone": "eu-west-1a",
    "InstanceMatchCriteria": "open",
    "Tags": [],
    "EphemeralStorage": false,
    "CreateDate": "2019-08-07T11:34:19.000Z",
    "AvailableInstanceCount": 3,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 3,
    "State": "cancelled",
    "Tenancy": "default",
    "EbsOptimized": true,
    "InstanceType": "m5.large"
  }
]
}

```

例 2: 1 つ以上のキャパシティ予約を記述するには

次のdescribe-capacity-reservations例では、指定されたキャパシティ予約の詳細を表示します。

```

aws ec2 describe-capacity-reservations \
  --capacity-reservation-ids cr-1234abcd56EXAMPLE

```

出力:

```

{
  "CapacityReservations": [
    {
      "CapacityReservationId": "cr-1234abcd56EXAMPLE",
      "EndDateType": "unlimited",
      "AvailabilityZone": "eu-west-1a",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "EphemeralStorage": false,
      "CreateDate": "2019-08-16T09:03:18.000Z",
      "AvailableInstanceCount": 1,
      "InstancePlatform": "Linux/UNIX",
      "TotalInstanceCount": 1,
      "State": "active",
      "Tenancy": "default",
    }
  ]
}

```

```
        "EbsOptimized": true,  
        "InstanceType": "a1.medium"  
    }  
]  
}
```

詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「キャパシティ予約の表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCapacityReservations](#)」の「」を参照してください。

describe-carrier-gateways

次の例は、describe-carrier-gateways を使用する方法を説明しています。

AWS CLI

すべてのキャリアゲートウェイを記述するには

次のdescribe-carrier-gateways例では、すべてのキャリアゲートウェイを一覧表示します。

```
aws ec2 describe-carrier-gateways
```

出力:

```
{  
  "CarrierGateways": [  
    {  
      "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",  
      "VpcId": "vpc-0c529aEXAMPLE",  
      "State": "available",  
      "OwnerId": "123456789012",  
      "Tags": [  
        {  
          "Key": "example",  
          "Value": "tag"  
        }  
      ]  
    }  
  ]  
}
```

```
]
}
```

詳細については、「Amazon Virtual Private Cloud ユーザーガイド」の「キャリアゲートウェイ <https://docs.aws.amazon.com/vpc/latest/userguide/Carrier_Gateway.html>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeCarrierGateways](#)」の「」を参照してください。

describe-classic-link-instances

次の例は、describe-classic-link-instances を使用する方法を説明しています。

AWS CLI

リンクされた EC2-Classic インスタンスを記述するには

この例では、リンクされた EC2-Classic インスタンスをすべて一覧表示します。

コマンド:

```
aws ec2 describe-classic-link-instances
```

出力:

```
{
  "Instances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "VpcId": "vpc-88888888",
      "Groups": [
        {
          "GroupId": "sg-11122233"
        }
      ],
      "Tags": [
        {
          "Value": "ClassicInstance",
          "Key": "Name"
        }
      ]
    }
  ],
}
```

```
{
  "InstanceId": "i-0598c7d356eba48d7",
  "VpcId": "vpc-12312312",
  "Groups": [
    {
      "GroupId": "sg-aabbccdd"
    }
  ],
  "Tags": [
    {
      "Value": "ClassicInstance2",
      "Key": "Name"
    }
  ]
}
]
```

この例では、リンクされた EC2-Classic インスタンスをすべて一覧表示し、vpc-88888888 VPC にリンクされたインスタンスのみを含めるようにレスポンスをフィルタリングします。

コマンド:

```
aws ec2 describe-classic-link-instances --filter "Name=vpc-id,Values=vpc-88888888"
```

出力:

```
{
  "Instances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "VpcId": "vpc-88888888",
      "Groups": [
        {
          "GroupId": "sg-11122233"
        }
      ],
      "Tags": [
        {
          "Value": "ClassicInstance",
          "Key": "Name"
        }
      ]
    }
  ]
}
```

```

    ]
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeClassicLinkInstances](#)」の「」を参照してください。

describe-client-vpn-authorization-rules

次の例は、describe-client-vpn-authorization-rules を使用する方法を説明しています。

AWS CLI

クライアントVPNエンドポイントの承認ルールを記述するには

次のdescribe-client-vpn-authorization-rules例では、指定されたクライアントVPNエンドポイントの承認ルールの詳細を表示します。

```

aws ec2 describe-client-vpn-authorization-rules \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

出力:

```

{
  "AuthorizationRules": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "GroupId": "",
      "AccessAll": true,
      "DestinationCidr": "0.0.0.0/0",
      "Status": {
        "Code": "active"
      }
    }
  ]
}

```

詳細については、「クライアント管理者ガイド」の「[認証ルール](#)」を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeClientVpnAuthorizationRules](#)」の「」を参照してください。

describe-client-vpn-connections

次のコード例は、describe-client-vpn-connections を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントへの接続を記述するには

次のdescribe-client-vpn-connections例では、指定されたクライアントVPNエンドポイントへのクライアント接続の詳細を表示します。

```
aws ec2 describe-client-vpn-connections \  
--client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

出力:

```
{  
  "Connections": [  
    {  
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",  
      "Timestamp": "2019-08-12 07:58:34",  
      "ConnectionId": "cvpn-connection-0e03eb24267165acd",  
      "ConnectionEstablishedTime": "2019-08-12 07:57:14",  
      "IngressBytes": "32302",  
      "EgressBytes": "5696",  
      "IngressPackets": "332",  
      "EgressPackets": "67",  
      "ClientIp": "172.31.0.225",  
      "CommonName": "client1.domain.tld",  
      "Status": {  
        "Code": "terminated"  
      },  
      "ConnectionEndTime": "2019-08-12 07:58:34"  
    },  
    {  
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",  
      "Timestamp": "2019-08-12 08:02:54",  
      "ConnectionId": "cvpn-connection-00668867a40f18253",  
      "ConnectionEstablishedTime": "2019-08-12 08:02:53",  
      "IngressBytes": "2951",  
      "EgressBytes": "2611",  
      "IngressPackets": "9",  
      "EgressPackets": "6",  
    }  
  ]  
}
```



```
    "ClientId": "172.31.0.226",
    "CommonName": "client1.domain.tld",
    "Status": {
      "Code": "active"
    },
    "ConnectionEndTime": "-"
  }
]
```

詳細については、[「クライアント管理者ガイド」の「クライアント接続AWS VPN」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClientVpnConnections](#)」の「」を参照してください。

describe-client-vpn-endpoints

次のコード例は、describe-client-vpn-endpoints を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントを記述するには

次のdescribe-client-vpn-endpoints例では、すべてのクライアントVPNエンドポイントの詳細を表示します。

```
aws ec2 describe-client-vpn-endpoints
```

出力:

```
{
  "ClientVpnEndpoints": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "Description": "Endpoint for Admin access",
      "Status": {
        "Code": "available"
      },
      "CreationTime": "2020-11-13T11:37:27",
      "DnsName": "*.cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com",
      "ClientCidrBlock": "172.31.0.0/16",
```

```
    "DnsServers": [
      "8.8.8.8"
    ],
    "SplitTunnel": false,
    "VpnProtocol": "openvpn",
    "TransportProtocol": "udp",
    "VpnPort": 443,
    "ServerCertificateArn": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "AuthenticationOptions": [
      {
        "Type": "certificate-authentication",
        "MutualAuthentication": {
          "ClientRootCertificateChain": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
        }
      }
    ],
    "ConnectionLogOptions": {
      "Enabled": true,
      "CloudwatchLogGroup": "Client-vpn-connection-logs",
      "CloudwatchLogStream": "cvpn-endpoint-123456789123abcde-ap-
south-1-2020/11/13-FCD8HEMVAcCw"
    },
    "Tags": [
      {
        "Key": "Name",
        "Value": "Client VPN"
      }
    ],
    "SecurityGroupIds": [
      "sg-aabbcc11223344567"
    ],
    "VpcId": "vpc-a87f92c1",
    "SelfServicePortalUrl": "https://self-service.clientvpn.amazonaws.com/
endpoints/cvpn-endpoint-123456789123abcde",
    "ClientConnectOptions": {
      "Enabled": false
    }
  }
]
```

詳細については、[「クライアント管理者ガイド」の「クライアントVPNエンドポイント」](#)を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClientVpnEndpoints](#)」の「」を参照してください。

describe-client-vpn-routes

次のコード例は、describe-client-vpn-routes を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントのルートを記述するには

次のdescribe-client-vpn-routes例では、指定されたクライアントVPNエンドポイントのルートの詳細を表示します。

```
aws ec2 describe-client-vpn-routes \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

出力:

```
{  
  "Routes": [  
    {  
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",  
      "DestinationCidr": "10.0.0.0/16",  
      "TargetSubnet": "subnet-0123456789abcabca",  
      "Type": "Nat",  
      "Origin": "associate",  
      "Status": {  
        "Code": "active"  
      },  
      "Description": "Default Route"  
    },  
    {  
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",  
      "DestinationCidr": "0.0.0.0/0",  
      "TargetSubnet": "subnet-0123456789abcabca",  
      "Type": "Nat",  
      "Origin": "add-route",  
      "Status": {
```

```
        "Code": "active"
      }
    ]
  }
}
```

詳細については、AWS クライアントVPN管理者ガイドの「[ルート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClientVpnRoutes](#)」の「」を参照してください。

describe-client-vpn-target-networks

次のコード例は、describe-client-vpn-target-networks を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントのターゲットネットワークを記述するには

次のdescribe-client-vpn-target-networks例では、指定されたクライアントVPNエンドポイントのターゲットネットワークの詳細を表示します。

```
aws ec2 describe-client-vpn-target-networks \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

出力:

```
{
  "ClientVpnTargetNetworks": [
    {
      "AssociationId": "cvpn-assoc-012e837060753dc3d",
      "VpcId": "vpc-11111222222333333",
      "TargetNetworkId": "subnet-0123456789abcabca",
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "Status": {
        "Code": "associating"
      },
      "SecurityGroups": [
        "sg-012345678910abcab"
      ]
    }
  ]
}
```

```
]
}
```

詳細については、クライアント管理者ガイドの「[ターゲットネットワーク](#)」を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClientVpnTargetNetworks](#)」の「」を参照してください。

describe-coip-pools

次のコード例は、describe-coip-pools を使用する方法を示しています。

AWS CLI

カスタマー所有の IP アドレスプールを記述するには

次のdescribe-coip-pools例では、AWS アカウント内のカスタマー所有の IP アドレスプールについて説明します。

```
aws ec2 describe-coip-pools
```

出力:

```
{
  "CoipPools": [
    {
      "PoolId": "ipv4pool-coip-123a45678bEXAMPLE",
      "PoolCidrs": [
        "0.0.0.0/0"
      ],
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
      "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-coip-123a45678bEXAMPLE"
    }
  ]
}
```

CoIP の詳細については、AWS Outposts ユーザーガイドの「[カスタマー所有の IP アドレス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCoipPools](#)」の「」を参照してください。

describe-conversion-tasks

次のコード例は、describe-conversion-tasks を使用する方法を示しています。

AWS CLI

変換タスクのステータスを表示するには

この例では、ID import-i-ffvko9js の変換タスクのステータスを返します。

コマンド:

```
aws ec2 describe-conversion-tasks --conversion-task-ids import-i-ffvko9js
```

出力:

```
{
  "ConversionTasks": [
    {
      "ConversionTaskId": "import-i-ffvko9js",
      "ImportInstance": {
        "InstanceId": "i-1234567890abcdef0",
        "Volumes": [
          {
            "Volume": {
              "Id": "vol-049df61146c4d7901",
              "Size": 16
            },
            "Status": "completed",
            "Image": {
              "Size": 1300687360,
              "ImportManifestUrl": "https://s3.amazonaws.com/myimportbucket/411443cd-d620-4f1c-9d66-13144EXAMPLE/RHEL5.vmdkmanifest.xml?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=140EXAMPLE&Signature=XYNhznHNgcQsjDxL9wRL%2FJvEXAMPLE",
              "Format": "VMDK"
            },
            "BytesConverted": 1300682960,
            "AvailabilityZone": "us-east-1d"
          }
        ]
      }
    }
  ]
}
```

```
        }
      ]
    },
    "ExpirationTime": "2014-05-14T22:06:23Z",
    "State": "completed"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeConversionTasks](#)」の「」を参照してください。

describe-customer-gateways

次のコード例は、describe-customer-gateways を使用する方法を示しています。

AWS CLI

カスタマーゲートウェイを記述するには

この例では、カスタマーゲートウェイについて説明します。

コマンド:

```
aws ec2 describe-customer-gateways
```

出力:

```
{
  "CustomerGateways": [
    {
      "CustomerGatewayId": "cgw-b4dc3961",
      "IpAddress": "203.0.113.12",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65000"
    },
    {
      "CustomerGatewayId": "cgw-0e11f167",
      "IpAddress": "12.1.2.3",
      "State": "available",
      "Type": "ipsec.1",

```

```
        "BgpAsn": "65534"
      }
    ]
  }
```

特定のカスタマーゲートウェイを記述するには

この例では、指定されたカスタマーゲートウェイについて説明します。

コマンド:

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

出力:

```
{
  "CustomerGateways": [
    {
      "CustomerGatewayId": "cgw-0e11f167",
      "IpAddress": "12.1.2.3",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65534"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCustomerGateways](#)」の「」を参照してください。

describe-dhcp-options

次の例は、describe-dhcp-options を使用する方法を説明しています。

AWS CLI

例 1: DHCPオプションを記述するには

次のdescribe-dhcp-options例では、DHCPオプションの詳細を取得します。

```
aws ec2 describe-dhcp-options
```


出力:

```
{
  "DhcpOptions": [
    {
      "DhcpConfigurations": [
        {
          "Key": "domain-name",
          "Values": [
            {
              "Value": "us-east-2.compute.internal"
            }
          ]
        },
        {
          "Key": "domain-name-servers",
          "Values": [
            {
              "Value": "AmazonProvidedDNS"
            }
          ]
        }
      ],
      "DhcpOptionsId": "dopt-19edf471",
      "OwnerId": "111122223333"
    },
    {
      "DhcpConfigurations": [
        {
          "Key": "domain-name",
          "Values": [
            {
              "Value": "us-east-2.compute.internal"
            }
          ]
        },
        {
          "Key": "domain-name-servers",
          "Values": [
            {
              "Value": "AmazonProvidedDNS"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    ],
    "DhcpOptionsId": "dopt-fEXAMPLE",
    "OwnerId": "111122223333"
  }
]
}

```

詳細については、AWS VPC「[ユーザーガイド](#)」のDHCP「[オプションセットの使用](#)」を参照してください。

例 2: DHCPオプションを記述し、出力をフィルタリングするには

次のdescribe-dhcp-options例では、DHCPオプションについて説明し、フィルターを使用して、ドメインネームサーバーexample.comにがあるDHCPオプションのみを返します。この例では、--queryパラメータを使用して、出力の設定情報と ID のみを表示します。

```

aws ec2 describe-dhcp-options \
  --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
  --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"

```

出力:

```

[
  [
    [
      {
        "Key": "domain-name",
        "Values": [
          {
            "Value": "example.com"
          }
        ]
      },
      {
        "Key": "domain-name-servers",
        "Values": [
          {
            "Value": "172.16.16.16"
          }
        ]
      }
    ]
  ],
  "dopt-001122334455667ab"

```

```
]
]
```

詳細については、AWS VPC 「ユーザーガイド」の[DHCP 「オプションセットの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDhcpOptions](#)」の「」を参照してください。

describe-egress-only-internet-gateways

次のコード例は、`describe-egress-only-internet-gateways` を使用する方法を示しています。

AWS CLI

Egress-Only インターネットゲートウェイを記述するには

この例では、エグレス専用インターネットゲートウェイについて説明します。

コマンド:

```
aws ec2 describe-egress-only-internet-gateways
```

出力:

```
{
  "EgressOnlyInternetGateways": [
    {
      "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
      "Attachments": [
        {
          "State": "attached",
          "VpcId": "vpc-0c62a468"
        }
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEgressOnlyInternetGateways](#)」の「」を参照してください。

describe-elastic-gpus

次のコード例は、describe-elastic-gpus を使用する方法を示しています。

AWS CLI

Elastic を記述するには GPU

コマンド:

```
aws ec2 describe-elastic-gpus --elastic-gpu-ids egpu-12345678901234567890abcdefghijkl
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeElasticGpus](#)」の「」を参照してください。

describe-export-image-tasks

次の例は、describe-export-image-tasks を使用する方法を説明しています。

AWS CLI

エクスポートイメージタスクをモニタリングするには

次のdescribe-export-image-tasks例では、指定されたエクスポートイメージタスクのステータスをチェックします。Amazon S3 の結果のイメージファイルは `my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk` です。

```
aws ec2 describe-export-image-tasks \
  --export-image-task-ids export-ami-1234567890abcdef0
```

進行中のエクスポートイメージタスクの出力。

```
{
  "ExportImageTasks": [
    {
      "ExportImageTaskId": "export-ami-1234567890abcdef0"
      "Progress": "21",
      "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
      }
    }
  ]
}
```

```
    },
    "Status": "active",
    "StatusMessage": "updating"
  }
]
}
```

完了したエクスポートイメージタスクの出力。

```
{
  "ExportImageTasks": [
    {
      "ExportImageTaskId": "export-ami-1234567890abcdef0"
      "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
      },
      "Status": "completed"
    }
  ]
}
```

詳細については、[VM Import/Export ユーザーガイドの「から AMI VM をエクスポートする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeExportImageTasks](#)」の「」を参照してください。

describe-export-tasks

次のコード例は、describe-export-tasks を使用する方法を示しています。

AWS CLI

インスタンスエクスポートタスクの詳細を一覧表示するには

この例では、ID export-i-fh8sjjsq のエクスポートタスクについて説明します。

コマンド:

```
aws ec2 describe-export-tasks --export-task-ids export-i-fh8sjjsq
```

出力:

```
{
  "ExportTasks": [
    {
      "State": "active",
      "InstanceExportDetails": {
        "InstanceId": "i-1234567890abcdef0",
        "TargetEnvironment": "vmware"
      },
      "ExportToS3Task": {
        "S3Bucket": "myexportbucket",
        "S3Key": "RHEL5export-i-fh8sjjsq.ova",
        "DiskImageFormat": "vmdk",
        "ContainerFormat": "ova"
      },
      "Description": "RHEL5 instance",
      "ExportTaskId": "export-i-fh8sjjsq"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeExportTasks](#)」の「」を参照してください。

describe-fast-launch-images

次のコード例は、describe-fast-launch-images を使用する方法を示しています。

AWS CLI

高速起動用に AMIs 設定された Windows の詳細を記述するには

次の describe-fast-launch-images 例では、リソースタイプ、スナップショット設定、起動テンプレートの詳細、並列起動の最大数、AMI 所有者 ID、高速起動設定の状態、状態が変更された理由、状態変更が発生した時間など、起動を高速化するように設定された AMIs アカウント内の各の詳細について説明します。

```
aws ec2 describe-fast-launch-images
```

出力:

```
{
  "FastLaunchImages": [
    {
      "ImageId": "ami-01234567890abcdef",
      "ResourceType": "snapshot",
      "SnapshotConfiguration": {},
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcdef",
        "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-
a8c6215d-94e6-441b-9272-dbd1f87b07e2",
        "Version": "1"
      },
      "MaxParallelLaunches": 6,
      "OwnerId": "0123456789123",
      "State": "enabled",
      "StateTransitionReason": "Client.UserInitiated",
      "StateTransitionTime": "2022-01-27T22:20:06.552000+00:00"
    }
  ]
}
```

起動を高速化AMIするための Windows の設定の詳細については、「Amazon EC2ユーザーガイド」の「[起動を高速化AMIするための の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFastLaunchImages](#)」の「」を参照してください。

describe-fast-snapshot-restores

次の例は、describe-fast-snapshot-restores を使用する方法を説明しています。

AWS CLI

高速スナップショット復元を記述するには

次のdescribe-fast-snapshot-restores例では、 の状態のすべての高速スナップショット復元の詳細を表示しますdisabled。

```
aws ec2 describe-fast-snapshot-restores \
  --filters Name=state,Values=disabled
```

出力:

```
{
  "FastSnapshotRestores": [
    {
      "SnapshotId": "snap-1234567890abcdef0",
      "AvailabilityZone": "us-west-2c",
      "State": "disabled",
      "StateTransitionReason": "Client.UserInitiated - Lifecycle state
transition",
      "OwnerId": "123456789012",
      "EnablingTime": "2020-01-25T23:57:49.596Z",
      "OptimizingTime": "2020-01-25T23:58:25.573Z",
      "EnabledTime": "2020-01-25T23:59:29.852Z",
      "DisablingTime": "2020-01-26T00:40:56.069Z",
      "DisabledTime": "2020-01-26T00:41:27.390Z"
    }
  ]
}
```

次のdescribe-fast-snapshot-restores例では、すべての高速スナップショット復元について説明します。

```
aws ec2 describe-fast-snapshot-restores
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFastSnapshotRestores](#)」の「」を参照してください。

describe-fleet-history

次のコード例は、describe-fleet-history を使用する方法を示しています。

AWS CLI

EC2フリート履歴を記述するには

次のdescribe-fleet-history例では、指定された時刻から始まる、指定されたEC2フリートの履歴を返します。出力は、2つの実行中のインスタンスを持つEC2フリート用です。

```
aws ec2 describe-fleet-history \
  --fleet-id fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
  --start-time 2020-09-01T00:00:00Z
```


出力:

```
{
  "HistoryRecords": [
    {
      "EventInformation": {
        "EventSubType": "submitted"
      },
      "EventType": "fleetRequestChange",
      "Timestamp": "2020-09-01T18:26:05.000Z"
    },
    {
      "EventInformation": {
        "EventSubType": "active"
      },
      "EventType": "fleetRequestChange",
      "Timestamp": "2020-09-01T18:26:15.000Z"
    },
    {
      "EventInformation": {
        "EventDescription": "t2.small, ami-07c8bc5c1ce9598c3, ...",
        "EventSubType": "progress"
      },
      "EventType": "fleetRequestChange",
      "Timestamp": "2020-09-01T18:26:17.000Z"
    },
    {
      "EventInformation": {
        "EventDescription": "{\"instanceType\": \"t2.small\", ...}",
        "EventSubType": "launched",
        "InstanceId": "i-083a1c446e66085d2"
      },
      "EventType": "instanceChange",
      "Timestamp": "2020-09-01T18:26:17.000Z"
    },
    {
      "EventInformation": {
        "EventDescription": "{\"instanceType\": \"t2.small\", ...}",
        "EventSubType": "launched",
        "InstanceId": "i-090db02406cc3c2d6"
      },
      "EventType": "instanceChange",
      "Timestamp": "2020-09-01T18:26:17.000Z"
    }
  ]
}
```

```
  ],  
  "LastEvaluatedTime": "2020-09-01T19:10:19.000Z",  
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
  "StartTime": "2020-08-31T23:53:20.000Z"  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の[EC2「フリートの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFleetHistory](#)」の「」を参照してください。

describe-fleet-instances

次の例は、describe-fleet-instances を使用する方法を説明しています。

AWS CLI

EC2フリートの実行中のインスタンスを記述するには

次のdescribe-fleet-instances例では、指定されたEC2フリートの実行インスタンスについて説明します。

```
aws ec2 describe-fleet-instances \  
  --fleet-id 12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

出力:

```
{  
  "ActiveInstances": [  
    {  
      "InstanceId": "i-090db02406cc3c2d6",  
      "InstanceType": "t2.small",  
      "SpotInstanceRequestId": "sir-a43gtpfk",  
      "InstanceHealth": "healthy"  
    },  
    {  
      "InstanceId": "i-083a1c446e66085d2",  
      "InstanceType": "t2.small",  
      "SpotInstanceRequestId": "sir-iwcit2nj",  
      "InstanceHealth": "healthy"  
    }  
  ],  
}
```

```
"FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の [EC2「フリートの管理」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DescribeFleetInstances](#)」の「」を参照してください。

describe-fleets

次のコード例は、describe-fleets を使用する方法を示しています。

AWS CLI

EC2フリートを記述するには

次のdescribe-fleets例では、指定されたEC2フリートについて説明します。

```
aws ec2 describe-fleets \
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

出力:

```
{
  "Fleets": [
    {
      "ActivityStatus": "pending_fulfillment",
      "CreateTime": "2020-09-01T18:26:05.000Z",
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",
      "FleetState": "active",
      "ExcessCapacityTerminationPolicy": "termination",
      "FulfilledCapacity": 0.0,
      "FulfilledOnDemandCapacity": 0.0,
      "LaunchTemplateConfigs": [
        {
          "LaunchTemplateSpecification": {
            "LaunchTemplateId": "lt-0e632f2855a979cd5",
            "Version": "1"
          }
        }
      ],
      "TargetCapacitySpecification": {
```

```
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity": 0,
        "SpotTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    },
    "TerminateInstancesWithExpiration": false,
    "Type": "maintain",
    "ReplaceUnhealthyInstances": false,
    "SpotOptions": {
        "AllocationStrategy": "lowestPrice",
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowestPrice"
    }
}
]
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の [EC2「フリートの管理」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DescribeFleets](#)」の「」を参照してください。

describe-flow-logs

次のコード例は、describe-flow-logs を使用する方法を示しています。

AWS CLI

例 1: すべてのフローログを記述するには

次のdescribe-flow-logs例では、すべてのフローログの詳細を表示します。

```
aws ec2 describe-flow-logs
```

出力:

```
{
  "FlowLogs": [
    {
```

```

    "CreationTime": "2018-02-21T13:22:12.644Z",
    "DeliverLogsPermissionArn": "arn:aws:iam::123456789012:role/flow-logs-
role",
    "DeliverLogsStatus": "SUCCESS",
    "FlowLogId": "fl-aabbccdd112233445",
    "MaxAggregationInterval": 600,
    "FlowLogStatus": "ACTIVE",
    "LogGroupName": "FlowLogGroup",
    "ResourceId": "subnet-12345678901234567",
    "TrafficType": "ALL",
    "LogDestinationType": "cloud-watch-logs",
    "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr}
${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end}
${action} ${log-status}"
  },
  {
    "CreationTime": "2020-02-04T15:22:29.986Z",
    "DeliverLogsStatus": "SUCCESS",
    "FlowLogId": "fl-01234567890123456",
    "MaxAggregationInterval": 60,
    "FlowLogStatus": "ACTIVE",
    "ResourceId": "vpc-00112233445566778",
    "TrafficType": "ACCEPT",
    "LogDestinationType": "s3",
    "LogDestination": "arn:aws:s3:::my-flow-log-bucket/custom",
    "LogFormat": "${version} ${vpc-id} ${subnet-id} ${instance-id}
${interface-id} ${account-id} ${type} ${srcaddr} ${dstaddr} ${srcport} ${dstport}
${pkt-srcaddr} ${pkt-dstaddr} ${protocol} ${bytes} ${packets} ${start} ${end}
${action} ${tcp-flags} ${log-status}"
  }
]
}

```

例 2: フローログのサブセットを記述するには

次のdescribe-flow-logs例では、フィルターを使用して、Amazon CloudWatch Logs の指定されたロググループにあるフローログの詳細のみを表示します。

```

aws ec2 describe-flow-logs \
  --filter "Name=log-group-name,Values=MyFlowLogs"

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeFlowLogs](#)」の「」を参照してください。

describe-fpga-image-attribute

次のコード例は、describe-fpga-image-attribute を使用する方法を示しています。

AWS CLI

Amazon FPGAイメージの属性を記述するには

この例では、指定された のロードアクセス許可について説明しますAFI。

コマンド:

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --  
attribute LoadPermission
```

出力:

```
{  
  "FpgaImageAttribute": {  
    "FpgaImageId": "afi-0d123e123bfc85abc",  
    "LoadPermissions": [  
      {  
        "UserId": "123456789012"  
      }  
    ]  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFpgaImageAttribute](#)」の「」を参照してください。

describe-fpga-images

次の例は、describe-fpga-images を使用する方法を説明しています。

AWS CLI

Amazon FPGAイメージを記述するには

この例ではAFIs、 アカウント が所有する について説明します123456789012。

コマンド:

```
aws ec2 describe-fpga-images --filters Name=owner-id,Values=123456789012
```

出力:

```
{
  "FpgaImages": [
    {
      "UpdateTime": "2017-12-22T12:09:14.000Z",
      "Name": "my-afi",
      "PciId": {
        "SubsystemVendorId": "0xfedd",
        "VendorId": "0x1d0f",
        "DeviceId": "0xf000",
        "SubsystemId": "0x1d51"
      },
      "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc",
      "Public": false,
      "State": {
        "Code": "available"
      },
      "ShellVersion": "0x071417d3",
      "OwnerId": "123456789012",
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "CreateTime": "2017-12-22T11:43:33.000Z",
      "Description": "my-afi"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFpgaImages](#)」の「」を参照してください。

describe-host-reservation-offerings

次のコード例は、describe-host-reservation-offerings を使用する方法を示しています。

AWS CLI

Dedicated Host Reservation のサービスを説明するには

この例では、購入可能な M4 インスタンスファミリーの Dedicated Host Reservations について説明します。

コマンド:

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

出力:

```
{
  "OfferingSet": [
    {
      "HourlyPrice": "1.499",
      "OfferingId": "hro-03f707bf363b6b324",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    },
    {
      "HourlyPrice": "1.045",
      "OfferingId": "hro-0ef9181cabdef7a02",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "UpfrontPrice": "0.000",
      "Duration": 94608000
    },
    {
      "HourlyPrice": "0.714",
      "OfferingId": "hro-04567a15500b92a51",
      "InstanceFamily": "m4",
      "PaymentOption": "PartialUpfront",
      "UpfrontPrice": "6254.000",
      "Duration": 31536000
    },
    {
      "HourlyPrice": "0.484",
      "OfferingId": "hro-0d5d7a9d23ed7fbfe",
      "InstanceFamily": "m4",
      "PaymentOption": "PartialUpfront",
      "UpfrontPrice": "12720.000",
      "Duration": 94608000
    },
    {
      "HourlyPrice": "0.000",
      "OfferingId": "hro-05da4108ca998c2e5",
```



```
    "InstanceFamily": "m4",
    "PaymentOption": "AllUpfront",
    "UpfrontPrice": "23913.000",
    "Duration": 94608000
  },
  {
    "HourlyPrice": "0.000",
    "OfferingId": "hro-0a9f9be3b95a3dc8f",
    "InstanceFamily": "m4",
    "PaymentOption": "AllUpfront",
    "UpfrontPrice": "12257.000",
    "Duration": 31536000
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeHostReservationOfferings](#)」の「」を参照してください。

describe-host-reservations

次の例は、describe-host-reservations を使用する方法を説明しています。

AWS CLI

アカウントの Dedicated Host Reservations を記述するには

この例では、アカウントの Dedicated Host Reservations について説明します。

コマンド:

```
aws ec2 describe-host-reservations
```

出力:

```
{
  "HostReservationSet": [
    {
      "Count": 1,
      "End": "2019-01-10T12:14:09Z",
      "HourlyPrice": "1.499",
      "InstanceFamily": "m4",
      "OfferingId": "hro-03f707bf363b6b324",
```

```
    "PaymentOption": "NoUpfront",
    "State": "active",
    "HostIdSet": [
      "h-013abcd2a00cbd123"
    ],
    "Start": "2018-01-10T12:14:09Z",
    "HostReservationId": "hr-0d418a3a4ffc669ae",
    "UpfrontPrice": "0.000",
    "Duration": 31536000
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeHostReservations](#)」の「」を参照してください。

describe-hosts

次の例は、describe-hosts を使用する方法を説明しています。

AWS CLI

Dedicated Hosts の詳細を表示するには

次のdescribe-hosts例では、AWS アカウントの available Dedicated Hosts の詳細を表示します。

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

出力:

```
{
  "Hosts": [
    {
      "HostId": "h-07879acf49EXAMPLE",
      "Tags": [
        {
          "Value": "production",
          "Key": "purpose"
        }
      ],
      "HostProperties": {
```

```
        "Cores": 48,  
        "TotalVCpus": 96,  
        "InstanceType": "m5.large",  
        "Sockets": 2  
    },  
    "Instances": [],  
    "State": "available",  
    "AvailabilityZone": "eu-west-1a",  
    "AvailableCapacity": {  
        "AvailableInstanceCapacity": [  
            {  
                "AvailableCapacity": 48,  
                "InstanceType": "m5.large",  
                "TotalCapacity": 48  
            }  
        ],  
        "AvailableVCpus": 96  
    },  
    "HostRecovery": "on",  
    "AllocationTime": "2019-08-19T08:57:44.000Z",  
    "AutoPlacement": "off"  
} ]  
}
```

詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「専有ホストの表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeHosts](#)」の「」を参照してください。

describe-iam-instance-profile-associations

次の例は、describe-iam-instance-profile-associations を使用する方法を説明しています。

AWS CLI

IAMインスタンスプロファイルの関連付けを記述するには

この例では、すべてのIAMインスタンスプロファイルの関連付けについて説明します。

コマンド:

```
aws ec2 describe-iam-instance-profile-associations
```

出力:

```
{
  "IamInstanceProfileAssociations": [
    {
      "InstanceId": "i-09eb09efa73ec1dee",
      "State": "associated",
      "AssociationId": "iip-assoc-0db249b1f25fa24b8",
      "IamInstanceProfile": {
        "Id": "AIPAJVQN4F5WVLGCJDRGM",
        "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
    },
    {
      "InstanceId": "i-0402909a2f4dfffd14",
      "State": "associating",
      "AssociationId": "iip-assoc-0d1ec06278d29f44a",
      "IamInstanceProfile": {
        "Id": "AGJAJVQN4F5WVLGCJABCM",
        "Arn": "arn:aws:iam::123456789012:instance-profile/user1-role"
      }
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeIamInstanceProfileAssociations](#)」の「」を参照してください。

describe-id-format

次の例は、describe-id-format を使用する方法を説明しています。

AWS CLI

例 1: リソースの ID 形式を記述するには

次のdescribe-id-format例では、セキュリティグループの ID 形式について説明します。

```
aws ec2 describe-id-format \
```

```
--resource security-group
```

次の出力例では、Deadlineこのリソースタイプの期限が 2018 年 8 月 15 日 00:00 UTCに有効期限切れになったことを示します。

```
{
  "Statuses": [
    {
      "Deadline": "2018-08-15T00:00:00.000Z",
      "Resource": "security-group",
      "UseLongIds": true
    }
  ]
}
```

例 2: すべてのリソースの ID 形式を記述するには

次のdescribe-id-format例では、すべてのリソースタイプの ID 形式について説明します。ショート ID 形式をサポートしたすべてのリソースタイプは、ロング ID 形式を使用するように切り替えられました。

```
aws ec2 describe-id-format
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeIdFormat](#)」の「」を参照してください。

describe-identity-id-format

次のコード例は、describe-identity-id-format を使用する方法を示しています。

AWS CLI

IAMロールの ID 形式を記述するには

次のdescribe-identity-id-format例では、AWS アカウントのIAMロールによって作成されたインスタンスEC2Roleによって受信される ID 形式について説明します。

```
aws ec2 describe-identity-id-format \
  --principal-arn arn:aws:iam::123456789012:role/my-iam-role \
  --resource instance
```

次の出力は、このロールによって作成されたインスタンスがIDs長い ID 形式で受信することを示します。

```
{
  "Statuses": [
    {
      "Deadline": "2016-12-15T00:00:00Z",
      "Resource": "instance",
      "UseLongIds": true
    }
  ]
}
```

IAMユーザーの ID 形式を記述するには

次のdescribe-identity-id-format例では、AWS アカウントAdminUser内のIAMユーザーが作成したスナップショットによって受信される ID 形式について説明します。

```
aws ec2 describe-identity-id-format \
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \
  --resource snapshot
```

出力は、このユーザーが作成したスナップショットがIDs長い ID 形式で受信されることを示します。

```
{
  "Statuses": [
    {
      "Deadline": "2016-12-15T00:00:00Z",
      "Resource": "snapshot",
      "UseLongIds": true
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeIdentityIdFormat](#)」の「」を参照してください。

describe-image-attribute

次の例は、describe-image-attribute を使用する方法を説明しています。

AWS CLI

の起動許可を記述するには AMI

この例では、指定された の起動許可について説明しますAMI。

コマンド:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --  
attribute LaunchPermission
```

出力:

```
{  
  "LaunchPermissions": [  
    {  
      "UserId": "123456789012"  
    }  
  ],  
  "ImageId": "ami-5731123e",  
}
```

の製品コードを記述するには AMI

この例では、指定された の製品コードについて説明しますAMI。これには製品コードAMIがないことに注意してください。

コマンド:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

出力:

```
{  
  "ProductCodes": [],  
  "ImageId": "ami-5731123e",  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeImageAttribute](#)」の「」を参照してください。

describe-images

次のコード例は、describe-images を使用する方法を示しています。

AWS CLI

例 1: を記述するには AMI

次のdescribe-images例では、指定されたリージョンAMIで指定された について説明します。

```
aws ec2 describe-images \  
  --region us-east-1 \  
  --image-ids ami-1234567890EXAMPLE
```

出力:

```
{  
  "Images": [  
    {  
      "VirtualizationType": "hvm",  
      "Description": "Provided by Red Hat, Inc.",  
      "PlatformDetails": "Red Hat Enterprise Linux",  
      "EnaSupport": true,  
      "Hypervisor": "xen",  
      "State": "available",  
      "SriovNetSupport": "simple",  
      "ImageId": "ami-1234567890EXAMPLE",  
      "UsageOperation": "RunInstances:0010",  
      "BlockDeviceMappings": [  
        {  
          "DeviceName": "/dev/sda1",  
          "Ebs": {  
            "SnapshotId": "snap-111222333444aaabb",  
            "DeleteOnTermination": true,  
            "VolumeType": "gp2",  
            "VolumeSize": 10,  
            "Encrypted": false  
          }  
        }  
      ],  
      "Architecture": "x86_64",  
      "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-  
GP2",
```



```

        "RootDeviceType": "ebs",
        "OwnerId": "123456789012",
        "RootDeviceName": "/dev/sda1",
        "CreationDate": "2019-05-10T13:17:12.000Z",
        "Public": true,
        "ImageType": "machine",
        "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
    }
]
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon マシンイメージ \(AMI\)」](#) を参照してください。 EC2

例 2: フィルターAMIsに基づいて記述するには

次のdescribe-images例では、Amazon AMIsが提供する Windows のうち、Amazon がサポートする Windows について説明しますEBS。

```

aws ec2 describe-images \
  --owners amazon \
  --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"

```

describe-images の出力例については、例 1 を参照してください。

フィルターを使用したその他の例については、「[Amazon EC2ユーザーガイド](#)」の「[リソースの一覧表示とフィルタリング](#)」を参照してください。

例 3: タグAMIsに基づいて記述するには

次のdescribe-images例では、タグ AMIs を持つすべての について説明しますType=Custom。この例では、--queryパラメータを使用して AMI のみを表示しますIDs。

```

aws ec2 describe-images \
  --filters "Name=tag:Type,Values=Custom" \
  --query 'Images[*].[ImageId]' \
  --output text

```

出力:

```

ami-1234567890EXAMPLE
ami-0abcdef1234567890

```

タグフィルターを使用するその他の例については、「Amazon EC2ユーザーガイド」の[「タグの操作」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeImages](#)」の「」を参照してください。

describe-import-image-tasks

次の例は、describe-import-image-tasks を使用する方法を説明しています。

AWS CLI

インポートイメージタスクをモニタリングするには

次のdescribe-import-image-tasks例では、指定されたインポートイメージタスクのステータスをチェックします。

```
aws ec2 describe-import-image-tasks \  
  --import-task-ids import-ami-1234567890abcdef0
```

進行中のインポートイメージタスクの出力。

```
{  
  "ImportImageTasks": [  
    {  
      "ImportTaskId": "import-ami-1234567890abcdef0",  
      "Progress": "28",  
      "SnapshotDetails": [  
        {  
          "DiskImageSize": 705638400.0,  
          "Format": "ova",  
          "Status": "completed",  
          "UserBucket": {  
            "S3Bucket": "my-import-bucket",  
            "S3Key": "vms/my-server-vm.ova"  
          }  
        }  
      ],  
      "Status": "active",  
      "StatusMessage": "converting"  
    }  
  ]  
}
```

```
}
```

完了したインポートイメージタスクの出力。結果の ID AMIは によって提供されますImageId。

```
{
  "ImportImageTasks": [
    {
      "ImportTaskId": "import-ami-1234567890abcdef0",
      "ImageId": "ami-1234567890abcdef0",
      "SnapshotDetails": [
        {
          "DiskImageSize": 705638400.0,
          "Format": "ova",
          "SnapshotId": "snap-1234567890abcdef0"
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "my-import-bucket",
            "S3Key": "vms/my-server-vm.ova"
          }
        }
      ],
      "Status": "completed"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeImportImageTasks](#)」の「」を参照してください。

describe-import-snapshot-tasks

次のコード例は、describe-import-snapshot-tasks を使用する方法を示しています。

AWS CLI

スナップショットのインポートタスクをモニタリングするには

次のdescribe-import-snapshot-tasks例では、指定されたインポートスナップショットタスクのステータスをチェックします。

```
aws ec2 describe-import-snapshot-tasks \
  --import-task-ids import-snap-1234567890abcdef0
```

進行中のインポートスナップショットタスクの出力 :

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "My server VMDK",
      "ImportTaskId": "import-snap-1234567890abcdef0",
      "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "705638400.0",
        "Format": "VMDK",
        "Progress": "42",
        "Status": "active",
        "StatusMessage": "downloading/convertng",
        "UserBucket": {
          "S3Bucket": "my-import-bucket",
          "S3Key": "vms/my-server-vm.vmdk"
        }
      }
    }
  ]
}
```

完了したインポートスナップショットタスクの出力。結果のスナップショットの ID は によって提供されます SnapshotId。

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "My server VMDK",
      "ImportTaskId": "import-snap-1234567890abcdef0",
      "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "705638400.0",
        "Format": "VMDK",
        "SnapshotId": "snap-1234567890abcdef0"
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "my-import-bucket",
          "S3Key": "vms/my-server-vm.vmdk"
        }
      }
    }
  ]
}
```

```
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeImportSnapshotTasks](#)」の「」を参照してください。

describe-instance-attribute

次の例は、describe-instance-attribute を使用する方法を説明しています。

AWS CLI

インスタンスタイプを記述するには

この例では、指定されたインスタンスのインスタンスタイプについて説明します。

コマンド:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute instanceType
```

出力:

```
{  
  "InstanceId": "i-1234567890abcdef0"  
  "InstanceType": {  
    "Value": "t1.micro"  
  }  
}
```

disableApiTermination 属性を記述するには

この例では、指定されたインスタンスの disableApiTermination 属性について説明します。

コマンド:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute disableApiTermination
```

出力:

```
{
```

```
"InstanceId": "i-1234567890abcdef0"
  "DisableApiTermination": {
    "Value": "false"
  }
}
```

インスタンスのブロックデバイスマッピングを記述するには

この例では、指定されたインスタンスの `blockDeviceMapping` 属性について説明します。

コマンド:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute blockDeviceMapping
```

出力:

```
{
  "InstanceId": "i-1234567890abcdef0"
  "BlockDeviceMappings": [
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": true,
        "VolumeId": "vol-049df61146c4d7901",
        "AttachTime": "2013-05-17T22:42:34.000Z"
      }
    },
    {
      "DeviceName": "/dev/sdf",
      "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": false,
        "VolumeId": "vol-049df61146c4d7901",
        "AttachTime": "2013-09-10T23:07:00.000Z"
      }
    }
  ],
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceAttribute](#)」の「」を参照してください。

describe-instance-connect-endpoints

次の例は、describe-instance-connect-endpoints を使用する方法を説明しています。

AWS CLI

EC2 Instance Connect エンドポイントを記述するには

次のdescribe-instance-connect-endpoints例では、指定された EC2 Instance Connect Endpoint について説明します。

```
aws ec2 describe-instance-connect-endpoints \  
  --region us-east-1 \  
  --instance-connect-endpoint-ids eice-0123456789example
```

出力:

```
{  
  "InstanceConnectEndpoints": [  
    {  
      "OwnerId": "111111111111",  
      "InstanceConnectEndpointId": "eice-0123456789example",  
      "InstanceConnectEndpointArn": "arn:aws:ec2:us-  
east-1:111111111111:instance-connect-endpoint/eice-0123456789example",  
      "State": "create-complete",  
      "StateMessage": "",  
      "DnsName": "eice-0123456789example.b67b86ba.ec2-instance-connect-  
endpoint.us-east-1.amazonaws.com",  
      "NetworkInterfaceIds": [  
        "eni-0123456789example"  
      ],  
      "VpcId": "vpc-0123abcd",  
      "AvailabilityZone": "us-east-1d",  
      "CreatedAt": "2023-02-07T12:05:37+00:00",  
      "SubnetId": "subnet-0123abcd",  
      "Tags": []  
    }  
  ]  
}
```

詳細については、「Amazon ユーザーガイド」の[EC2「インスタンス接続エンドポイントの作成」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceConnectEndpoints](#)」の「」を参照してください。

describe-instance-credit-specifications

次のコード例は、describe-instance-credit-specifications を使用する方法を示しています。

AWS CLI

1 つ以上のインスタンスCPUの使用に関するクレジットオプションを記述するには

次のdescribe-instance-credit-specifications例では、指定されたインスタンスのCPUクレジットオプションについて説明します。

```
aws ec2 describe-instance-credit-specifications \  
  --instance-ids i-1234567890abcdef0
```

出力:

```
{  
  "InstanceCreditSpecifications": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "CpuCredits": "unlimited"  
    }  
  ]  
}
```

詳細については、「Amazon EC2ユーザーガイド」の [「バーストパフォーマンスインスタンスの操作」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceCreditSpecifications](#)」の「」を参照してください。

describe-instance-event-notification-attributes

次の例は、describe-instance-event-notification-attributes を使用する方法を説明しています。

AWS CLI

スケジュールされたイベント通知のタグを記述するには

次のdescribe-instance-event-notification-attributes例では、スケジュールされたイベント通知に表示されるタグについて説明します。

```
aws ec2 describe-instance-event-notification-attributes
```

出力:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [],
    "IncludeAllTagsOfInstance": true
  }
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[インスタンスのスケジュールされたイベント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceEventNotificationAttributes](#)」の「」を参照してください。

describe-instance-event-windows

次のコード例は、describe-instance-event-windows を使用する方法を示しています。

AWS CLI

例 1: すべてのイベントウィンドウを記述するには

次のdescribe-instance-event-windows例では、指定されたリージョンのすべてのイベントウィンドウについて説明します。

```
aws ec2 describe-instance-event-windows \
  --region us-east-1
```

出力:

```
{
```

```

"InstanceEventWindows": [
  {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [
        "i-1234567890abcdef0",
        "i-0598c7d356eba48d7"
      ],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "active",
    "Tags": []
  }
  ...
],
"NextToken": "9d624e0c-388b-4862-a31e-a85c64fc1d4a"
}

```

例 2: 特定のイベントウィンドウを記述するには

次のdescribe-instance-event-windows例では、instance-event-windowパラメータを使用して特定のイベントウィンドウを記述することで、特定のイベントについて説明します。

```

aws ec2 describe-instance-event-windows \
  --region us-east-1 \
  --instance-event-window-ids iew-0abcdef1234567890

```

出力:

```

{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "Name": "myEventWindowName",
      "CronExpression": "* 21-23 * * 2,3",
      "AssociationTarget": {
        "InstanceIds": [
          "i-1234567890abcdef0",

```

```

        "i-0598c7d356eba48d7"
    ],
    "Tags": [],
    "DedicatedHostIds": []
  },
  "State": "active",
  "Tags": []
}
}

```

例 3: 1 つ以上のフィルターに一致するイベントウィンドウを記述するには

次のdescribe-instance-event-windows例では、filterパラメータを使用して1つ以上のフィルターに一致するイベントウィンドウについて説明します。instance-id フィルターは、指定されたインスタンスに関連付けられているすべてのイベントウィンドウを記述するために使用されます。フィルタを使用すると、直接的な一致が評価されます。ただし、instance-id フィルターの場合は異なります。インスタンス ID に直接一致しない場合、インスタンスのタグや Dedicated Host ID (インスタンスが Dedicated Host の場合) など、イベントウィンドウとの間接的な関連付けにフォールバックされます。

```

aws ec2 describe-instance-event-windows \
  --region us-east-1 \
  --filters Name=instance-id,Values=i-1234567890abcdef0 \
  --max-results 100 \
  --next-token <next-token-value>

```

出力:

```

{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0dbc0adb66f235982",
      "TimeRanges": [
        {
          "StartWeekDay": "sunday",
          "StartHour": 2,
          "EndWeekDay": "sunday",
          "EndHour": 8
        }
      ],
      "Name": "myEventWindowName",
      "AssociationTarget": {

```

```
        "InstanceIds": [],
        "Tags": [],
        "DedicatedHostIds": [
            "h-0140d9a7ecbd102dd"
        ]
    },
    "State": "active",
    "Tags": []
}
]
```

この例では、インスタンスは Dedicated Host にあり、イベントウィンドウに関連付けられていません。

イベントウィンドウの制約については、「Amazon ユーザーガイド」の「[考慮事項](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInstanceEventWindows](#)」の「」を参照してください。

describe-instance-status

次の例は、describe-instance-status を使用方法を説明しています。

AWS CLI

インスタンスのステータスを表示するには

次の describe-instance-status の例では、指定したインスタンスの現在のステータスを示しています。

```
aws ec2 describe-instance-status \
  --instance-ids i-1234567890abcdef0
```

出力:

```
{
  "InstanceStatuses": [
    {
      "InstanceId": "i-1234567890abcdef0",
```

```
    "InstanceState": {
      "Code": 16,
      "Name": "running"
    },
    "AvailabilityZone": "us-east-1d",
    "SystemStatus": {
      "Status": "ok",
      "Details": [
        {
          "Status": "passed",
          "Name": "reachability"
        }
      ]
    },
    "InstanceStatus": {
      "Status": "ok",
      "Details": [
        {
          "Status": "passed",
          "Name": "reachability"
        }
      ]
    }
  }
]
```

詳細については、「[Amazon ユーザーガイド](#)」の「[インスタンスのステータスのモニタリング](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeInstanceStatus](#)」の「」を参照してください。

describe-instance-topology

次のコード例は、describe-instance-topology を使用する方法を示しています。

AWS CLI

すべてのインスタンスのインスタストポロジを記述するには

次のdescribe-instance-topology例では、このコマンドでサポートされているインスタスタイプに一致するすべてのインスタスのトポロジについて説明します。

```
aws ec2 describe-instance-topology \  
--region us-west-2
```

出力:

```
{  
  "Instances": [  
    {  
      "InstanceId": "i-1111111111example",  
      "InstanceType": "p4d.24xlarge",  
      "GroupName": "my-m1-cpg",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-2222222222example",  
      "InstanceType": "p4d.24xlarge",  
      "NetworkNodes": [  
        "nn-1111111111example",  
        "nn-2222222222example",  
        "nn-3333333333example"  
      ],  
      "ZoneId": "usw2-az2",  
      "AvailabilityZone": "us-west-2a"  
    },  
    {  
      "InstanceId": "i-3333333333example",  
      "InstanceType": "trn1.32xlarge",  
      "NetworkNodes": [  
        "nn-1212121212example",  
        "nn-1211122211example",  
        "nn-1311133311example"  
      ],  
      "ZoneId": "usw2-az4",  
      "AvailabilityZone": "us-west-2d"  
    },  
    {  
      "InstanceId": "i-4444444444example",
```

```
        "InstanceType": "trn1.2xlarge",
        "NetworkNodes": [
            "nn-1111111111example",
            "nn-5434334334example",
            "nn-1235301234example"
        ],
        "ZoneId": "usw2-az2",
        "AvailabilityZone": "us-west-2a"
    }
],
"NextToken": "SomeEncryptedToken"
}
```

その他の例を含む詳細については、[「Amazon ユーザーガイド」の「Amazon EC2 インスタンス トポロジ」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceTopology](#)」の「」を参照してください。

describe-instance-type-offerings

次のコード例は、describe-instance-type-offerings を使用する方法を示しています。

AWS CLI

例 1: リージョンで提供されるインスタンスタイプを一覧表示するには

次のdescribe-instance-type-offerings例では、のデフォルトリージョンとして設定されたリージョンで提供されるインスタンスタイプを一覧表示します AWS CLI。

```
aws ec2 describe-instance-type-offerings
```

別のリージョンで提供されるインスタンスタイプを一覧表示するには、--regionパラメータを使用してリージョンを指定します。

```
aws ec2 describe-instance-type-offerings \
  --region us-east-2
```

出力:

```
{
  "InstanceTypeOfferings": [
```

```
{
  "InstanceType": "m5.2xlarge",
  "LocationType": "region",
  "Location": "us-east-2"
},
{
  "InstanceType": "t3.micro",
  "LocationType": "region",
  "Location": "us-east-2"
},
...
]
```

例 2: アベイラビリティゾーンで提供されるインスタンスタイプを一覧表示するには

次のdescribe-instance-type-offerings例では、指定されたアベイラビリティゾーンで提供されるインスタンスタイプを一覧表示します。アベイラビリティゾーンは、指定されたリージョンにある必要があります。

```
aws ec2 describe-instance-type-offerings \
  --location-type availability-zone \
  --filters Name=location,Values=us-east-2a \
  --region us-east-2
```

例 3: インスタンスタイプがサポートされているかどうかを確認するには

次のdescribe-instance-type-offeringsコマンドは、c5.xlargeインスタンスタイプが指定されたリージョンでサポートされているかどうかを示します。

```
aws ec2 describe-instance-type-offerings \
  --filters Name=instance-type,Values=c5.xlarge \
  --region us-east-2
```

次のdescribe-instance-type-offerings例では、指定されたリージョンでサポートされているすべての C5 インスタンスタイプを一覧表示します。

```
aws ec2 describe-instance-type-offerings \
  --filters Name=instance-type,Values=c5* \
  --query "InstanceTypeOfferings[].InstanceType" \
  --region us-east-2
```


出力:

```
[
  "c5d.12xlarge",
  "c5d.9xlarge",
  "c5n.xlarge",
  "c5.xlarge",
  "c5d.metal",
  "c5n.metal",
  "c5.large",
  "c5d.2xlarge",
  "c5n.4xlarge",
  "c5.2xlarge",
  "c5n.large",
  "c5n.9xlarge",
  "c5d.large",
  "c5.18xlarge",
  "c5d.18xlarge",
  "c5.12xlarge",
  "c5n.18xlarge",
  "c5.metal",
  "c5d.4xlarge",
  "c5.24xlarge",
  "c5d.xlarge",
  "c5n.2xlarge",
  "c5d.24xlarge",
  "c5.9xlarge",
  "c5.4xlarge"
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceTypeOfferings](#)」の「」を参照してください。

describe-instance-types

次のコード例は、describe-instance-types を使用する方法を示しています。

AWS CLI

例 1: インスタンスタイプを説明するには

次の describe-instance-types の例では、指定されたインスタンスタイプの詳細を表示します。

```
aws ec2 describe-instance-types \  
--instance-types t2.micro
```

出力:

```
{  
  "InstanceTypes": [  
    {  
      "InstanceType": "t2.micro",  
      "CurrentGeneration": true,  
      "FreeTierEligible": true,  
      "SupportedUsageClasses": [  
        "on-demand",  
        "spot"  
      ],  
      "SupportedRootDeviceTypes": [  
        "ebs"  
      ],  
      "BareMetal": false,  
      "Hypervisor": "xen",  
      "ProcessorInfo": {  
        "SupportedArchitectures": [  
          "i386",  
          "x86_64"  
        ],  
        "SustainedClockSpeedInGhz": 2.5  
      },  
      "VCpuInfo": {  
        "DefaultVCpus": 1,  
        "DefaultCores": 1,  
        "DefaultThreadsPerCore": 1,  
        "ValidCores": [  
          1  
        ],  
        "ValidThreadsPerCore": [  
          1  
        ]  
      },  
      "MemoryInfo": {  
        "SizeInMiB": 1024  
      },  
      "InstanceStorageSupported": false,  
      "EbsInfo": {
```

```

        "EbsOptimizedSupport": "unsupported",
        "EncryptionSupport": "supported"
    },
    "NetworkInfo": {
        "NetworkPerformance": "Low to Moderate",
        "MaximumNetworkInterfaces": 2,
        "Ipv4AddressesPerInterface": 2,
        "Ipv6AddressesPerInterface": 2,
        "Ipv6Supported": true,
        "EnaSupport": "unsupported"
    },
    "PlacementGroupInfo": {
        "SupportedStrategies": [
            "partition",
            "spread"
        ]
    },
    "HibernationSupported": false,
    "BurstablePerformanceSupported": true,
    "DedicatedHostsSupported": false,
    "AutoRecoverySupported": true
    }
]
}

```

詳細については、「Linux [インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド](#)」の「[インスタンスタイプ](#)」を参照してください。

例 2: 使用可能なインスタンスタイプをフィルタリングするには

フィルターを指定して、特定の特性を持つインスタンスタイプに結果を絞り込みます。次の describe-instance-types の例では、休止状態をサポートするインスタンスタイプを一覧表示しています。

```

aws ec2 describe-instance-types \
  --filters Name=hibernation-supported,Values=true --query
  'InstanceTypes[*].InstanceType'

```

出力:

```

[
  "m5.8xlarge",
  "r3.large",

```

```
"c3.8xlarge",  
"r5.large",  
"m4.4xlarge",  
"c4.large",  
"m5.xlarge",  
"m4.xlarge",  
"c3.large",  
"c4.8xlarge",  
"c4.4xlarge",  
"c5.xlarge",  
"c5.12xlarge",  
"r5.4xlarge",  
"c5.4xlarge"  
]
```

詳細については、「Linux [インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド](#)」の「[インスタンスタイプ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInstanceTypes](#)」の「」を参照してください。

describe-instances

次のコード例は、describe-instances を使用する方法を示しています。

AWS CLI

例 1: インスタンスを説明するには

次の describe-instances の例では、指定したインスタンスを示しています。

```
aws ec2 describe-instances \  
  --instance-ids i-1234567890abcdef0
```

出力:

```
{  
  "Reservations": [  
    {  
      "Groups": [],  
      "Instances": [  
        {  
          "AmiLaunchIndex": 0,
```

```
"ImageId": "ami-0abcdef1234567890",
"InstanceId": "i-1234567890abcdef0",
"InstanceType": "t3.nano",
"KeyName": "my-key-pair",
"LaunchTime": "2022-11-15T10:48:59+00:00",
"Monitoring": {
  "State": "disabled"
},
"Placement": {
  "AvailabilityZone": "us-east-2a",
  "GroupName": "",
  "Tenancy": "default"
},
"PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
"PrivateIpAddress": "10-0-0-157",
"ProductCodes": [],
"PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
"PublicIpAddress": "34.253.223.13",
"State": {
  "Code": 16,
  "Name": "running"
},
"StateTransitionReason": "",
"SubnetId": "subnet-04a636d18e83cfacb",
"VpcId": "vpc-1234567890abcdef0",
"Architecture": "x86_64",
"BlockDeviceMappings": [
  {
    "DeviceName": "/dev/xvda",
    "Ebs": {
      "AttachTime": "2022-11-15T10:49:00+00:00",
      "DeleteOnTermination": true,
      "Status": "attached",
      "VolumeId": "vol-02e6ccdca7de29cf2"
    }
  }
],
"ClientToken": "1234abcd-1234-abcd-1234-d46a8903e9bc",
"EbsOptimized": true,
"EnaSupport": true,
"Hypervisor": "xen",
"IamInstanceProfile": {
```

```
        "Arn": "arn:aws:iam::111111111111:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
        "Id": "11111111111111111111111111111111"
    },
    "NetworkInterfaces": [
        {
            "Association": {
                "IpOwnerId": "amazon",
                "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                "PublicIp": "34.253.223.13"
            },
            "Attachment": {
                "AttachTime": "2022-11-15T10:48:59+00:00",
                "AttachmentId": "eni-attach-1234567890abcdefg",
                "DeleteOnTermination": true,
                "DeviceIndex": 0,
                "Status": "attached",
                "NetworkCardIndex": 0
            },
            "Description": "",
            "Groups": [
                {
                    "GroupName": "launch-wizard-146",
                    "GroupId": "sg-1234567890abcdefg"
                }
            ],
            "Ipv6Addresses": [],
            "MacAddress": "00:11:22:33:44:55",
            "NetworkInterfaceId": "eni-1234567890abcdefg",
            "OwnerId": "104024344472",
            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
            "PrivateIpAddress": "10-0-0-157",
            "PrivateIpAddresses": [
                {
                    "Association": {
                        "IpOwnerId": "amazon",
                        "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                        "PublicIp": "34.253.223.13"
                    },
                    "Primary": true,
```

```
        "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
        "PrivateIpAddress": "10-0-0-157"
    }
],
"SourceDestCheck": true,
"Status": "in-use",
"SubnetId": "subnet-1234567890abcdefg",
"VpcId": "vpc-1234567890abcdefg",
"InterfaceType": "interface"
}
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "launch-wizard-146",
        "GroupId": "sg-1234567890abcdefg"
    }
],
"SourceDestCheck": true,
"Tags": [
    {
        "Key": "Name",
        "Value": "my-instance"
    }
],
"VirtualizationType": "hvm",
"CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 2
},
"CapacityReservationSpecification": {
    "CapacityReservationPreference": "open"
},
"HibernationOptions": {
    "Configured": false
},
"MetadataOptions": {
    "State": "applied",
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "enabled",
    "HttpProtocolIpv6": "disabled",
```

```

        "InstanceMetadataTags": "enabled"
    },
    "EnclaveOptions": {
        "Enabled": false
    },
    "PlatformDetails": "Linux/UNIX",
    "UsageOperation": "RunInstances",
    "UsageOperationUpdateTime": "2022-11-15T10:48:59+00:00",
    "PrivateDnsNameOptions": {
        "HostnameType": "ip-name",
        "EnableResourceNameDnsARecord": true,
        "EnableResourceNameDnsAAAARecord": false
    },
    "MaintenanceOptions": {
        "AutoRecovery": "default"
    }
}
],
"OwnerId": "111111111111",
"ReservationId": "r-1234567890abcdefg"
}
]
}

```

例 2: 指定したタイプのインスタンスをフィルタリングするには

次の `describe-instances` の例では、フィルターを使用して、指定されたタイプのインスタンスに結果の範囲を限定しています。

```
aws ec2 describe-instances \
  --filters Name=instance-type,Values=m5.large
```

出力例については、例 1 を参照してください。

詳細については、「Amazon ユーザーガイド」の [「を使用したリストとフィルタリングCLI」](#) を参照してください。 EC2

例 3: 指定したタイプとアベイラビリティゾーンでインスタンスをフィルタリングするには

次の `describe-instances` の例では、複数のフィルターを使用して、指定されたアベイラビリティゾーンにある、指定されたタイプのインスタンスに結果を絞り込みます。

```
aws ec2 describe-instances \
```



```
--filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-zone,Values=us-east-2c
```

出力例については、例 1 を参照してください。

例 4: JSON ファイルを使用して、指定されたタイプとアベイラビリティゾーンを持つインスタンスをフィルタリングするには

次の describe-instances 例では、JSON 入力ファイルを使用して、前の例と同じフィルタリングを実行します。フィルターがより複雑になると、JSON ファイルで簡単に指定できます。

```
aws ec2 describe-instances \  
--filters file://filters.json
```

filters.json の内容:

```
[  
  {  
    "Name": "instance-type",  
    "Values": ["t2.micro", "t3.micro"]  
  },  
  {  
    "Name": "availability-zone",  
    "Values": ["us-east-2c"]  
  }  
]
```

出力例については、例 1 を参照してください。

例 5: 指定した Owner タグを持つインスタンスをフィルタリングするには

次の describe-instances の例では、タグフィルターを使用して、タグ値に関係なく、指定されたタグキー (Owner) のタグを持つインスタンスに結果を絞り込みます。

```
aws ec2 describe-instances \  
--filters "Name=tag-key,Values=Owner"
```

出力例については、例 1 を参照してください。

例 6: 指定した my-team タグ値を持つインスタンスをフィルタリングするには

次の describe-instances の例では、タグフィルターを使用して、タグキーに関係なく、指定されたタグ値 (my-team) のタグを持つインスタンスに結果を絞り込みます。

```
aws ec2 describe-instances \  
  --filters "Name=tag-value,Values=my-team"
```

出力例については、例 1 を参照してください。

例 7: 指定した Owner タグと my-team 値を持つインスタンスをフィルタリングするには

次の describe-instances の例では、タグフィルターを使用して、指定したタグ (Owner=my-team) を持つインスタンスに結果を絞り込みます。

```
aws ec2 describe-instances \  
  --filters "Name=tag:Owner,Values=my-team"
```

出力例については、例 1 を参照してください。

例 8: すべてのインスタンスIDsのインスタンスとサブネットのみを表示するには

次の describe-instances 例では、--query パラメータを使用して、IDs すべてのインスタンスのインスタンスとサブネットのみを JSON 形式で表示します。

Linux および macOS:

```
aws ec2 describe-instances \  
  --query 'Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}' \  
  --output json
```

Windows :

```
aws ec2 describe-instances ^  
  --query "Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}" ^  
  --output json
```

出力:

```
[  
  {
```

```

    "Instance": "i-057750d42936e468a",
    "Subnet": "subnet-069beee9b12030077"
  },
  {
    "Instance": "i-001efd250faaa6ffa",
    "Subnet": "subnet-0b715c6b7db68927a"
  },
  {
    "Instance": "i-027552a73f021f3bd",
    "Subnet": "subnet-0250c25a1f4e15235"
  }
  ...
]

```

例 9: 指定されたタイプのインスタンスをフィルタリングし、そのインスタンスのみを表示するには IDs

次の describe-instances 例では、フィルターを使用して、指定されたタイプのインスタンスに結果をスコープし、--query パラメータを使用してインスタンスのみを表示します IDs。

```

aws ec2 describe-instances \
  --filters "Name=instance-type,Values=t2.micro" \
  --query "Reservations[*].Instances[*].[InstanceId]" \
  --output text

```

出力:

```

i-031c0dc19de2fb70c
i-00d8bff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c

```

例 10: 指定されたタイプのインスタンスをフィルタリングし、そのインスタンス IDs、アベイラビリティゾーン、および指定されたタグ値のみを表示するには

次の describe-instances の例では、tag-key という名前のタグを持つインスタンスのインスタンス ID、アベイラビリティゾーン、および Name タグの値を表形式で表示します。

Linux および macOS:

```
aws ec2 describe-instances \
  --filters Name=tag-key,Values=Name \
  --query 'Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key=='Name `]]
[0].Value}' \
  --output table
```

Windows :

```
aws ec2 describe-instances ^
  --filters Name=tag-key,Values=Name ^
  --query "Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key=='Name `]]
[0].Value}" ^
  --output table
```

出力:

```
-----
|                               DescribeInstances                               |
+-----+-----+-----+
|      AZ      | Instance      |      Name      |
+-----+-----+-----+
| us-east-2b  | i-057750d42936e468a | my-prod-server |
| us-east-2a  | i-001efd250faaa6ffa | test-server-1   |
| us-east-2a  | i-027552a73f021f3bd | test-server-2   |
+-----+-----+-----+
```

例 11: パーティションプレースメントグループ内のインスタンスを説明するには

次の describe-instances の例では、指定したインスタンスを示しています。出力にはインスタンスのプレースメント情報が含まれています。この情報にはインスタンスのプレースメントグループ名とパーティション番号が含まれます。

```
aws ec2 describe-instances \
  --instance-ids i-0123a456700123456 \
  --query "Reservations[*].Instances[*].Placement"
```

出力:

```
[
```

```
[
  {
    "AvailabilityZone": "us-east-1c",
    "GroupName": "HDFS-Group-A",
    "PartitionNumber": 3,
    "Tenancy": "default"
  }
]
```

詳細については、「Amazon EC2ユーザーガイド」の[「プレイスメントグループのインスタンスの説明」](#)を参照してください。

例 12: 指定したプレイスメントグループとパーティション番号を持つインスタンスでフィルタリングするには

次の describe-instances の例では、指定したプレイスメントグループとパーティション番号のインスタンスのみに結果をフィルタリングします。

```
aws ec2 describe-instances \
  --filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-
  partition-number,Values=7"
```

次の例では、出力の関連情報のみが表示されます。

```
"Instances": [
  {
    "InstanceId": "i-0123a456700123456",
    "InstanceType": "r4.large",
    "Placement": {
      "AvailabilityZone": "us-east-1c",
      "GroupName": "HDFS-Group-A",
      "PartitionNumber": 7,
      "Tenancy": "default"
    }
  },
  {
    "InstanceId": "i-9876a543210987654",
    "InstanceType": "r4.large",
    "Placement": {
      "AvailabilityZone": "us-east-1c",
```

```
        "GroupName": "HDFS-Group-A",
        "PartitionNumber": 7,
        "Tenancy": "default"
    }
],
```

詳細については、「Amazon EC2ユーザーガイド」の[「プレースメントグループのインスタンスの説明」](#)を参照してください。

例 13: インスタンスメタデータのタグへのアクセスを許可するように設定されているインスタンスに絞り込むには

次の describe-instances の例では、インスタンスメタデータからインスタスタグへのアクセスを許可するように設定されているインスタンスのみに結果をフィルタリングします。

```
aws ec2 describe-instances \
  --filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \
  --query "Reservations[*].Instances[*].InstanceId" \
  --output text
```

次のような出力が予想されます。

```
i-1234567890abcdefg
i-abcdefg1234567890
i-1111111111aaaaaaaa
i-aaaaaaaa1111111111
```

詳細については、「Amazon EC2ユーザーガイド」の[「インスタンスメタデータでのインスタスタグの操作」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInstances](#)」の「」を参照してください。

describe-internet-gateways

次の例は、describe-internet-gateways を使用する方法を説明しています。

AWS CLI

インターネットゲートウェイを記述するには

次のdescribe-internet-gateways例では、指定されたインターネットゲートウェイについて説明します。

```
aws ec2 describe-internet-gateways \
  --internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

出力:

```
{
  "InternetGateways": [
    {
      "Attachments": [
        {
          "State": "available",
          "VpcId": "vpc-0a60eb65b4EXAMPLE"
        }
      ],
      "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",
      "OwnerId": "123456789012",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-igw"
        }
      ]
    }
  ]
}
```

詳細については、「Amazon VPCユーザーガイド」の「[インターネットゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInternetGateways](#)」の「」を参照してください。

describe-ipam-pools

次のコード例は、describe-ipam-pools を使用する方法を示しています。

AWS CLI

IPAM プールの詳細を表示するには

次のdescribe-ipam-pools例は、プールの詳細を示しています。

(Linux):

```
aws ec2 describe-ipam-pools \  
  --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-  
scope-02fc38cd4c48e7d38
```

(Windows):

```
aws ec2 describe-ipam-pools ^  
  --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-  
scope-02fc38cd4c48e7d38
```

出力:

```
{  
  "IpamPools": [  
    {  
      "OwnerId": "123456789012",  
      "IpamPoolId": "ipam-pool-02ec043a19bbe5d08",  
      "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-  
pool-02ec043a19bbe5d08",  
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-02fc38cd4c48e7d38",  
      "IpamScopeType": "private",  
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
      "IpamRegion": "us-east-1",  
      "Locale": "None",  
      "PoolDepth": 1,  
      "State": "create-complete",  
      "AutoImport": true,  
      "AddressFamily": "ipv4",  
      "AllocationMinNetmaskLength": 16,  
      "AllocationMaxNetmaskLength": 26,  
      "AllocationDefaultNetmaskLength": 24,  
      "AllocationResourceTags": [  
        {  
          "Key": "Environment",  
          "Value": "Preprod"  
        }  
      ],  
    },  
  ],  
}
```



```
    "Tags": [
      {
        "Key": "Name",
        "Value": "Preprod pool"
      }
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeIpamPools](#)」の「」を参照してください。

describe-ipam-resource-discoveries

次の例は、describe-ipam-resource-discoveries を使用する方法を説明しています。

AWS CLI

例 1: リソース検出の詳細を表示する

この例では、リソース検出を作成して別の AWS Organization IPAM の IPAM 管理者と共有し、管理者が組織内のリソースの IP アドレスを管理およびモニタリングできるようにする委任管理者です。

この例は、次の場合に便利です。

リソース検出を作成しようとしたが、上限の 1 に達したというエラーが表示されました。リソース検出を既に作成していて、アカウントで表示したい可能性があることを認識しています。によって検出されていないリソースがリージョンに存在します IPAM。リソースに --operating-regions 定義されている を表示し、適切なリージョンをオペレーションリージョンとして追加して、そのリソースを検出できるようにする必要があります。

次の describe-ipam-resource-discoveries 例では、AWS アカウントのリソース検出の詳細を一覧表示します。AWS リージョンごとに 1 つのリソース検出を行うことができます。

```
aws ec2 describe-ipam-resource-discoveries \
  --region us-east-1
```

出力:

```
{
  "IpamResourceDiscoveries": [
    {
      "OwnerId": "149977607591",
      "IpamResourceDiscoveryId": "ipam-res-disco-0f8bdee9067137c0d",
      "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-
discovery/ipam-res-disco-0f8bdee9067137c0d",
      "IpamResourceDiscoveryRegion": "us-east-1",
      "OperatingRegions": [
        {
          "RegionName": "us-east-1"
        }
      ],
      "IsDefault": false,
      "State": "create-complete",
      "Tags": []
    }
  ]
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の[「組織外のアカウントIPAMとの統合」](#)を参照してください。

例 2: リソース検出のみを表示する IDs

次のdescribe-ipam-resource-discoveries例では、AWS アカウントのリソース検出の ID を一覧表示します。AWS リージョンごとに 1 つのリソース検出を行うことができます。

```
aws ec2 describe-ipam-resource-discoveries \
  --query "IpamResourceDiscoveries[*].IpamResourceDiscoveryId" \
  --output text
```

出力:

```
ipam-res-disco-0481e39b242860333
```

詳細については、「Amazon VPCIPAMユーザーガイド」の[「組織外のアカウントIPAMとの統合」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeIpamResourceDiscoveries](#)」の「」を参照してください。

describe-ipam-resource-discovery-associations

次の例は、describe-ipam-resource-discovery-associations を使用する方法を説明しています。

AWS CLI

とのすべてのリソース検出の関連付けを表示するには IPAM

この例では、リソース検出をに関連付けIPAMで、他のアカウントをと統合しているIPAM委任管理者ですIPAM。IPAM がリソース検出のオペレーションリージョンのリソースを期待どおりに検出していないことに気付きました。リソース検出のステータスと状態をチェックして、リソース検出を作成したアカウントがまだアクティブであり、リソース検出が共有されていることを確認します。

は、のホームリージョン--regionである必要がありますIPAM。

次のdescribe-ipam-resource-discovery-associations例では、AWS アカウントのリソース検出の関連付けを一覧表示します。

```
aws ec2 describe-ipam-resource-discovery-associations \  
  --region us-east-1
```

出力:

```
{  
  "IpamResourceDiscoveryAssociations": [  
    {  
      "OwnerId": "320805250157",  
      "IpamResourceDiscoveryAssociationId": "ipam-res-disco-  
assoc-05e6b45eca5bf5cf7",  
      "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::320805250157:ipam-  
resource-discovery-association/ipam-res-disco-assoc-05e6b45eca5bf5cf7",  
      "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",  
      "IpamId": "ipam-005f921c17ebd5107",  
      "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",  
      "IpamRegion": "us-east-1",  
      "IsDefault": true,  
      "ResourceDiscoveryStatus": "active",  
      "State": "associate-complete",  
      "Tags": []  
    },  
  ],  
}
```

```

    {
      "OwnerId": "149977607591",
      "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-0dfd21ae189ab5f62",
      "IpamResourceDiscoveryAssociationArn": "arn:aws:ec2::149977607591:ipam-
resource-discovery-association/ipam-res-disco-assoc-0dfd21ae189ab5f62",
      "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
      "IpamId": "ipam-005f921c17ebd5107",
      "IpamArn": "arn:aws:ec2::149977607591:ipam/ipam-005f921c17ebd5107",
      "IpamRegion": "us-east-1",
      "IsDefault": false,
      "ResourceDiscoveryStatus": "active",
      "State": "create-complete",
      "Tags": []
    }
  ]
}

```

この例では、このコマンドを実行した後、デフォルト以外のリソース検出が "IsDefault": false ` `) that is ` ` "ResourceDiscoveryStatus": "not-found" 1 つ (と) あることに気づきます "State": "create-complete"。リソース検出所有者のアカウントは閉じられました。別の場合、 が "ResourceDiscoveryStatus": "not-found" と であることがわかった場合は "State": "associate-complete"、次のいずれかが発生したことを示します。

リソース検出は、リソース検出所有者によって削除されました。リソース検出所有者は、リソース検出の共有を解除しました。

詳細については、「Amazon VPC IPAM ユーザーガイド」の [「組織外のアカウント IPAM との統合」](#) を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス [DescribeIpamResourceDiscoveryAssociations](#) 」の 「 」 を参照してください。

describe-ipam-scopes

次のコード例は、describe-ipam-scopes を使用する方法を示しています。

AWS CLI

IPAM スコープの詳細を表示するには

次の describe-ipam-scopes 例は、スコープの詳細を示しています。

```
aws ec2 describe-ipam-scopes \  
  --filters Name=owner-id,Values=123456789012 Name=ipam-  
  id,Values=ipam-08440e7a3acde3908
```

出力:

```
{  
  "IpamScopes": [  
    {  
      "OwnerId": "123456789012",  
      "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",  
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-02fc38cd4c48e7d38",  
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
      "IpamRegion": "us-east-1",  
      "IpamScopeType": "private",  
      "IsDefault": true,  
      "PoolCount": 2,  
      "State": "create-complete",  
      "Tags": []  
    },  
    {  
      "OwnerId": "123456789012",  
      "IpamScopeId": "ipam-scope-0b9eed026396dbc16",  
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-0b9eed026396dbc16",  
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
      "IpamRegion": "us-east-1",  
      "IpamScopeType": "public",  
      "IsDefault": true,  
      "PoolCount": 0,  
      "State": "create-complete",  
      "Tags": []  
    },  
    {  
      "OwnerId": "123456789012",  
      "IpamScopeId": "ipam-scope-0f1aff29486355c22",  
      "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-0f1aff29486355c22",  
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
      "IpamRegion": "us-east-1",  
      "IpamScopeType": "private",  
      "IsDefault": false,  
    }  
  ]  
}
```

```

    "Description": "Example description",
    "PoolCount": 0,
    "State": "create-complete",
    "Tags": [
      {
        "Key": "Name",
        "Value": "Example name value"
      }
    ]
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeIpamScopes](#)」の「」を参照してください。

describe-ipams

次のコード例は、describe-ipams を使用する方法を示しています。

AWS CLI

の詳細を表示するには IPAM

次のdescribe-ipams例は、の詳細を示していますIPAM。

```

aws ec2 describe-ipams \
  --filters Name=owner-id,Values=123456789012

```

出力:

```

{
  "Ipams": [
    {
      "OwnerId": "123456789012",
      "IpamId": "ipam-08440e7a3acde3908",
      "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
      "IpamRegion": "us-east-1",
      "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
      "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
      "ScopeCount": 3,
      "OperatingRegions": [

```

```
    {
      "RegionName": "us-east-1"
    },
    {
      "RegionName": "us-east-2"
    },
    {
      "RegionName": "us-west-1"
    }
  ],
  "State": "create-complete",
  "Tags": [
    {
      "Key": "Name",
      "Value": "ExampleIPAM"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeIpams](#)」の「」を参照してください。

describe-ipv6-pools

次の例は、describe-ipv6-pools を使用する方法を説明しています。

AWS CLI

IPv6 アドレスプールを記述するには

次のdescribe-ipv6-pools例では、すべてのIPv6アドレスプールの詳細を表示します。

```
aws ec2 describe-ipv6-pools
```

出力:

```
{
  "Ipv6Pools": [
    {
      "PoolId": "ipv6pool-ec2-012345abc12345abc",
```

```
    "PoolCidrBlocks": [  
      {  
        "Cidr": "2001:db8:123::/48"  
      }  
    ],  
    "Tags": [  
      {  
        "Key": "pool-1",  
        "Value": "public"  
      }  
    ]  
  }  
]
```

- API 詳細については、[DescribeIpv6Pools](#) を参照してください。AWS CLI

describe-key-pairs

次のコード例は、describe-key-pairs を使用する方法を示しています。

AWS CLI

キーペアを表示するには

次の describe-key-pairs の例では、指定されたキーペアの情報が表示されます。

```
aws ec2 describe-key-pairs \  
  --key-names my-key-pair
```

出力:

```
{  
  "KeyPairs": [  
    {  
      "KeyPairId": "key-0b94643da6EXAMPLE",  
      "KeyFingerprint":  
"1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",  
      "KeyName": "my-key-pair",  
      "KeyType": "rsa",  
      "Tags": [],  
      "CreateTime": "2022-05-27T21:51:16.000Z"  
    }  
  ]  
}
```



```
]
}
```

詳細については、「Amazon ユーザーガイド」の「[パブリックキーの説明](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeKeyPairs](#)」の「」を参照してください。

describe-launch-template-versions

次の例は、describe-launch-template-versions を使用する方法を説明しています。

AWS CLI

起動テンプレートのバージョンを記述するには

この例では、指定された起動テンプレートのバージョンについて説明します。

コマンド:

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b72934aff71
```

出力:

```
{
  "LaunchTemplateVersions": [
    {
      "LaunchTemplateId": "lt-068f72b72934aff71",
      "LaunchTemplateName": "Webservers",
      "VersionNumber": 3,
      "CreatedBy": "arn:aws:iam::123456789102:root",
      "LaunchTemplateData": {
        "KeyName": "kp-us-east",
        "ImageId": "ami-6057e21a",
        "InstanceType": "t2.small",
        "NetworkInterfaces": [
          {
            "SubnetId": "subnet-7b16de0c",
            "DeviceIndex": 0,
            "Groups": [
              "sg-7c227019"
            ]
          }
        ]
      }
    }
  ]
}
```

```
    }
  ]
},
"DefaultVersion": false,
"CreateTime": "2017-11-20T13:19:54.000Z"
},
{
  "LaunchTemplateId": "lt-068f72b72934aff71",
  "LaunchTemplateName": "Webservers",
  "VersionNumber": 2,
  "CreatedBy": "arn:aws:iam::123456789102:root",
  "LaunchTemplateData": {
    "KeyName": "kp-us-east",
    "ImageId": "ami-6057e21a",
    "InstanceType": "t2.medium",
    "NetworkInterfaces": [
      {
        "SubnetId": "subnet-1a2b3c4d",
        "DeviceIndex": 0,
        "Groups": [
          "sg-7c227019"
        ]
      }
    ]
  }
},
"DefaultVersion": false,
"CreateTime": "2017-11-20T13:12:32.000Z"
},
{
  "LaunchTemplateId": "lt-068f72b72934aff71",
  "LaunchTemplateName": "Webservers",
  "VersionNumber": 1,
  "CreatedBy": "arn:aws:iam::123456789102:root",
  "LaunchTemplateData": {
    "UserData": "",
    "KeyName": "kp-us-east",
    "ImageId": "ami-aabbcc11",
    "InstanceType": "t2.medium",
    "NetworkInterfaces": [
      {
        "SubnetId": "subnet-7b16de0c",
        "DeviceIndex": 0,
        "DeleteOnTermination": false,
        "Groups": [
```

```
        "sg-7c227019"
      ],
      "AssociatePublicIpAddress": true
    }
  ]
},
"DefaultVersion": true,
"CreateTime": "2017-11-20T12:52:33.000Z"
}
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLaunchTemplateVersions](#)」の「」を参照してください。

describe-launch-templates

次のコード例は、describe-launch-templates を使用する方法を示しています。

AWS CLI

起動テンプレートを記述するには

この例では、起動テンプレートについて説明します。

コマンド:

```
aws ec2 describe-launch-templates
```

出力:

```
{
  "LaunchTemplates": [
    {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0e06d290751193123",
      "LaunchTemplateName": "TemplateForWebServer",
      "DefaultVersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "CreateTime": "2017-11-27T09:30:23.000Z"
    },
    {
      "LatestVersionNumber": 6,
```

```
    "LaunchTemplateId": "lt-0c45b5e061ec98456",
    "LaunchTemplateName": "DBServersTemplate",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-20T09:25:22.000Z"
  },
  {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0d47d774e8e52dabc",
    "LaunchTemplateName": "MyLaunchTemplate2",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-02T12:06:21.000Z"
  },
  {
    "LatestVersionNumber": 3,
    "LaunchTemplateId": "lt-01e5f948eb4f589d6",
    "LaunchTemplateName": "testingtemplate2",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:sts::123456789012:assumed-role/AdminRole/i-03ee35176e2e5aabc",
    "CreateTime": "2017-12-01T08:19:48.000Z"
  },
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLaunchTemplates](#)」の「」を参照してください。

describe-local-gateway-route-table-virtual-interface-group-associations

次のコード例は、describe-local-gateway-route-table-virtual-interface-group-associations を使用する方法を示しています。

AWS CLI

仮想インターフェイスグループとローカルゲートウェイルートテーブル間の関連付けを記述するには

次のdescribe-local-gateway-route-table-virtual-interface-group-associations例では、仮想インターフェイスグループと AWS アカウントのローカルゲートウェイルートテーブルの関連付けについて説明します。

```
aws ec2 describe-local-gateway-route-table-virtual-interface-group-associations
```

出力:

```
{
  "LocalGatewayRouteTableVirtualInterfaceGroupAssociations": [
    {
      "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-
grp-assoc-07145b276bEXAMPLE",
      "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
      "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:123456789012:local-
gateway-route-table/lgw-rtb-059615ef7dEXAMPLE",
      "OwnerId": "123456789012",
      "State": "associated",
      "Tags": []
    }
  ]
}
```

詳細については、[「Outposts ユーザーガイド」の「ローカルゲートウェイの使用AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations](#)」の「」を参照してください。

describe-local-gateway-route-table-vpc-associations

次のコード例は、describe-local-gateway-route-table-vpc-associations を使用する方法を示しています。

AWS CLI

VPCsとローカルゲートウェイルートテーブルの関連付けを記述するには

次のdescribe-local-gateway-route-table-vpc-associations例では、VPCsとローカルゲートウェイルートテーブル間の指定された関連付けに関する情報を表示します。

```
aws ec2 describe-local-gateway-route-table-vpc-associations \
```

```
--local-gateway-route-table-vpc-association-ids lgw-vpc-assoc-0e0f27af15EXAMPLE
```

出力:

```
{
  "LocalGatewayRouteTableVpcAssociation": {
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0e0f27af15EXAMPLE",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
    "VpcId": "vpc-0efe9bde08EXAMPLE",
    "State": "associated"
  }
}
```

詳細については、「Outposts ユーザーガイド」の「[ローカルゲートウェイテーブル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLocalGatewayRouteTableVpcAssociations](#)」の「」を参照してください。

describe-local-gateway-route-tables

次のコード例は、describe-local-gateway-route-tables を使用する方法を示しています。

AWS CLI

ローカルゲートウェイルートテーブルを記述するには

次のdescribe-local-gateway-route-tables例では、ローカルゲートウェイルートテーブルの詳細を表示します。

```
aws ec2 describe-local-gateway-route-tables
```

出力:

```
{
  "LocalGatewayRouteTables": [
    {
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",

```

```
    "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-0dc11b66edEXAMPLE",
    "State": "available"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLocalGatewayRouteTables](#)」の「」を参照してください。

describe-local-gateway-virtual-interface-groups

次のコード例は、describe-local-gateway-virtual-interface-groups を使用する方法を示しています。

AWS CLI

ローカルゲートウェイの仮想インターフェイスグループを記述するには

次のdescribe-local-gateway-virtual-interface-groups例では、AWS アカウントのローカルゲートウェイ仮想インターフェイスグループについて説明します。

```
aws ec2 describe-local-gateway-virtual-interface-groups
```

出力:

```
{
  "LocalGatewayVirtualInterfaceGroups": [
    {
      "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
      "LocalGatewayVirtualInterfaceIds": [
        "lgw-vif-01a23bc4d5EXAMPLE",
        "lgw-vif-543ab21012EXAMPLE"
      ],
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "OwnerId": "123456789012",
      "Tags": []
    }
  ]
}
```

詳細については、[「Outposts ユーザーガイド」の「ローカルゲートウェイの使用AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLocalGatewayVirtualInterfaceGroups](#)」の「」を参照してください。

describe-local-gateway-virtual-interfaces

次のコード例は、describe-local-gateway-virtual-interfaces を使用する方法を示しています。

AWS CLI

ローカルゲートウェイの仮想インターフェイスを記述するには

次のdescribe-local-gateway-virtual-interfaces例では、AWS アカウントのローカルゲートウェイ仮想インターフェイスについて説明します。

```
aws ec2 describe-local-gateway-virtual-interfaces
```

出力:

```
{
  "LocalGatewayVirtualInterfaces": [
    {
      "LocalGatewayVirtualInterfaceId": "lgw-vif-01a23bc4d5EXAMPLE",
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "Vlan": 2410,
      "LocalAddress": "0.0.0.0/0",
      "PeerAddress": "0.0.0.0/0",
      "LocalBgpAsn": 65010,
      "PeerBgpAsn": 65000,
      "OwnerId": "123456789012",
      "Tags": []
    },
    {
      "LocalGatewayVirtualInterfaceId": "lgw-vif-543ab21012EXAMPLE",
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "Vlan": 2410,
      "LocalAddress": "0.0.0.0/0",
      "PeerAddress": "0.0.0.0/0",
      "LocalBgpAsn": 65010,
```



```
        "PeerBgpAsn": 65000,  
        "OwnerId": "123456789012",  
        "Tags": []  
    }  
]  
}
```

詳細については、[「Outposts ユーザーガイド」の「ローカルゲートウェイの使用AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLocalGatewayVirtualInterfaces](#)」の「」を参照してください。

describe-local-gateways

次の例は、describe-local-gateways を使用する方法を説明しています。

AWS CLI

ローカルゲートウェイを記述するには

次のdescribe-local-gateways例では、使用可能なローカルゲートウェイの詳細を表示します。

```
aws ec2 describe-local-gateways
```

出力:

```
{  
  "LocalGateways": [  
    {  
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",  
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/  
op-0dc11b66ed59f995a",  
      "OwnerId": "123456789012",  
      "State": "available"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLocalGateways](#)」の「」を参照してください。

describe-locked-snapshots

次のコード例は、describe-locked-snapshots を使用する方法を示しています。

AWS CLI

スナップショットのロックステータスを記述するには

次のdescribe-locked-snapshots例では、指定されたスナップショットのロックステータスについて説明します。

```
aws ec2 describe-locked-snapshots \  
  --snapshot-ids snap-0b5e733b4a8df6e0d
```

出力:

```
{  
  "Snapshots": [  
    {  
      "OwnerId": "123456789012",  
      "SnapshotId": "snap-0b5e733b4a8df6e0d",  
      "LockState": "governance",  
      "LockDuration": 365,  
      "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",  
      "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00",  
      "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00"  
    }  
  ]  
}
```

詳細については、「Amazon ユーザーガイド」の「[スナップショットロック](#)」を参照してください。 EBS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLockedSnapshots](#)」の「」を参照してください。

describe-managed-prefix-lists

次の例は、describe-managed-prefix-lists を使用する方法を説明しています。

AWS CLI

マネージドプレフィックスリストを記述するには

次のdescribe-managed-prefix-lists例では、AWS アカウント が所有するプレフィックスリストについて説明します123456789012。

```
aws ec2 describe-managed-prefix-lists \  
--filters Name=owner-id,Values=123456789012
```

出力:

```
{  
  "PrefixLists": [  
    {  
      "PrefixListId": "pl-11223344556677aab",  
      "AddressFamily": "IPv6",  
      "State": "create-complete",  
      "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-11223344556677aab",  
      "PrefixListName": "vpc-ipv6-cidrs",  
      "MaxEntries": 25,  
      "Version": 1,  
      "Tags": [],  
      "OwnerId": "123456789012"  
    },  
    {  
      "PrefixListId": "pl-0123456abcabcabc1",  
      "AddressFamily": "IPv4",  
      "State": "active",  
      "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-0123456abcabcabc1",  
      "PrefixListName": "vpc-cidrs",  
      "MaxEntries": 10,  
      "Version": 1,  
      "Tags": [],  
      "OwnerId": "123456789012"  
    }  
  ]  
}
```

詳細については、「Amazon ユーザーガイド」の「[マネージドプレフィックスリスト](#)」を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeManagedPrefixLists](#)」の「」を参照してください。

describe-moving-addresses

次のコード例は、describe-moving-addresses を使用する方法を示しています。

AWS CLI

引越し先住所を記述するには

この例では、移動するすべての Elastic IP アドレスについて説明します。

コマンド:

```
aws ec2 describe-moving-addresses
```

出力:

```
{
  "MovingAddressStatuses": [
    {
      "PublicIp": "198.51.100.0",
      "MoveStatus": "MovingToVpc"
    }
  ]
}
```

この例では、EC2-VPC プラットフォームに移動しているすべてのアドレスについて説明します。

コマンド:

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeMovingAddresses](#)」の「」を参照してください。

describe-nat-gateways

次の例は、describe-nat-gateways を使用する方法を説明しています。

AWS CLI

例 1: パブリック NAT ゲートウェイを記述するには

次のdescribe-nat-gateways例では、指定されたパブリックNATゲートウェイについて説明します。

```
aws ec2 describe-nat-gateways \  
--nat-gateway-id nat-01234567890abcdef
```

出力:

```
{  
  "NatGateways": [  
    {  
      "CreateTime": "2023-08-25T01:56:51.000Z",  
      "NatGatewayAddresses": [  
        {  
          "AllocationId": "eipalloc-0790180cd2EXAMPLE",  
          "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
          "PrivateIp": "10.0.0.211",  
          "PublicIp": "54.85.121.213",  
          "AssociationId": "eipassoc-04d295cc9b8815b24",  
          "IsPrimary": true,  
          "Status": "succeeded"  
        },  
        {  
          "AllocationId": "eipalloc-0be6ecac95EXAMPLE",  
          "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
          "PrivateIp": "10.0.0.74",  
          "PublicIp": "3.211.231.218",  
          "AssociationId": "eipassoc-0f96bdca17EXAMPLE",  
          "IsPrimary": false,  
          "Status": "succeeded"  
        }  
      ],  
      "NatGatewayId": "nat-01234567890abcdef",  
      "State": "available",  
      "SubnetId": "subnet-655eab5f08EXAMPLE",  
      "VpcId": "vpc-098eb5ef58EXAMPLE",  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "public-nat"  
        }  
      ],  
      "ConnectivityType": "public"  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

例 2: プライベートNATゲートウェイを記述するには

次のdescribe-nat-gateways例では、指定されたプライベートNATゲートウェイについて説明します。

```
aws ec2 describe-nat-gateways \  
  --nat-gateway-id nat-1234567890abcdef0
```

出力:

```
{  
  "NatGateways": [  
    {  
      "CreateTime": "2023-08-25T00:50:05.000Z",  
      "NatGatewayAddresses": [  
        {  
          "NetworkInterfaceId": "eni-0065a61b324d1897a",  
          "PrivateIp": "10.0.20.240",  
          "IsPrimary": true,  
          "Status": "succeeded"  
        },  
        {  
          "NetworkInterfaceId": "eni-0065a61b324d1897a",  
          "PrivateIp": "10.0.20.33",  
          "IsPrimary": false,  
          "Status": "succeeded"  
        },  
        {  
          "NetworkInterfaceId": "eni-0065a61b324d1897a",  
          "PrivateIp": "10.0.20.197",  
          "IsPrimary": false,  
          "Status": "succeeded"  
        }  
      ],  
      "NatGatewayId": "nat-1234567890abcdef0",  
      "State": "available",  
      "SubnetId": "subnet-08fc749671EXAMPLE",  
      "VpcId": "vpc-098eb5ef58EXAMPLE",  
      "Tags": [  

```

```
        {
            "Key": "Name",
            "Value": "private-nat"
        }
    ],
    "ConnectivityType": "private"
}
]
```

詳細については、「Amazon VPCユーザーガイド [NAT](#)」の「[ゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeNatGateways](#)」の「」を参照してください。

describe-network-acls

次の例は、describe-network-acls を使用する方法を説明しています。

AWS CLI

ネットワークを記述するには ACLs

次のdescribe-network-acls例では、ネットワークの詳細を取得しますACLs。

```
aws ec2 describe-network-acls
```

出力:

```
{
  "NetworkAcls": [
    {
      "Associations": [
        {
          "NetworkAclAssociationId": "aclassoc-0c1679dc41EXAMPLE",
          "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
          "SubnetId": "subnet-0931fc2fa5EXAMPLE"
        }
      ],
      "Entries": [
        {
          "CidrBlock": "0.0.0.0/0",
```

```
        "Egress": true,
        "Protocol": "-1",
        "RuleAction": "allow",
        "RuleNumber": 100
    },
    {
        "CidrBlock": "0.0.0.0/0",
        "Egress": true,
        "Protocol": "-1",
        "RuleAction": "deny",
        "RuleNumber": 32767
    },
    {
        "CidrBlock": "0.0.0.0/0",
        "Egress": false,
        "Protocol": "-1",
        "RuleAction": "allow",
        "RuleNumber": 100
    },
    {
        "CidrBlock": "0.0.0.0/0",
        "Egress": false,
        "Protocol": "-1",
        "RuleAction": "deny",
        "RuleNumber": 32767
    }
],
"IsDefault": true,
"NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
"Tags": [],
"VpcId": "vpc-06e4ab6c6cEXAMPLE",
"OwnerId": "111122223333"
},
{
    "Associations": [],
    "Entries": [
        {
            "CidrBlock": "0.0.0.0/0",
            "Egress": true,
            "Protocol": "-1",
            "RuleAction": "allow",
            "RuleNumber": 100
        },
    ],
}
```



```
    "Egress": true,
    "Ipv6CidrBlock": "::/0",
    "Protocol": "-1",
    "RuleAction": "allow",
    "RuleNumber": 101
  },
  {
    "CidrBlock": "0.0.0.0/0",
    "Egress": true,
    "Protocol": "-1",
    "RuleAction": "deny",
    "RuleNumber": 32767
  },
  {
    "Egress": true,
    "Ipv6CidrBlock": "::/0",
    "Protocol": "-1",
    "RuleAction": "deny",
    "RuleNumber": 32768
  },
  {
    "CidrBlock": "0.0.0.0/0",
    "Egress": false,
    "Protocol": "-1",
    "RuleAction": "allow",
    "RuleNumber": 100
  },
  {
    "Egress": false,
    "Ipv6CidrBlock": "::/0",
    "Protocol": "-1",
    "RuleAction": "allow",
    "RuleNumber": 101
  },
  {
    "CidrBlock": "0.0.0.0/0",
    "Egress": false,
    "Protocol": "-1",
    "RuleAction": "deny",
    "RuleNumber": 32767
  },
  {
    "Egress": false,
    "Ipv6CidrBlock": "::/0",
```

```
        "Protocol": "-1",
        "RuleAction": "deny",
        "RuleNumber": 32768
      }
    ],
    "IsDefault": true,
    "NetworkAclId": "acl-0e2a78e4e2EXAMPLE",
    "Tags": [],
    "VpcId": "vpc-03914afb3eEXAMPLE",
    "OwnerId": "111122223333"
  }
]
```

詳細については、「ユーザーガイド」の「[ネットワークACLs](#)」を参照してください。AWS VPC

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeNetworkAcls](#)」の「」を参照してください。

describe-network-insights-access-scope-analyses

次の例は、describe-network-insights-access-scope-analyses を使用する方法を説明しています。

AWS CLI

Network Insights アクセス範囲分析を記述するには

次のdescribe-network-insights-access-scope-analyses例では、AWS アカウントのアクセス範囲分析について説明します。

```
aws ec2 describe-network-insights-access-scope-analyses \
  --region us-east-1
```

出力:

```
{
  "NetworkInsightsAccessScopeAnalyses": [
    {
      "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111",
```

```
    "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789111",
    "NetworkInsightsAccessScopeId": "nis-123456789222",
    "Status": "succeeded",
    "StartDate": "2022-01-25T19:45:36.842000+00:00",
    "FindingsFound": "true",
    "Tags": []
  }
]
}
```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeNetworkInsightsAccessScopeAnalyses](#)」の「」を参照してください。

describe-network-insights-access-scopes

次のコード例は、`describe-network-insights-access-scopes` を使用する方法を示しています。

AWS CLI

Network Insights アクセススコープを記述するには

次の `describe-network-insights-access-scopes` 例では、AWS アカウントのアクセス範囲分析について説明します。

```
aws ec2 describe-network-insights-access-scopes \
  --region us-east-1
```

出力:

```
{
  "NetworkInsightsAccessScopes": [
    {
      "NetworkInsightsAccessScopeId": "nis-123456789111",
      "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789111",
      "CreateDate": "2021-11-29T21:12:41.416000+00:00",
```

```

        "UpdatedDate": "2021-11-29T21:12:41.416000+00:00",
        "Tags": []
    }
]
}

```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeNetworkInsightsAccessScopes](#)」の「」を参照してください。

describe-network-insights-analyses

次の例は、describe-network-insights-analyses を使用する方法を説明しています。

AWS CLI

パス分析の結果を表示するには

次のdescribe-network-insights-analyses例では、指定された分析について説明します。この例では、送信元はインターネットゲートウェイ、送信先はEC2インスタンス、プロトコルはTCPです。分析は成功し (Status は succeeded)、パスに到達できません (NetworkPathFound は false)。説明コードENI_SG_RULES_MISMATCHは、インスタンスのセキュリティグループに、送信先ポートでのトラフィックを許可するルールが含まれていないことを示します。

```

aws ec2 describe-network-insights-analyses \
  --network-insights-analysis-ids nia-02207aa13eb480c7a

```

出力:

```

{
  "NetworkInsightsAnalyses": [
    {
      "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
      "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-insights-analysis/nia-02207aa13eb480c7a",
      "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
      "StartDate": "2021-01-20T22:58:37.495Z",
      "Status": "succeeded",

```

```
    "NetworkPathFound": false,
    "Explanations": [
      {
        "Direction": "ingress",
        "ExplanationCode": "ENI_SG_RULES_MISMATCH",
        "NetworkInterface": {
          "Id": "eni-0a25edef15a6cc08c",
          "Arn": "arn:aws:ec2:us-east-1:123456789012:network-
interface/eni-0a25edef15a6cc08c"
        },
        "SecurityGroups": [
          {
            "Id": "sg-02f0d35a850ba727f",
            "Arn": "arn:aws:ec2:us-east-1:123456789012:security-
group/sg-02f0d35a850ba727f"
          }
        ],
        "Subnet": {
          "Id": "subnet-004ff41eccb4d1194",
          "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
        },
        "Vpc": {
          "Id": "vpc-f1663d98ad28331c7",
          "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
        }
      }
    ],
    "Tags": []
  }
]
```

詳細については、「Reachability Analyzer ガイド」の「[の使用開始 AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNetworkInsightsAnalyses](#)」の「」を参照してください。

describe-network-insights-paths

次の例は、describe-network-insights-paths を使用する方法を説明しています。

AWS CLI

パスを記述するには

次のdescribe-network-insights-paths例では、指定されたパスについて説明します。

```
aws ec2 describe-network-insights-paths \  
  --network-insights-path-ids nip-0b26f224f1d131fa8
```

出力:

```
{  
  "NetworkInsightsPaths": [  
    {  
      "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",  
      "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-  
insights-path/nip-0b26f224f1d131fa8",  
      "CreateDate": "2021-01-20T22:43:46.933Z",  
      "Source": "igw-0797cccdc9d73b0e5",  
      "Destination": "i-0495d385ad28331c7",  
      "Protocol": "tcp"  
    }  
  ]  
}
```

詳細については、「Reachability Analyzer ガイド」の「[の使用開始 AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNetworkInsightsPaths](#)」の「」を参照してください。

describe-network-interface-attribute

次のコード例は、describe-network-interface-attribute を使用する方法を示しています。

AWS CLI

ネットワークインターフェイスのアタッチメント属性を記述するには

このコマンド例では、指定されたネットワークインターフェイスの attachment 属性について説明します。

コマンド:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute attachment
```

出力:

```
{  
  "NetworkInterfaceId": "eni-686ea200",  
  "Attachment": {  
    "Status": "attached",  
    "DeviceIndex": 0,  
    "AttachTime": "2015-05-21T20:02:20.000Z",  
    "InstanceId": "i-1234567890abcdef0",  
    "DeleteOnTermination": true,  
    "AttachmentId": "eni-attach-43348162",  
    "InstanceOwnerId": "123456789012"  
  }  
}
```

ネットワークインターフェイスの説明属性を記述するには

このコマンド例では、指定されたネットワークインターフェイスの `description` 属性について説明します。

コマンド:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute description
```

出力:

```
{  
  "NetworkInterfaceId": "eni-686ea200",  
  "Description": {  
    "Value": "My description"  
  }  
}
```

ネットワークインターフェイスの `groupSet` 属性を記述するには

このコマンド例では、指定されたネットワークインターフェイスの `groupSet` 属性について説明します。

コマンド:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --attribute groupSet
```

出力:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Groups": [
    {
      "GroupName": "my-security-group",
      "GroupId": "sg-903004f8"
    }
  ]
}
```

ネットワークインターフェイスの `sourceDestCheck` 属性を記述するには

このコマンド例では、指定されたネットワークインターフェイスの `sourceDestCheck` 属性について説明します。

コマンド:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --attribute sourceDestCheck
```

出力:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "SourceDestCheck": {
    "Value": true
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeNetworkInterfaceAttribute](#)」の「」を参照してください。

describe-network-interface-permissions

次のコード例は、describe-network-interface-permissions を使用する方法を示しています。

AWS CLI

ネットワークインターフェイスのアクセス許可を記述するには

この例では、すべてのネットワークインターフェイスのアクセス許可について説明します。

コマンド:

```
aws ec2 describe-network-interface-permissions
```

出力:

```
{
  "NetworkInterfacePermissions": [
    {
      "PermissionState": {
        "State": "GRANTED"
      },
      "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
      "NetworkInterfaceId": "eni-b909511a",
      "Permission": "INSTANCE-ATTACH",
      "AwsAccountId": "123456789012"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeNetworkInterfacePermissions](#)」の「」を参照してください。

describe-network-interfaces

次のコード例は、describe-network-interfaces を使用する方法を示しています。

AWS CLI

ネットワークインターフェイスを記述するには

この例では、すべてのネットワークインターフェイスについて説明します。

コマンド:

```
aws ec2 describe-network-interfaces
```

出力:

```
{
  "NetworkInterfaces": [
    {
      "Status": "in-use",
      "MacAddress": "02:2f:8f:b0:cf:75",
      "SourceDestCheck": true,
      "VpcId": "vpc-a01106c2",
      "Description": "my network interface",
      "Association": {
        "PublicIp": "203.0.113.12",
        "AssociationId": "eipassoc-0fbb766a",
        "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
        "IpOwnerId": "123456789012"
      },
      "NetworkInterfaceId": "eni-e5aa89a3",
      "PrivateIpAddresses": [
        {
          "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
          "Association": {
            "PublicIp": "203.0.113.12",
            "AssociationId": "eipassoc-0fbb766a",
            "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
            "IpOwnerId": "123456789012"
          },
          "Primary": true,
          "PrivateIpAddress": "10.0.1.17"
        }
      ],
      "RequesterManaged": false,
      "Ipv6Addresses": [],
      "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
      "AvailabilityZone": "us-east-1d",
      "Attachment": {
        "Status": "attached",
        "DeviceIndex": 1,
        "AttachTime": "2013-11-30T23:36:42.000Z",
        "InstanceId": "i-1234567890abcdef0",

```

```
    "DeleteOnTermination": false,
    "AttachmentId": "eni-attach-66c4350a",
    "InstanceOwnerId": "123456789012"
  },
  "Groups": [
    {
      "GroupName": "default",
      "GroupId": "sg-8637d3e3"
    }
  ],
  "SubnetId": "subnet-b61f49f0",
  "OwnerId": "123456789012",
  "TagSet": [],
  "PrivateIpAddress": "10.0.1.17"
},
{
  "Status": "in-use",
  "MacAddress": "02:58:f5:ef:4b:06",
  "SourceDestCheck": true,
  "VpcId": "vpc-a01106c2",
  "Description": "Primary network interface",
  "Association": {
    "PublicIp": "198.51.100.0",
    "IpOwnerId": "amazon"
  },
  "NetworkInterfaceId": "eni-f9ba99bf",
  "PrivateIpAddresses": [
    {
      "Association": {
        "PublicIp": "198.51.100.0",
        "IpOwnerId": "amazon"
      },
      "Primary": true,
      "PrivateIpAddress": "10.0.1.149"
    }
  ],
  "RequesterManaged": false,
  "Ipv6Addresses": [],
  "AvailabilityZone": "us-east-1d",
  "Attachment": {
    "Status": "attached",
    "DeviceIndex": 0,
    "AttachTime": "2013-11-30T23:35:33.000Z",
    "InstanceId": "i-0598c7d356eba48d7",
```

```

        "DeleteOnTermination": true,
        "AttachmentId": "eni-attach-1b9db777",
        "InstanceOwnerId": "123456789012"
    },
    "Groups": [
        {
            "GroupName": "default",
            "GroupId": "sg-8637d3e3"
        }
    ],
    "SubnetId": "subnet-b61f49f0",
    "OwnerId": "123456789012",
    "TagSet": [],
    "PrivateIpAddress": "10.0.1.149"
}
]
}

```

この例では、キーPurposeと値を持つタグを持つネットワークインターフェイスについて説明しますProd。

コマンド:

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

出力:

```

{
  "NetworkInterfaces": [
    {
      "Status": "available",
      "MacAddress": "12:2c:bd:f9:bf:17",
      "SourceDestCheck": true,
      "VpcId": "vpc-8941ebec",
      "Description": "ProdENI",
      "NetworkInterfaceId": "eni-b9a5ac93",
      "PrivateIpAddresses": [
        {
          "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
          "Primary": true,
          "PrivateIpAddress": "10.0.1.55"
        }
      ],
    }
  ],
}

```

```
    {
      "PrivateDnsName": "ip-10-0-1-117.ec2.internal",
      "Primary": false,
      "PrivateIpAddress": "10.0.1.117"
    }
  ],
  "RequesterManaged": false,
  "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
  "AvailabilityZone": "us-east-1d",
  "Ipv6Addresses": [],
  "Groups": [
    {
      "GroupName": "MySG",
      "GroupId": "sg-905002f5"
    }
  ],
  "SubnetId": "subnet-31d6c219",
  "OwnerId": "123456789012",
  "TagSet": [
    {
      "Value": "Prod",
      "Key": "Purpose"
    }
  ],
  "PrivateIpAddress": "10.0.1.55"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNetworkInterfaces](#)」の「」を参照してください。

describe-placement-groups

次の例は、describe-placement-groups を使用する方法を説明しています。

AWS CLI

プレイズメントグループを記述するには

この例では、すべてのプレイズメントグループについて説明します。

コマンド:

```
aws ec2 describe-placement-groups
```

出力:

```
{
  "PlacementGroups": [
    {
      "GroupName": "my-cluster",
      "State": "available",
      "Strategy": "cluster"
    },
    ...
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePlacementGroups](#)」の「」を参照してください。

describe-prefix-lists

次の例は、describe-prefix-lists を使用する方法を説明しています。

AWS CLI

プレフィックスリストを記述するには

この例では、リージョンで使用可能なすべてのプレフィックスリストを一覧表示します。

コマンド:

```
aws ec2 describe-prefix-lists
```

出力:

```
{
  "PrefixLists": [
    {
      "PrefixListName": "com.amazonaws.us-east-1.s3",
      "Cidrs": [
        "54.231.0.0/17"
      ],
      "PrefixListId": "pl-63a5400a"
    }
  ]
}
```

```
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePrefixLists](#)」の「」を参照してください。

describe-principal-id-format

次の例は、describe-principal-id-format を使用する方法を説明しています。

AWS CLI

ID 形式が長いIAMユーザーとロールの ID 形式を記述するには

次のdescribe-principal-id-format例では、ルートユーザー、すべてのIAMロール、および長い ID 形式が有効になっているすべてのIAMユーザーの ID 形式について説明します。

```
aws ec2 describe-principal-id-format \  
  --resource instance
```

出力:

```
{  
  "Principals": [  
    {  
      "Arn": "arn:aws:iam::123456789012:root",  
      "Statuses": [  
        {  
          "Deadline": "2016-12-15T00:00:00.000Z",  
          "Resource": "reservation",  
          "UseLongIds": true  
        },  
        {  
          "Deadline": "2016-12-15T00:00:00.000Z",  
          "Resource": "instance",  
          "UseLongIds": true  
        },  
        {  
          "Deadline": "2016-12-15T00:00:00.000Z",  
          "Resource": "volume",  
          "UseLongIds": true  
        }  
      ]  
    }  
  ]  
}
```

```
    },  
  ],  
  },  
  ...  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePrincipalIdFormat](#)」の「」を参照してください。

describe-public-ipv4-pools

次の例は、describe-public-ipv4-pools を使用する方法を説明しています。

AWS CLI

パブリックIPv4アドレスプールを記述するには

次のdescribe-public-ipv4-pools例では、Bring Your Own IP Addresses () を使用してパブリックIPv4アドレス範囲をプロビジョニングしたときに作成されたアドレスプールの詳細を表示しますBYOIP。

```
aws ec2 describe-public-ipv4-pools
```

出力:

```
{  
  "PublicIpv4Pools": [  
    {  
      "PoolId": "ipv4pool-ec2-1234567890abcdef0",  
      "PoolAddressRanges": [  
        {  
          "FirstAddress": "203.0.113.0",  
          "LastAddress": "203.0.113.255",  
          "AddressCount": 256,  
          "AvailableAddressCount": 256  
        }  
      ],  
      "TotalAddressCount": 256,  
      "TotalAvailableAddressCount": 256  
    }  
  ]  
}
```



```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス」の[DescribePublicIpv4Pools](#)を参照してください。

describe-regions

次のコード例は、describe-regions を使用する方法を示しています。

AWS CLI

例 1: 有効になっているすべてのリージョンを説明するには

次の describe-regions の例は、アカウントで有効なすべてのリージョンを説明しています。

```
aws ec2 describe-regions
```

出力:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-south-1.amazonaws.com",
      "RegionName": "ap-south-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-3.amazonaws.com",
      "RegionName": "eu-west-3",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-2.amazonaws.com",
      "RegionName": "eu-west-2",
      "OptInStatus": "opt-in-not-required"
    },
    {
```

```
    "Endpoint": "ec2.eu-west-1.amazonaws.com",
    "RegionName": "eu-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
    "RegionName": "ap-northeast-3",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
    "RegionName": "ap-northeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
    "RegionName": "ap-northeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.sa-east-1.amazonaws.com",
    "RegionName": "sa-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ca-central-1.amazonaws.com",
    "RegionName": "ca-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
    "RegionName": "ap-southeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
    "RegionName": "ap-southeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-central-1.amazonaws.com",
    "RegionName": "eu-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  },
```

```
{
  "Endpoint": "ec2.us-east-1.amazonaws.com",
  "RegionName": "us-east-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.us-east-2.amazonaws.com",
  "RegionName": "us-east-2",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.us-west-1.amazonaws.com",
  "RegionName": "us-west-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.us-west-2.amazonaws.com",
  "RegionName": "us-west-2",
  "OptInStatus": "opt-in-not-required"
}
]
```

詳細については、「Amazon EC2ユーザーガイド」の「[リージョンとゾーン](#)」を参照してください。

例 2: エンドポイント名に特定の文字列が含まれる有効なリージョンを説明するには

次の describe-regions の例では、エンドポイントに「us」という文字列が含まれる、有効にしたすべてのリージョンを説明しています。

```
aws ec2 describe-regions \
  --filters "Name=endpoint,Values=*us*"
```

出力:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.us-east-1.amazonaws.com",
      "RegionName": "us-east-1"
    },
    {
```

```
    "Endpoint": "ec2.us-east-2.amazonaws.com",
    "RegionName": "us-east-2"
  },
  {
    "Endpoint": "ec2.us-west-1.amazonaws.com",
    "RegionName": "us-west-1"
  },
  {
    "Endpoint": "ec2.us-west-2.amazonaws.com",
    "RegionName": "us-west-2"
  }
]
```

詳細については、「[Amazon EC2ユーザーガイド](#)」の「[リージョンとゾーン](#)」を参照してください。

例 3: すべてのリージョンを説明するには

次の describe-regions の例では、無効になっているリージョンを含め、使用可能なすべてのリージョンについて説明しています。

```
aws ec2 describe-regions \
  --all-regions
```

出力:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-south-1.amazonaws.com",
      "RegionName": "ap-south-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-3.amazonaws.com",
      "RegionName": "eu-west-3",
      "OptInStatus": "opt-in-not-required"
    }
  ]
}
```

```
  },
  {
    "Endpoint": "ec2.eu-west-2.amazonaws.com",
    "RegionName": "eu-west-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-west-1.amazonaws.com",
    "RegionName": "eu-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
    "RegionName": "ap-northeast-3",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.me-south-1.amazonaws.com",
    "RegionName": "me-south-1",
    "OptInStatus": "not-opted-in"
  },
  {
    "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
    "RegionName": "ap-northeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
    "RegionName": "ap-northeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.sa-east-1.amazonaws.com",
    "RegionName": "sa-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ca-central-1.amazonaws.com",
    "RegionName": "ca-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-east-1.amazonaws.com",
    "RegionName": "ap-east-1",
```

```
    "OptInStatus": "not-opted-in"
  },
  {
    "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
    "RegionName": "ap-southeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
    "RegionName": "ap-southeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-central-1.amazonaws.com",
    "RegionName": "eu-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-1.amazonaws.com",
    "RegionName": "us-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-2.amazonaws.com",
    "RegionName": "us-east-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-1.amazonaws.com",
    "RegionName": "us-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-2.amazonaws.com",
    "RegionName": "us-west-2",
    "OptInStatus": "opt-in-not-required"
  }
]
}
```

詳細については、「[Amazon EC2ユーザーガイド](#)」の「[リージョンとゾーン](#)」を参照してください。

例 4: リージョン名だけを一覧表示するには

次の describe-regions の例では、--query パラメータを使用して出力をフィルタリングし、リージョンの名前のみをテキストとして返します。

```
aws ec2 describe-regions \  
  --all-regions \  
  --query "Regions[].{Name:RegionName}" \  
  --output text
```

出力:

```
eu-north-1  
ap-south-1  
eu-west-3  
eu-west-2  
eu-west-1  
ap-northeast-3  
ap-northeast-2  
me-south-1  
ap-northeast-1  
sa-east-1  
ca-central-1  
ap-east-1  
ap-southeast-1  
ap-southeast-2  
eu-central-1  
us-east-1  
us-east-2  
us-west-1  
us-west-2
```

詳細については、[「Amazon ユーザーガイド」の「リージョンとゾーン」](#)を参照してください。

EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRegions](#)」の「」を参照してください。

describe-replace-root-volume-tasks

次の例は、describe-replace-root-volume-tasks を使用する方法を説明しています。

AWS CLI

例 1: 特定のルートボリューム置換タスクに関する情報を表示するには

次のdescribe-replace-root-volume-tasks例では、ルートボリューム置換タスクreplacevol-0111122223333abcd について説明します。

```
aws ec2 describe-replace-root-volume-tasks \  
  --replace-root-volume-task-ids replacevol-0111122223333abcd
```

出力:

```
{  
  "ReplaceRootVolumeTasks": [  
    {  
      "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",  
      "Tags": [],  
      "InstanceId": "i-0123456789abcdefa",  
      "TaskState": "succeeded",  
      "StartTime": "2022-03-14T15:16:28Z",  
      "CompleteTime": "2022-03-14T15:16:52Z"  
    }  
  ]  
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の [「ルートボリュームを置き換える」](#) を参照してください。

例 2: 特定のインスタンスのすべてのルートボリューム置換タスクに関する情報を表示するには

次のdescribe-replace-root-volume-tasks例では、インスタンス i-0123456789abcdefa のすべてのルートボリューム置換タスクについて説明します。

```
aws ec2 describe-replace-root-volume-tasks \  
  --filters Name=instance-id,Values=i-0123456789abcdefa
```

出力:

```
{  
  "ReplaceRootVolumeTasks": [  
    {
```



```
    "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
    "Tags": [],
    "InstanceId": "i-0123456789abcdefa",
    "TaskState": "succeeded",
    "StartTime": "2022-03-14T15:06:38Z",
    "CompleteTime": "2022-03-14T15:07:03Z"
  },
  {
    "ReplaceRootVolumeTaskId": "replacevol-0444455555555abcd",
    "Tags": [],
    "InstanceId": "i-0123456789abcdefa",
    "TaskState": "succeeded",
    "StartTime": "2022-03-14T15:16:28Z",
    "CompleteTime": "2022-03-14T15:16:52Z"
  }
]
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[ルートボリュームを置き換える](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReplaceRootVolumeTasks](#)」の「」を参照してください。

describe-reserved-instances-listings

次のコード例は、describe-reserved-instances-listings を使用する方法を示しています。

AWS CLI

リザーブドインスタンスのリストを記述するには

次のdescribe-reserved-instances-listings例では、指定されたリザーブドインスタンスのリストに関する情報を取得します。

```
aws ec2 describe-reserved-instances-listings \
  --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReservedInstancesListings](#)」の「」を参照してください。

describe-reserved-instances-modifications

次の例は、describe-reserved-instances-modifications を使用する方法を説明しています。

AWS CLI

リザーブドインスタンスの変更を記述するには

このコマンド例では、アカウントに対して送信されたすべてのリザーブドインスタンス変更リクエストについて説明します。

コマンド:

```
aws ec2 describe-reserved-instances-modifications
```

出力:

```
{
  "ReservedInstancesModifications": [
    {
      "Status": "fulfilled",
      "ModificationResults": [
        {
          "ReservedInstancesId": "93bbbca2-62f1-4d9d-b225-16bada29e6c7",
          "TargetConfiguration": {
            "AvailabilityZone": "us-east-1b",
            "InstanceType": "m1.large",
            "InstanceCount": 3
          }
        },
        {
          "ReservedInstancesId": "1ba8e2e3-aabb-46c3-bcf5-3fe2fda922e6",
          "TargetConfiguration": {
            "AvailabilityZone": "us-east-1d",
            "InstanceType": "m1.xlarge",
            "InstanceCount": 1
          }
        }
      ]
    },
    {
      "EffectiveDate": "2015-08-12T17:00:00.000Z",
      "CreateDate": "2015-08-12T17:52:52.630Z",
    }
  ]
}
```

```
    "UpdateDate": "2015-08-12T18:08:06.698Z",
    "ClientToken": "c9adb218-3222-4889-8216-0cf0e52dc37e:
    "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-
ab31-0f13aaf46687",
    "ReservedInstancesIds": [
      {
        "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342f5bd7c02"
      }
    ]
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeReservedInstancesModifications](#)」の「」を参照してください。

describe-reserved-instances-offerings

次のコード例は、describe-reserved-instances-offerings を使用する方法を示しています。

AWS CLI

リザーブドインスタンスのオファーを記述するには

このコマンド例では、リージョンで購入できるすべてのリザーブドインスタンスについて説明します。

コマンド:

```
aws ec2 describe-reserved-instances-offerings
```

出力:

```
{
  "ReservedInstancesOfferings": [
    {
      "OfferingType": "Partial Upfront",
      "AvailabilityZone": "us-east-1b",
      "InstanceTenancy": "default",
      "PricingDetails": [],
```

```
"ProductDescription": "Red Hat Enterprise Linux",
"UsagePrice": 0.0,
"RecurringCharges": [
  {
    "Amount": 0.088,
    "Frequency": "Hourly"
  }
],
"Marketplace": false,
"CurrencyCode": "USD",
"FixedPrice": 631.0,
"Duration": 94608000,
"ReservedInstancesOfferingId": "9a06095a-bdc6-47fe-a94a-2a382f016040",
"InstanceType": "c1.medium"
},
{
  "OfferingType": "PartialUpfront",
  "AvailabilityZone": "us-east-1b",
  "InstanceTenancy": "default",
  "PricingDetails": [],
  "ProductDescription": "Linux/UNIX",
  "UsagePrice": 0.0,
  "RecurringCharges": [
    {
      "Amount": 0.028,
      "Frequency": "Hourly"
    }
  ],
  "Marketplace": false,
  "CurrencyCode": "USD",
  "FixedPrice": 631.0,
  "Duration": 94608000,
  "ReservedInstancesOfferingId": "bfbefc6c-0d10-418d-b144-7258578d329d",
  "InstanceType": "c1.medium"
},
...
}
```

オプションを使用してリザーブドインスタンスのオファーを記述するには

この例では、t1.micro インスタンスタイプ、Windows (Amazon VPC) 製品、Heavy Utilization 製品など、AWS が提供するリザーブドインスタンスを一覧表示します。

コマンド:

```
aws ec2 describe-reserved-instances-offerings --no-include-marketplace --instance-type "t1.micro" --product-description "Windows (Amazon VPC)" --offering-type "no upfront"
```

出力:

```
{
  "ReservedInstancesOfferings": [
    {
      "OfferingType": "No Upfront",
      "AvailabilityZone": "us-east-1b",
      "InstanceTenancy": "default",
      "PricingDetails": [],
      "ProductDescription": "Windows",
      "UsagePrice": 0.0,
      "RecurringCharges": [
        {
          "Amount": 0.015,
          "Frequency": "Hourly"
        }
      ],
      "Marketplace": false,
      "CurrencyCode": "USD",
      "FixedPrice": 0.0,
      "Duration": 31536000,
      "ReservedInstancesOfferingId": "c48ab04c-fe69-4f94-8e39-a23842292823",
      "InstanceType": "t1.micro"
    },
    ...
    {
      "OfferingType": "No Upfront",
      "AvailabilityZone": "us-east-1d",
      "InstanceTenancy": "default",
      "PricingDetails": [],
      "ProductDescription": "Windows (Amazon VPC)",
      "UsagePrice": 0.0,
      "RecurringCharges": [
        {
          "Amount": 0.015,
          "Frequency": "Hourly"
        }
      ],
    },
  ],
}
```

```
    "Marketplace": false,  
    "CurrencyCode": "USD",  
    "FixedPrice": 0.0,  
    "Duration": 31536000,  
    "ReservedInstancesOfferingId": "3a98bf7d-2123-42d4-b4f5-8dbec4b06dc6",  
    "InstanceType": "t1.micro"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeReservedInstancesOfferings](#)」の「」を参照してください。

describe-reserved-instances

次のコード例は、describe-reserved-instances を使用する方法を示しています。

AWS CLI

リザーブドインスタンスを記述するには

この例では、所有しているリザーブドインスタンスについて説明します。

コマンド:

```
aws ec2 describe-reserved-instances
```

出力:

```
{  
  "ReservedInstances": [  
    {  
      "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342fexample",  
      "OfferingType": "No Upfront",  
      "AvailabilityZone": "us-west-1c",  
      "End": "2016-08-14T21:34:34.000Z",  
      "ProductDescription": "Linux/UNIX",  
      "UsagePrice": 0.00,  
      "RecurringCharges": [  
        {  
          "Amount": 0.104,  
          "Frequency": "Hourly"  
        }  
      ]  
    }  
  ]  
}
```

```

    ],
    "Start": "2015-08-15T21:34:35.086Z",
    "State": "active",
    "FixedPrice": 0.0,
    "CurrencyCode": "USD",
    "Duration": 31536000,
    "InstanceTenancy": "default",
    "InstanceType": "m3.medium",
    "InstanceCount": 2
  },
  ...
]
}

```

フィルターを使用してリザーブドインスタンスを記述するには

この例では、us-west-1c に 3 年間の t2.micro Linux/UNIX リザーブドインスタンスのみを含めるようにレスポンスをフィルタリングします。

コマンド:

```

aws ec2 describe-reserved-instances --
filters Name=duration,Values=94608000 Name=instance-
type,Values=t2.micro Name=product-description,Values=Linux/UNIX Name=availability-
zone,Values=us-east-1e

```

出力:

```

{
  "ReservedInstances": [
    {
      "ReservedInstancesId": "f127bd27-edb7-44c9-a0eb-0d7e09259af0",
      "OfferingType": "All Upfront",
      "AvailabilityZone": "us-east-1e",
      "End": "2018-03-26T21:34:34.000Z",
      "ProductDescription": "Linux/UNIX",
      "UsagePrice": 0.00,
      "RecurringCharges": [],
      "Start": "2015-03-27T21:34:35.848Z",
      "State": "active",
      "FixedPrice": 151.0,
      "CurrencyCode": "USD",
      "Duration": 94608000,

```

```
        "InstanceTenancy": "default",
        "InstanceType": "t2.micro",
        "InstanceCount": 1
    }
]
}
```

詳細については、AWS 「コマンドラインインターフェイスユーザーガイド」の「Amazon EC2 インスタンスの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeReservedInstances](#)」の「」を参照してください。

describe-route-tables

次のコード例は、describe-route-tables を使用する方法を示しています。

AWS CLI

ルートテーブルを記述するには

次のdescribe-route-tables例では、ルートテーブルの詳細を取得します。

```
aws ec2 describe-route-tables
```

出力:

```
{
  "RouteTables": [
    {
      "Associations": [
        {
          "Main": true,
          "RouteTableAssociationId": "rtbassoc-0df3f54e06EXAMPLE",
          "RouteTableId": "rtb-09ba434c1bEXAMPLE"
        }
      ],
      "PropagatingVgws": [],
      "RouteTableId": "rtb-09ba434c1bEXAMPLE",
      "Routes": [
        {
          "DestinationCidrBlock": "10.0.0.0/16",
          "GatewayId": "local",
```



```
        "Origin": "CreateRouteTable",
        "State": "active"
    },
    {
        "DestinationCidrBlock": "0.0.0.0/0",
        "NatGatewayId": "nat-06c018cbd8EXAMPLE",
        "Origin": "CreateRoute",
        "State": "blackhole"
    }
],
"Tags": [],
"VpcId": "vpc-0065acced4EXAMPLE",
"OwnerId": "111122223333"
},
{
    "Associations": [
        {
            "Main": true,
            "RouteTableAssociationId": "rtbassoc-9EXAMPLE",
            "RouteTableId": "rtb-a1eec7de"
        }
    ],
    "PropagatingVgws": [],
    "RouteTableId": "rtb-a1eec7de",
    "Routes": [
        {
            "DestinationCidrBlock": "172.31.0.0/16",
            "GatewayId": "local",
            "Origin": "CreateRouteTable",
            "State": "active"
        },
        {
            "DestinationCidrBlock": "0.0.0.0/0",
            "GatewayId": "igw-fEXAMPLE",
            "Origin": "CreateRoute",
            "State": "active"
        }
    ],
    "Tags": [],
    "VpcId": "vpc-3EXAMPLE",
    "OwnerId": "111122223333"
},
{
    "Associations": [
```

```
    {
      "Main": false,
      "RouteTableAssociationId": "rtbassoc-0b100c28b2EXAMPLE",
      "RouteTableId": "rtb-07a98f76e5EXAMPLE",
      "SubnetId": "subnet-0d3d002af8EXAMPLE"
    }
  ],
  "PropagatingVgws": [],
  "RouteTableId": "rtb-07a98f76e5EXAMPLE",
  "Routes": [
    {
      "DestinationCidrBlock": "10.0.0.0/16",
      "GatewayId": "local",
      "Origin": "CreateRouteTable",
      "State": "active"
    },
    {
      "DestinationCidrBlock": "0.0.0.0/0",
      "GatewayId": "igw-06cf664d80EXAMPLE",
      "Origin": "CreateRoute",
      "State": "active"
    }
  ],
  "Tags": [],
  "VpcId": "vpc-0065acced4EXAMPLE",
  "OwnerId": "111122223333"
}
]
```

詳細については、AWS VPC「ユーザーガイド」の「[ルートテーブルの使用](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeRouteTables](#)」の「」を参照してください。

describe-scheduled-instance-availability

次のコード例は、describe-scheduled-instance-availability を使用する方法を示しています。

AWS CLI

使用可能なスケジュールを記述するには

この例では、指定された日付から毎週日曜日に発生するスケジュールについて説明します。

コマンド:

```
aws ec2 describe-scheduled-instance-availability --  
recurrence Frequency=Weekly,Interval=1,OccurrenceDays=[1] --first-slot-start-time-  
range EarliestTime=2016-01-31T00:00:00Z,LatestTime=2016-01-31T04:00:00Z
```

出力:

```
{  
  "ScheduledInstanceAvailabilitySet": [  
    {  
      "AvailabilityZone": "us-west-2b",  
      "TotalScheduledInstanceHours": 1219,  
      "PurchaseToken": "eyJ2IjoiMSIsInMiOiJEsImMiOi...",  
      "MinTermDurationInDays": 366,  
      "AvailableInstanceCount": 20,  
      "Recurrence": {  
        "OccurrenceDaySet": [  
          1  
        ],  
        "Interval": 1,  
        "Frequency": "Weekly",  
        "OccurrenceRelativeToEnd": false  
      },  
      "Platform": "Linux/UNIX",  
      "FirstSlotStartTime": "2016-01-31T00:00:00Z",  
      "MaxTermDurationInDays": 366,  
      "SlotDurationInHours": 23,  
      "NetworkPlatform": "EC2-VPC",  
      "InstanceType": "c4.large",  
      "HourlyPrice": "0.095"  
    },  
    ...  
  ]  
}
```

結果を絞り込むには、オペレーティングシステム、ネットワーク、インスタンスタイプを指定するフィルターを追加できます。

コマンド:

```
--filters Name=platform,Values=Linux/UNIX Name=network-platform,Values=EC2-VPC  
Name=instance-type,Values=c4.large
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeScheduledInstanceAvailability](#)」の「」を参照してください。

describe-scheduled-instances

次の例は、describe-scheduled-instances を使用する方法を説明しています。

AWS CLI

スケジュールされたインスタンスを記述するには

この例では、指定されたスケジュールされたインスタンスについて説明します。

コマンド:

```
aws ec2 describe-scheduled-instances --scheduled-instance-  
ids sci-1234-1234-1234-1234-123456789012
```

出力:

```
{  
  "ScheduledInstanceSet": [  
    {  
      "AvailabilityZone": "us-west-2b",  
      "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",  
      "HourlyPrice": "0.095",  
      "CreateDate": "2016-01-25T21:43:38.612Z",  
      "Recurrence": {  
        "OccurrenceDaySet": [  
          1  
        ],  
        "Interval": 1,  
        "Frequency": "Weekly",  
        "OccurrenceRelativeToEnd": false,  
        "OccurrenceUnit": ""  
      }  
    }  
  ]  
}
```

```
    },
    "Platform": "Linux/UNIX",
    "TermEndDate": "2017-01-31T09:00:00Z",
    "InstanceCount": 1,
    "SlotDurationInHours": 32,
    "TermStartDate": "2016-01-31T09:00:00Z",
    "NetworkPlatform": "EC2-VPC",
    "TotalScheduledInstanceHours": 1696,
    "NextSlotStartTime": "2016-01-31T09:00:00Z",
    "InstanceType": "c4.large"
  }
]
}
```

この例では、すべてのスケジュールされたインスタンスについて説明します。

コマンド:

```
aws ec2 describe-scheduled-instances
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeScheduledInstances](#)」の「」を参照してください。

describe-security-group-references

次のコード例は、describe-security-group-references を使用する方法を示しています。

AWS CLI

セキュリティグループの参照を記述するには

この例では、のセキュリティグループリファレンスについて説明しますsg-bbbb2222。レスポンスは、セキュリティグループsg-bbbb2222が VPC のセキュリティグループによって参照されていることを示しますvpc-aaaaaaaa。

コマンド:

```
aws ec2 describe-security-group-references --group-id sg-bbbbb22222
```

出力:

```
{
```

```
"SecurityGroupsReferenceSet": [  
  {  
    "ReferencingVpcId": "vpc-aaaaaaaa ",  
    "GroupId": "sg-bbbbb22222",  
    "VpcPeeringConnectionId": "pcx-b04deed9"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeSecurityGroupReferences](#)」の「」を参照してください。

describe-security-group-rules

次のコード例は、describe-security-group-rules を使用する方法を示しています。

AWS CLI

例 1: セキュリティグループのセキュリティグループルールを記述するには

次のdescribe-security-group-rules例では、指定されたセキュリティグループのセキュリティグループルールについて説明します。filters オプションを使用して、結果を特定のセキュリティグループにスコープします。

```
aws ec2 describe-security-group-rules \  
  --filters Name="group-id",Values="sg-1234567890abcdef0"
```

出力:

```
{  
  "SecurityGroupRules": [  
    {  
      "SecurityGroupRuleId": "sgr-abcdef01234567890",  
      "GroupId": "sg-1234567890abcdef0",  
      "GroupOwnerId": "111122223333",  
      "IsEgress": false,  
      "IpProtocol": "-1",  
      "FromPort": -1,  
      "ToPort": -1,  
      "ReferencedGroupInfo": {  
        "GroupId": "sg-1234567890abcdef0",  
        "UserId": "111122223333"  
      }  
    }  
  ]  
}
```

```
    },
    "Tags": []
  },
  {
    "SecurityGroupId": "sg-1234567890abcdef0",
    "SecurityGroupRuleId": "sgr-bcdef01234567890a",
    "GroupOwnerId": "111122223333",
    "IsEgress": true,
    "IpProtocol": "-1",
    "FromPort": -1,
    "ToPort": -1,
    "CidrIpv6": "::/0",
    "Tags": []
  },
  {
    "SecurityGroupId": "sg-1234567890abcdef0",
    "SecurityGroupRuleId": "sgr-cdef01234567890ab",
    "GroupOwnerId": "111122223333",
    "IsEgress": true,
    "IpProtocol": "-1",
    "FromPort": -1,
    "ToPort": -1,
    "CidrIpv4": "0.0.0.0/0",
    "Tags": []
  }
]
}
```

例 2: セキュリティグループルールを記述するには

次のdescribe-security-group-rules例では、指定されたセキュリティグループルールについて説明します。

```
aws ec2 describe-security-group-rules \
  --security-group-rule-ids sgr-cdef01234567890ab
```

出力:

```
{
  "SecurityGroupRules": [
    {
      "SecurityGroupId": "sg-1234567890abcdef0",
      "SecurityGroupRuleId": "sgr-cdef01234567890ab",
      "GroupOwnerId": "111122223333",
      "IsEgress": true,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv4": "0.0.0.0/0",
      "Tags": []
    }
  ]
}
```

```
        "GroupOwnerId": "111122223333",
        "IsEgress": true,
        "IpProtocol": "-1",
        "FromPort": -1,
        "ToPort": -1,
        "CidrIpv4": "0.0.0.0/0",
        "Tags": []
    }
]
}
```

詳細については、「Amazon ユーザーガイド」の「[セキュリティグループルール](#)」を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSecurityGroupRules](#)」の「」を参照してください。

describe-security-groups

次の例は、describe-security-groups を使用する方法を説明しています。

AWS CLI

例 1: セキュリティグループを説明するには

次の describe-security-groups の例では、指定したセキュリティグループを示しています。

```
aws ec2 describe-security-groups \
  --group-ids sg-903004f8
```

出力:

```
{
  "SecurityGroups": [
    {
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ]
        }
      ]
    }
  ]
}
```



```
    ],
    "UserIdGroupPairs": [],
    "PrefixListIds": []
  }
],
"Description": "My security group",
"Tags": [
  {
    "Value": "SG1",
    "Key": "Name"
  }
],
"IpPermissions": [
  {
    "IpProtocol": "-1",
    "IpRanges": [],
    "UserIdGroupPairs": [
      {
        "UserId": "123456789012",
        "GroupId": "sg-903004f8"
      }
    ],
    "PrefixListIds": []
  },
  {
    "PrefixListIds": [],
    "FromPort": 22,
    "IpRanges": [
      {
        "Description": "Access from NY office",
        "CidrIp": "203.0.113.0/24"
      }
    ],
    "ToPort": 22,
    "IpProtocol": "tcp",
    "UserIdGroupPairs": []
  }
],
"GroupName": "MySecurityGroup",
"VpcId": "vpc-1a2b3c4d",
"OwnerId": "123456789012",
"GroupId": "sg-903004f8",
}
]
```

```
}
```

例 2: 特定のルールを持つセキュリティグループを説明するには

次のdescribe-security-groups例では、フィルターを使用して、SSHトラフィックを許可するルール (ポート 22) とすべてのアドレスからのトラフィックを許可するルール () を持つセキュリティグループに結果をスコープします0.0.0.0/0。例では、--query パラメータを使用してセキュリティグループの名前のみを表示しています。セキュリティグループが結果で返されるようにするには、すべてのフィルターに一致する必要があります。ただし、1つのルールがすべてのフィルターに一致する必要はありません。例えば、出力は、特定の IP アドレスからのSSHトラフィックを許可するルールと、すべてのアドレスからのHTTPトラフィックを許可する別のルールを持つセキュリティグループを返します。

```
aws ec2 describe-security-groups \  
  --filters Name=ip-permission.from-port,Values=22 Name=ip-permission.to-  
port,Values=22 Name=ip-permission.cidr,Values='0.0.0.0/0' \  
  --query "SecurityGroups[*].[GroupName]" \  
  --output text
```

出力:

```
default  
my-security-group  
web-servers  
launch-wizard-1
```

例 3: タグに基づいてセキュリティグループを説明するには

次の describe-security-groups の例では、フィルターを使用して、セキュリティグループ名に test が含まれ、タグ Test=To-delete が付けられているセキュリティグループに結果を絞り込みます。この例では、--queryパラメータを使用して、IDsセキュリティグループの名前とのみを表示します。

```
aws ec2 describe-security-groups \  
  --filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \  
  --query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

出力:

```
[
```

```
{
  "Name": "testfornewinstance",
  "ID": "sg-33bb22aa"
},
{
  "Name": "newgroupptest",
  "ID": "sg-1a2b3c4d"
}
]
```

タグフィルターを使用するその他の例については、「Amazon EC2ユーザーガイド」の「[タグの操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSecurityGroups](#)」の「」を参照してください。

describe-snapshot-attribute

次の例は、describe-snapshot-attribute を使用する方法を説明しています。

AWS CLI

スナップショットのスナップショット属性を記述するには

次のdescribe-snapshot-attribute例では、スナップショットを共有するアカウントを一覧表示します。

```
aws ec2 describe-snapshot-attribute \
  --snapshot-id snap-01234567890abcdef \
  --attribute createVolumePermission
```

出力:

```
{
  "SnapshotId": "snap-01234567890abcdef",
  "CreateVolumePermissions": [
    {
      "UserId": "123456789012"
    }
  ]
}
```

詳細については、[「Amazon Elastic Compute Cloud ユーザーガイド」の「Amazon EBSスナップショットの共有」](#)を参照してください。

- API 詳細については、「コマンドリファレンス[DescribeSnapshotAttribute](#)」の「」を参照してください。AWS CLI

describe-snapshot-tier-status

次の例は、describe-snapshot-tier-status を使用する方法を説明しています。

AWS CLI

アーカイブされたスナップショットに関するアーカイブ情報を表示するには

次のdescribe-snapshot-tier-status例は、アーカイブされたスナップショットに関するアーカイブ情報を示しています。

```
aws ec2 describe-snapshot-tier-status \  
  --filters "Name=snapshot-id, Values=snap-01234567890abcdef"
```

出力:

```
{  
  "SnapshotTierStatuses": [  
    {  
      "Status": "completed",  
      "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",  
      "LastTieringProgress": 100,  
      "Tags": [],  
      "VolumeId": "vol-01234567890abcdef",  
      "LastTieringOperationState": "archival-completed",  
      "StorageTier": "archive",  
      "OwnerId": "123456789012",  
      "SnapshotId": "snap-01234567890abcdef",  
      "LastTieringStartTime": "2021-09-15T16:44:37.574Z"  
    }  
  ]  
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の[「アーカイブされたスナップショットを表示する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSnapshotTierStatus](#)」の「」を参照してください。

describe-snapshots

次の例は、describe-snapshots を使用する方法を説明しています。

AWS CLI

例 1: スナップショットを説明するには

次の describe-snapshots の例では、指定したスナップショットを示しています。

```
aws ec2 describe-snapshots \  
  --snapshot-ids snap-1234567890abcdef0
```

出力:

```
{  
  "Snapshots": [  
    {  
      "Description": "This is my snapshot",  
      "Encrypted": false,  
      "VolumeId": "vol-049df61146c4d7901",  
      "State": "completed",  
      "VolumeSize": 8,  
      "StartTime": "2019-02-28T21:28:32.000Z",  
      "Progress": "100%",  
      "OwnerId": "012345678910",  
      "SnapshotId": "snap-01234567890abcdef",  
      "Tags": [  
        {  
          "Key": "Stack",  
          "Value": "test"  
        }  
      ]  
    }  
  ]  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EBSスナップショット」](#)を参照してください。 EC2

例 2: フィルターに基づいてスナップショットを説明するには

次のdescribe-snapshots例では、フィルターを使用して、pending状態にある AWS アカウントが所有するスナップショットに結果をスコープします。この例では、--queryパラメータを使用して、スナップショットIDsとスナップショットが開始された時刻のみを表示します。

```
aws ec2 describe-snapshots \  
  --owner-ids self \  
  --filters Name=status,Values=pending \  
  --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"
```

出力:

```
[  
  {  
    "ID": "snap-1234567890abcdef0",  
    "Time": "2019-08-04T12:48:18.000Z"  
  },  
  {  
    "ID": "snap-066877671789bd71b",  
    "Time": "2019-08-04T02:45:16.000Z"  
  },  
  ...  
]
```

次の describe-snapshots の例では、フィルターを使用して、指定したボリュームから作成されたスナップショットに結果を絞っています。この例では、--queryパラメータを使用してスナップショットのみを表示しますIDs。

```
aws ec2 describe-snapshots \  
  --filters Name=volume-id,Values=049df61146c4d7901 \  
  --query "Snapshots[*].[SnapshotId]" \  
  --output text
```

出力:

```
snap-1234567890abcdef0  
snap-08637175a712c3fb9  
...
```

フィルターを使用したその他の例については、Amazon EC2ユーザーガイドの[「リソースのリスト化とフィルタリング」](#)を参照してください。

例 3: タグに基づいてスナップショットを説明するには

次の describe-snapshots の例では、タグフィルターを使用して、結果の範囲をタグ Stack=Prod を含むスナップショットに限定しています。

```
aws ec2 describe-snapshots \  
  --filters Name=tag:Stack,Values=prod
```

describe-snapshots の出力例については、例 1 を参照してください。

タグフィルターを使用するその他の例については、「Amazon EC2ユーザーガイド」の[「タグの操作」](#)を参照してください。

例 4: 日付に基づいてスナップショットを説明するには

次の describe-snapshots 例では、JMESPath式を使用して、指定された日付より前に AWS アカウントによって作成されたすべてのスナップショットを記述します。スナップショットのみが表示されます IDs。

```
aws ec2 describe-snapshots \  
  --owner-ids 012345678910 \  
  --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

フィルターを使用したその他の例については、「Amazon EC2ユーザーガイド」の[「リソースの一覧表示とフィルタリング」](#)を参照してください。

例 5: アーカイブされたスナップショットのみを表示するには

次の describe-snapshots の例では、アーカイブ階層に保存されたスナップショットのみを説明しています。

```
aws ec2 describe-snapshots \  
  --filters "Name=storage-tier,Values=archive"
```

出力:

```
{
```

```
"Snapshots": [
  {
    "Description": "Snap A",
    "Encrypted": false,
    "VolumeId": "vol-01234567890aaaaaa",
    "State": "completed",
    "VolumeSize": 8,
    "StartTime": "2021-09-07T21:00:00.000Z",
    "Progress": "100%",
    "OwnerId": "123456789012",
    "SnapshotId": "snap-01234567890aaaaaa",
    "StorageTier": "archive",
    "Tags": []
  },
]
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[アーカイブされたスナップショットを表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSnapshots](#)」の「」を参照してください。

describe-spot-datafeed-subscription

次の例は、describe-spot-datafeed-subscription を使用する方法を説明しています。

AWS CLI

アカウントのスポットインスタンスデータフィードサブスクリプションを記述するにはこのコマンド例では、アカウントのデータフィードについて説明します。

コマンド:

```
aws ec2 describe-spot-datafeed-subscription
```

出力:

```
{
  "SpotDatafeedSubscription": {
    "OwnerId": "123456789012",
```



```
    "Prefix": "spotdata",
    "Bucket": "my-s3-bucket",
    "State": "Active"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSpotDatafeedSubscription](#)」の「」を参照してください。

describe-spot-fleet-instances

次のコード例は、describe-spot-fleet-instances を使用する方法を示しています。

AWS CLI

スポットフリートに関連付けられたスポットインスタンスを記述するには

この例では、指定されたスポットフリートに関連付けられたスポットインスタンスを一覧表示します。

コマンド:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

出力:

```
{
  "ActiveInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "InstanceType": "m3.medium",
      "SpotInstanceRequestId": "sir-08b93456"
    },
    ...
  ],
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSpotFleetInstances](#)」の「」を参照してください。

describe-spot-fleet-request-history

次のコード例は、describe-spot-fleet-request-history を使用する方法を示しています。

AWS CLI

スポットフリートの履歴を記述するには

このコマンド例では、指定された時刻から始まる、指定されたスポットフリートの履歴を返します。

コマンド:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-26T00:00:00Z
```

次の出力例は、スポットフリートの 2 つのスポットインスタンスの正常な起動を示しています。

出力:

```
{
  "HistoryRecords": [
    {
      "Timestamp": "2015-05-26T23:17:20.697Z",
      "EventInformation": {
        "EventSubType": "submitted"
      },
      "EventType": "fleetRequestChange"
    },
    {
      "Timestamp": "2015-05-26T23:17:20.873Z",
      "EventInformation": {
        "EventSubType": "active"
      },
      "EventType": "fleetRequestChange"
    },
    {
      "Timestamp": "2015-05-26T23:21:21.712Z",
      "EventInformation": {
        "InstanceId": "i-1234567890abcdef0",
        "EventSubType": "launched"
      },
      "EventType": "instanceChange"
    }
  ]
}
```

```
{
  "Timestamp": "2015-05-26T23:21:21.816Z",
  "EventInformation": {
    "InstanceId": "i-1234567890abcdef1",
    "EventSubType": "launched"
  },
  "EventType": "instanceChange"
}
],
"SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
"NextToken": "CpHNsscimcV5oH7bSbub03CI2Qms5+ypNpNm
+53MNlR0YcXAkp0xF1fKf91yVxSExmbtma3awYxMFzNA663ZskT0AHtJ6TCb2Z8bQC2EnZgyELbymtWPfpZ1ZbauVg
+P+TfG1WxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
"StartTime": "2015-05-26T00:00:00Z"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSpotFleetRequestHistory](#)」の「」を参照してください。

describe-spot-fleet-requests

次のコード例は、describe-spot-fleet-requests を使用する方法を示しています。

AWS CLI

スポットフリートリクエストを記述するには

この例では、すべてのスポットフリートリクエストについて説明します。

コマンド:

```
aws ec2 describe-spot-fleet-requests
```

出力:

```
{
  "SpotFleetRequestConfigs": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "SpotFleetRequestConfig": {
        "TargetCapacity": 20,
        "LaunchSpecifications": [
          {
```

```
    "EbsOptimized": false,
    "NetworkInterfaces": [
      {
        "SubnetId": "subnet-a61dafcf",
        "DeviceIndex": 0,
        "DeleteOnTermination": false,
        "AssociatePublicIpAddress": true,
        "SecondaryPrivateIpAddressCount": 0
      }
    ],
    "InstanceType": "cc2.8xlarge",
    "ImageId": "ami-1a2b3c4d"
  },
  {
    "EbsOptimized": false,
    "NetworkInterfaces": [
      {
        "SubnetId": "subnet-a61dafcf",
        "DeviceIndex": 0,
        "DeleteOnTermination": false,
        "AssociatePublicIpAddress": true,
        "SecondaryPrivateIpAddressCount": 0
      }
    ],
    "InstanceType": "r3.8xlarge",
    "ImageId": "ami-1a2b3c4d"
  }
],
"SpotPrice": "0.05",
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
},
"SpotFleetRequestState": "active"
},
{
  "SpotFleetRequestId": "sfr-306341ed-9739-402e-881b-ce47bEXAMPLE",
  "SpotFleetRequestConfig": {
    "TargetCapacity": 20,
    "LaunchSpecifications": [
      {
        "EbsOptimized": false,
        "NetworkInterfaces": [
          {
            "SubnetId": "subnet-6e7f829e",
            "DeviceIndex": 0,
```

```

        "DeleteOnTermination": false,
        "AssociatePublicIpAddress": true,
        "SecondaryPrivateIpAddressCount": 0
      }
    ],
    "InstanceType": "m3.medium",
    "ImageId": "ami-1a2b3c4d"
  }
],
"SpotPrice": "0.05",
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
},
"SpotFleetRequestState": "active"
}
]
}

```

スポットフリートリクエストを記述するには

この例では、指定されたスポットフリートリクエストについて説明します。

コマンド:

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

出力:

```

{
  "SpotFleetRequestConfigs": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "SpotFleetRequestConfig": {
        "TargetCapacity": 20,
        "LaunchSpecifications": [
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,

```

```
        "AssociatePublicIpAddress": true,
        "SecondaryPrivateIpAddressCount": 0
      }
    ],
    "InstanceType": "cc2.8xlarge",
    "ImageId": "ami-1a2b3c4d"
  },
  {
    "EbsOptimized": false,
    "NetworkInterfaces": [
      {
        "SubnetId": "subnet-a61dafcf",
        "DeviceIndex": 0,
        "DeleteOnTermination": false,
        "AssociatePublicIpAddress": true,
        "SecondaryPrivateIpAddressCount": 0
      }
    ],
    "InstanceType": "r3.8xlarge",
    "ImageId": "ami-1a2b3c4d"
  }
],
"SpotPrice": "0.05",
"IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
},
"SpotFleetRequestState": "active"
}
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSpotFleetRequests](#)」の「」を参照してください。

describe-spot-instance-requests

次のコード例は、describe-spot-instance-requests を使用する方法を示しています。

AWS CLI

例 1: スポットインスタンスリクエストを記述するには

次のdescribe-spot-instance-requests例では、指定されたスポットインスタンスリクエストについて説明します。

```
aws ec2 describe-spot-instance-requests \  
--spot-instance-request-ids sir-08b93456
```

出力:

```
{  
  "SpotInstanceRequests": [  
    {  
      "CreateTime": "2018-04-30T18:14:55.000Z",  
      "InstanceId": "i-1234567890abcdef1",  
      "LaunchSpecification": {  
        "InstanceType": "t2.micro",  
        "ImageId": "ami-003634241a8fcdec0",  
        "KeyName": "my-key-pair",  
        "SecurityGroups": [  
          {  
            "GroupName": "default",  
            "GroupId": "sg-e38f24a7"  
          }  
        ],  
        "BlockDeviceMappings": [  
          {  
            "DeviceName": "/dev/sda1",  
            "Ebs": {  
              "DeleteOnTermination": true,  
              "SnapshotId": "snap-0e54a519c999adbbd",  
              "VolumeSize": 8,  
              "VolumeType": "standard",  
              "Encrypted": false  
            }  
          }  
        ],  
        "NetworkInterfaces": [  
          {  
            "DeleteOnTermination": true,  
            "DeviceIndex": 0,  
            "SubnetId": "subnet-049df61146c4d7901"  
          }  
        ],  
        "Placement": {  
          "AvailabilityZone": "us-east-2b",  
          "Tenancy": "default"  
        }  
      }  
    ]  
  }  
}
```

```

        "Monitoring": {
            "Enabled": false
        },
        "LaunchedAvailabilityZone": "us-east-2b",
        "ProductDescription": "Linux/UNIX",
        "SpotInstanceRequestId": "sir-08b93456",
        "SpotPrice": "0.010000",
        "State": "active",
        "Status": {
            "Code": "fulfilled",
            "Message": "Your Spot request is fulfilled.",
            "UpdateTime": "2018-04-30T18:16:21.000Z"
        },
        "Tags": [],
        "Type": "one-time",
        "InstanceInterruptionBehavior": "terminate"
    }
]
}

```

例 2: フィルターに基づいてスポットインスタンスリクエストを記述するには

次のdescribe-spot-instance-requests例では、フィルターを使用して、指定されたアベイラビリティゾーンで指定されたインスタンスタイプを持つスポットインスタンスリクエストに結果をスコープします。この例では、--queryパラメータを使用してインスタンスのみを表示しますIDs。

```

aws ec2 describe-spot-instance-requests \
  --filters Name=launch.instance-type,Values=m3.medium Name=launched-availability-zone,Values=us-east-2a \
  --query "SpotInstanceRequests[*].[InstanceId]" \
  --output text

```

出力:

```

i-057750d42936e468a
i-001efd250faaa6ffa
i-027552a73f021f3bd
...

```


フィルターを使用したその他の例については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[リソースの一覧表示とフィルタリング](#)」を参照してください。

例 3: タグに基づいてスポットインスタンスリクエストを記述するには

次のdescribe-spot-instance-requests例では、タグフィルターを使用して、タグを持つスポットインスタンスリクエストに結果をスコープしますcost-center=cc123。

```
aws ec2 describe-spot-instance-requests \  
  --filters Name=tag:cost-center,Values=cc123
```

describe-spot-instance-requests の出力例については、例 1 を参照してください。

タグフィルターを使用するその他の例については、「Amazon EC2ユーザーガイド」の「[タグの操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSpotInstanceRequests](#)」の「」を参照してください。

describe-spot-price-history

次のコード例は、describe-spot-price-history を使用する方法を示しています。

AWS CLI

スポット料金履歴を記述するには

この例では、1月の特定の日の m1.xlarge インスタンスのスポット料金履歴を返します。

コマンド:

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --start-  
time 2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

出力:

```
{  
  "SpotPriceHistory": [  
    {  
      "Timestamp": "2014-01-06T07:10:55.000Z",  
      "ProductDescription": "SUSE Linux",
```

```

    "InstanceType": "m1.xlarge",
    "SpotPrice": "0.087000",
    "AvailabilityZone": "us-west-1b"
  },
  {
    "Timestamp": "2014-01-06T07:10:55.000Z",
    "ProductDescription": "SUSE Linux",
    "InstanceType": "m1.xlarge",
    "SpotPrice": "0.087000",
    "AvailabilityZone": "us-west-1c"
  },
  {
    "Timestamp": "2014-01-06T05:42:36.000Z",
    "ProductDescription": "SUSE Linux (Amazon VPC)",
    "InstanceType": "m1.xlarge",
    "SpotPrice": "0.087000",
    "AvailabilityZone": "us-west-1a"
  },
  ...
}

```

Linux/UNIX Amazon のスポット料金履歴を記述するには VPC

この例では、1月の特定の日の m1.xlarge、Linux/UNIX Amazon VPC インスタンスのスポット料金履歴を返します。

コマンド:

```

aws ec2 describe-spot-price-history --instance-types m1.xlarge --product-
description "Linux/UNIX (Amazon VPC)" --start-time 2014-01-06T07:08:09 --end-
time 2014-01-06T08:09:10

```

出力:

```

{
  "SpotPriceHistory": [
    {
      "Timestamp": "2014-01-06T04:32:53.000Z",
      "ProductDescription": "Linux/UNIX (Amazon VPC)",
      "InstanceType": "m1.xlarge",
      "SpotPrice": "0.080000",
      "AvailabilityZone": "us-west-1a"
    },

```

```
{
  "Timestamp": "2014-01-05T11:28:26.000Z",
  "ProductDescription": "Linux/UNIX (Amazon VPC)",
  "InstanceType": "m1.xlarge",
  "SpotPrice": "0.080000",
  "AvailabilityZone": "us-west-1c"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSpotPriceHistory](#)」の「」を参照してください。

describe-stale-security-groups

次のコード例は、describe-stale-security-groups を使用する方法を示しています。

AWS CLI

古いセキュリティグループを記述するには

この例では、の古いセキュリティグループルールについて説明します vpc-11223344。このレスポンスは、アカウントの sg-5fa68d3a にピア sg-279ab042 で参照する古い進入 SSH ルールがあり VPC、アカウントの sg-fe6fba9a にピア sg-ef6fba8b で参照する古い進入 SSH ルールがあることを示しています VPC。

コマンド:

```
aws ec2 describe-stale-security-groups --vpc-id vpc-11223344
```

出力:

```
{
  "StaleSecurityGroupSet": [
    {
      "VpcId": "vpc-11223344",
      "StaleIpPermissionsEgress": [
        {
          "ToPort": 22,
          "FromPort": 22,
          "UserIdGroupPairs": [
```

```
        {
            "VpcId": "vpc-7a20e51f",
            "GroupId": "sg-ef6fba8b",
            "VpcPeeringConnectionId": "pcx-b04deed9",
            "PeeringStatus": "active"
        }
    ],
    "IpProtocol": "tcp"
}
],
"GroupName": "MySG1",
"StaleIpPermissions": [],
"GroupId": "sg-fe6fba9a",
>Description": "MySG1"
},
{
    "VpcId": "vpc-11223344",
    "StaleIpPermissionsEgress": [],
    "GroupName": "MySG2",
    "StaleIpPermissions": [
        {
            "ToPort": 22,
            "FromPort": 22,
            "UserIdGroupPairs": [
                {
                    "VpcId": "vpc-7a20e51f",
                    "GroupId": "sg-279ab042",
                    "Description": "Access from pcx-b04deed9",
                    "VpcPeeringConnectionId": "pcx-b04deed9",
                    "PeeringStatus": "active"
                }
            ],
            "IpProtocol": "tcp"
        }
    ],
    "GroupId": "sg-5fa68d3a",
    "Description": "MySG2"
}
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStaleSecurityGroups](#)」の「」を参照してください。

describe-store-image-tasks

次の例は、describe-store-image-tasks を使用する方法を説明しています。

AWS CLI

AMIストアタスクの進行状況を記述するには

次のdescribe-store-image-tasks例では、AMIストアタスクの進行状況について説明します。

```
aws ec2 describe-store-image-tasks
```

出力:

```
{
  "StoreImageTaskResults": [
    {
      "AmiId": "ami-1234567890abcdef0",
      "Bucket": "my-ami-bucket",
      "ProgressPercentage": 17,
      "S3objectKey": "ami-1234567890abcdef0.bin",
      "StoreTaskState": "InProgress",
      "StoreTaskFailureReason": null,
      "TaskStartTime": "2022-01-01T01:01:01.001Z"
    }
  ]
}
```

S3 AMIを使用した の保存と復元の詳細については、「Amazon EC2ユーザーガイド」のS3 <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-store-restore.html>> AMIを使用した の保存と復元」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStoreImageTasks](#)」の「」を参照してください。

describe-subnets

次のコード例は、describe-subnets を使用する方法を示しています。

AWS CLI

例 1: すべてのサブネットを説明するには

次の describe-subnets の例では、サブネットの詳細を示します。

```
aws ec2 describe-subnets
```

出力:

```
{
  "Subnets": [
    {
      "AvailabilityZone": "us-east-1d",
      "AvailabilityZoneId": "use1-az2",
      "AvailableIpAddressCount": 4089,
      "CidrBlock": "172.31.80.0/20",
      "DefaultForAz": true,
      "MapPublicIpOnLaunch": false,
      "MapCustomerOwnedIpOnLaunch": true,
      "State": "available",
      "SubnetId": "subnet-0bb1c79de3EXAMPLE",
      "VpcId": "vpc-0ee975135dEXAMPLE",
      "OwnerId": "111122223333",
      "AssignIpv6AddressOnCreation": false,
      "Ipv6CidrBlockAssociationSet": [],
      "CustomerOwnedIpv4Pool": "pool-2EXAMPLE",
      "SubnetArn": "arn:aws:ec2:us-east-2:111122223333:subnet/
subnet-0bb1c79de3EXAMPLE",
      "EnableDns64": false,
      "Ipv6Native": false,
      "PrivateDnsNameOptionsOnLaunch": {
        "HostnameType": "ip-name",
        "EnableResourceNameDnsARecord": false,
        "EnableResourceNameDnsAAAARecord": false
      }
    },
    {
      "AvailabilityZone": "us-east-1d",
      "AvailabilityZoneId": "use1-az2",
      "AvailableIpAddressCount": 4089,
      "CidrBlock": "172.31.80.0/20",
      "DefaultForAz": true,
```

```

    "MapPublicIpOnLaunch": true,
    "MapCustomerOwnedIpOnLaunch": false,
    "State": "available",
    "SubnetId": "subnet-8EXAMPLE",
    "VpcId": "vpc-3EXAMPLE",
    "OwnerId": "1111222233333",
    "AssignIpv6AddressOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "MySubnet"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
      "HostnameType": "ip-name",
      "EnableResourceNameDnsARecord": false,
      "EnableResourceNameDnsAAAARecord": false
    }
  }
]
}

```

詳細については、AWS VPC「ユーザーガイド」の「[VPCsおよびサブネットの使用](#)」を参照してください。

例 2: 特定の のサブネットを記述するには VPC

次のdescribe-subnets例では、フィルターを使用して、指定された のサブネットの詳細を取得しますVPC。

```

aws ec2 describe-subnets \
  --filters "Name=vpc-id,Values=vpc-3EXAMPLE"

```

出力:

```

{
  "Subnets": [
    {

```

```

    "AvailabilityZone": "us-east-1d",
    "AvailabilityZoneId": "use1-az2",
    "AvailableIpAddressCount": 4089,
    "CidrBlock": "172.31.80.0/20",
    "DefaultForAz": true,
    "MapPublicIpOnLaunch": true,
    "MapCustomerOwnedIpOnLaunch": false,
    "State": "available",
    "SubnetId": "subnet-8EXAMPLE",
    "VpcId": "vpc-3EXAMPLE",
    "OwnerId": "111122223333",
    "AssignIpv6AddressOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "MySubnet"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
      "HostnameType": "ip-name",
      "EnableResourceNameDnsARecord": false,
      "EnableResourceNameDnsAAAARecord": false
    }
  }
]
}

```

詳細については、AWS VPC「ユーザーガイド」の「[VPCsおよびサブネットの使用](#)」を参照してください。

例 3: 特定のタグを持つサブネットを説明するには

次のdescribe-subnets例では、フィルターを使用して、タグを持つサブネットの詳細を取得しCostCenter=123、--queryパラメータを使用して、このタグを持つサブネットIDsのサブネットを表示します。

```

aws ec2 describe-subnets \
  --filters "Name=tag:CostCenter,Values=123" \

```



```
--query "Subnets[*].SubnetId" \  
--output text
```

出力:

```
subnet-0987a87c8b37348ef  
subnet-02a95061c45f372ee  
subnet-03f720e7de2788d73
```

詳細については、「Amazon VPCユーザーガイド」の「[VPCsとサブネットの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSubnets](#)」の「」を参照してください。

describe-tags

次のコード例は、describe-tags を使用する方法を示しています。

AWS CLI

例 1: 1 つのリソースのすべてのタグを記述するには

次のdescribe-tags例では、指定されたインスタンスのタグについて説明します。

```
aws ec2 describe-tags \  
--filters "Name=resource-id,Values=i-1234567890abcdef8"
```

出力:

```
{  
  "Tags": [  
    {  
      "ResourceType": "instance",  
      "ResourceId": "i-1234567890abcdef8",  
      "Value": "Test",  
      "Key": "Stack"  
    },  
    {  
      "ResourceType": "instance",  
      "ResourceId": "i-1234567890abcdef8",  
      "Value": "Beta Server",
```

```
        "Key": "Name"
      }
    ]
  }
```

例 2: リソースタイプのすべてのタグを記述するには

次のdescribe-tags例では、ボリュームのタグについて説明します。

```
aws ec2 describe-tags \
  --filters "Name=resource-type,Values=volume"
```

出力:

```
{
  "Tags": [
    {
      "ResourceType": "volume",
      "ResourceId": "vol-1234567890abcdef0",
      "Value": "Project1",
      "Key": "Purpose"
    },
    {
      "ResourceType": "volume",
      "ResourceId": "vol-049df61146c4d7901",
      "Value": "Logs",
      "Key": "Purpose"
    }
  ]
}
```

例 3: すべてのタグを記述するには

次のdescribe-tags例では、すべてのリソースのタグについて説明します。

```
aws ec2 describe-tags
```

例 4: タグキーに基づいてリソースのタグを記述するには

次のdescribe-tags例では、キー のタグを持つリソースのタグについて説明しますStack。

```
aws ec2 describe-tags \
```

```
--filters Name=key,Values=Stack
```

出力:

```
{
  "Tags": [
    {
      "ResourceType": "volume",
      "ResourceId": "vol-027552a73f021f3b",
      "Value": "Production",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",
      "Key": "Stack"
    }
  ]
}
```

例 5: タグキーとタグ値に基づいてリソースのタグを記述するには

次のdescribe-tags例では、タグを持つリソースのタグについて説明しますStack=Test。

```
aws ec2 describe-tags \  
--filters Name=key,Values=Stack Name=value,Values=Test
```

出力:

```
{
  "Tags": [
    {
      "ResourceType": "image",
      "ResourceId": "ami-3ac336533f021f3bd",
      "Value": "Test",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",

```

```

        "Key": "Stack"
      }
    ]
  }

```

次のdescribe-tags例では、代替構文を使用して、タグ でリソースを記述しますStack=Test。

```

aws ec2 describe-tags \
  --filters "Name=tag:Stack,Values=Test"

```

次のdescribe-tags例では、キーと 値Purposeのないタグを持つすべてのインスタスのタグについて説明します。

```

aws ec2 describe-tags \
  --filters "Name=resource-
type,Values=instance" "Name=key,Values=Purpose" "Name=value,Values="

```

出力:

```

{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef5",
      "Value": null,
      "Key": "Purpose"
    }
  ]
}

```

- API 詳細については、AWS CLI 「 コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

describe-traffic-mirror-filters

次の例は、describe-traffic-mirror-filters を使用する方法を説明しています。

AWS CLI

トラフィックミラーフィルターを表示するには

次のdescribe-traffic-mirror-filters例では、すべてのトラフィックミラーフィルターの詳細を表示します。

```
aws ec2 describe-traffic-mirror-filters
```

出力:

```
{
  "TrafficMirrorFilters": [
    {
      "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
      "IngressFilterRules": [
        {
          "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",
          "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",
          "TrafficDirection": "ingress",
          "RuleNumber": 100,
          "RuleAction": "accept",
          "Protocol": 6,
          "DestinationCidrBlock": "10.0.0.0/24",
          "SourceCidrBlock": "10.0.0.0/24",
          "Description": "TCP Rule"
        }
      ],
      "EgressFilterRules": [],
      "NetworkServices": [],
      "Description": "Example filter",
      "Tags": []
    }
  ]
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーフィルターを表示する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrafficMirrorFilters](#)」の「」を参照してください。

describe-traffic-mirror-sessions

次の例は、describe-traffic-mirror-sessions を使用する方法を説明しています。

AWS CLI

トラフィックミラーセッションを記述するには

次のdescribe-traffic-mirror-sessions例では、トラフィックミラーセッションの詳細を表示します。

```
aws ec2 describe-traffic-mirror-sessions
```

出力:

```
{
  "TrafficMirrorSessions": [
    {
      "Tags": [],
      "VirtualNetworkId": 42,
      "OwnerId": "111122223333",
      "Description": "TCP Session",
      "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
      "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",
      "TrafficMirrorFilterId": "tmf-083e18f985EXAMPLE",
      "PacketLength": 20,
      "SessionNumber": 1,
      "TrafficMirrorSessionId": "tms-0567a4c684EXAMPLE"
    },
    {
      "Tags": [
        {
          "Key": "Name",
          "Value": "tag test"
        }
      ],
      "VirtualNetworkId": 13314501,
      "OwnerId": "111122223333",
      "Description": "TCP Session",
      "NetworkInterfaceId": "eni-0a471a5cf3EXAMPLE",
      "TrafficMirrorTargetId": "tmt-03665551cbEXAMPLE",
      "TrafficMirrorFilterId": "tmf-06c787846cEXAMPLE",
      "SessionNumber": 2,
      "TrafficMirrorSessionId": "tms-0060101cf8EXAMPLE"
    }
  ]
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーセッションの詳細を表示するAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrafficMirrorSessions](#)」の「」を参照してください。

describe-traffic-mirror-targets

次のコード例は、describe-traffic-mirror-targets を使用する方法を示しています。

AWS CLI

トラフィックミラーターゲットを記述するには

次のdescribe-traffic-mirror-targets例では、指定されたトラフィックミラーターゲットに関する情報を表示します。

```
aws ec2 describe-traffic-mirror-targets \  
  --traffic-mirror-target-ids tmt-0dabe9b0a6EXAMPLE
```

出力:

```
{  
  "TrafficMirrorTargets": [  
    {  
      "TrafficMirrorTargetId": "tmt-0dabe9b0a6EXAMPLE",  
      "NetworkLoadBalancerArn": "arn:aws:elasticloadbalancing:us-  
east-1:111122223333:loadbalancer/net/NLB/7cdec873fEXAMPLE",  
      "Type": "network-load-balancer",  
      "Description": "Example Network Load Balancer target",  
      "OwnerId": "111122223333",  
      "Tags": []  
    }  
  ]  
}
```

詳細については、「Amazon VPC [Traffic Mirroring Guide](#)」の「トラフィックミラーターゲット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrafficMirrorTargets](#)」の「」を参照してください。

describe-transit-gateway-attachments

次のコード例は、describe-transit-gateway-attachments を使用する方法を示しています。

AWS CLI

Transit Gateway アタッチメントを表示するには

次のdescribe-transit-gateway-attachments例では、トランジットゲートウェイアタッチメントの詳細を表示します。

```
aws ec2 describe-transit-gateway-attachments
```

出力:

```
{
  "TransitGatewayAttachments": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
      "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
      "TransitGatewayOwnerId": "123456789012",
      "ResourceOwnerId": "123456789012",
      "ResourceType": "vpc",
      "ResourceId": "vpc-3EXAMPLE",
      "State": "available",
      "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "State": "associated"
      },
      "CreationTime": "2019-08-26T14:59:25.000Z",
      "Tags": [
        {
          "Key": "Name",
          "Value": "Example"
        }
      ]
    },
    {
      "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
      "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
      "TransitGatewayOwnerId": "123456789012",
      "ResourceOwnerId": "123456789012",
      "ResourceType": "vpc",
```



```
    "ResourceId": "vpc-0065acced4EXAMPLE",
    "State": "available",
    "Association": {
      "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
      "State": "associated"
    },
    "CreationTime": "2019-08-07T17:03:07.000Z",
    "Tags": []
  },
  {
    "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "TransitGatewayOwnerId": "123456789012",
    "ResourceOwnerId": "123456789012",
    "ResourceType": "direct-connect-gateway",
    "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "State": "available",
    "Association": {
      "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
      "State": "associated"
    },
    "CreationTime": "2019-08-14T20:27:44.000Z",
    "Tags": []
  },
  {
    "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "TransitGatewayOwnerId": "123456789012",
    "ResourceOwnerId": "123456789012",
    "ResourceType": "direct-connect-gateway",
    "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
    "State": "available",
    "Association": {
      "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
      "State": "associated"
    },
    "CreationTime": "2019-08-14T20:33:02.000Z",
    "Tags": []
  }
]
}
```

詳細については、[「トランジットゲートウェイガイド」](#)の「トランジットゲートウェイの操作」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTransitGatewayAttachments](#)」の「」を参照してください。

describe-transit-gateway-connect-peers

次の例は、describe-transit-gateway-connect-peers を使用する方法を説明しています。

AWS CLI

Transit Gateway Connect ピアを記述するには

次のdescribe-transit-gateway-connect-peers例では、指定された Connect ピアについて説明します。

```
aws ec2 describe-transit-gateway-connect-peers \  
--transit-gateway-connect-peer-ids tgw-connect-peer-0666adbac4EXAMPLE
```

出力:

```
{  
  "TransitGatewayConnectPeers": [  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-0f0927767cEXAMPLE",  
      "TransitGatewayConnectPeerId": "tgw-connect-peer-0666adbac4EXAMPLE",  
      "State": "available",  
      "CreationTime": "2021-10-13T03:35:17.000Z",  
      "ConnectPeerConfiguration": {  
        "TransitGatewayAddress": "10.0.0.234",  
        "PeerAddress": "172.31.1.11",  
        "InsideCidrBlocks": [  
          "169.254.6.0/29"  
        ],  
        "Protocol": "gre",  
        "BgpConfigurations": [  
          {  
            "TransitGatewayAsn": 64512,  
            "PeerAsn": 64512,  
            "TransitGatewayAddress": "169.254.6.2",  
            "PeerAddress": "169.254.6.1",
```

```

        "BgpStatus": "down"
      },
      {
        "TransitGatewayAsn": 64512,
        "PeerAsn": 64512,
        "TransitGatewayAddress": "169.254.6.3",
        "PeerAddress": "169.254.6.1",
        "BgpStatus": "down"
      }
    ]
  },
  "Tags": []
}
]
}

```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway Connect アタッチメント」と「Transit Gateway Connect ピア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTransitGatewayConnectPeers](#)」の「」を参照してください。

describe-transit-gateway-connects

次のコード例は、describe-transit-gateway-connects を使用する方法を示しています。

AWS CLI

Transit Gateway Connect アタッチメントを記述するには

次のdescribe-transit-gateway-connects例では、指定された Connect アタッチメントについて説明します。

```
aws ec2 describe-transit-gateway-connects \
  --transit-gateway-attachment-ids tgw-attach-037012e5dcEXAMPLE
```

出力:

```
{
  "TransitGatewayConnects": [
    {
```

```
    "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
    "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "State": "available",
    "CreationTime": "2021-03-09T19:59:17+00:00",
    "Options": {
      "Protocol": "gre"
    },
    "Tags": []
  }
]
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway Connect アタッチメント」と「Transit Gateway Connect ピア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTransitGatewayConnects](#)」の「」を参照してください。

describe-transit-gateway-multicast-domains

次の例は、describe-transit-gateway-multicast-domains を使用する方法を説明しています。

AWS CLI

Transit Gateway マルチキャストドメインを記述するには

次のdescribe-transit-gateway-multicast-domains例では、トランジットゲートウェイのすべてのマルチキャストドメインの詳細を表示します。

```
aws ec2 describe-transit-gateway-multicast-domains
```

出力:

```
{
  "TransitGatewayMulticastDomains": [
    {
      "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
      "TransitGatewayId": "tgw-0bf0bfffefaEXAMPLE",
```

```
    "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
east-1:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
    "OwnerId": "123456789012",
    "Options": {
      "Icmpv2Support": "disable",
      "StaticSourcesSupport": "enable",
      "AutoAcceptSharedAssociations": "disable"
    },
    "State": "available",
    "CreationTime": "2019-12-10T18:32:50+00:00",
    "Tags": [
      {
        "Key": "Name",
        "Value": "mc1"
      }
    ]
  }
]
```

詳細については、「Transit Gateways ガイド」の「[マルチキャストドメインの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTransitGatewayMulticastDomains](#)」の「」を参照してください。

describe-transit-gateway-peering-attachments

次のコード例は、describe-transit-gateway-peering-attachments を使用する方法を示しています。

AWS CLI

トランジットゲートウェイピアリングアタッチメントを記述するには

次のdescribe-transit-gateway-peering-attachments例では、トランジットゲートウェイピアリングアタッチメントの詳細を表示します。

```
aws ec2 describe-transit-gateway-peering-attachments
```

出力:

```
{
  "TransitGatewayPeeringAttachments": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbcccd",
      "RequesterTgwInfo": {
        "TransitGatewayId": "tgw-123abc05e04123abc",
        "OwnerId": "123456789012",
        "Region": "us-west-2"
      },
      "AcceptorTgwInfo": {
        "TransitGatewayId": "tgw-11223344aabbcc112",
        "OwnerId": "123456789012",
        "Region": "us-east-2"
      },
      "State": "pendingAcceptance",
      "CreationTime": "2019-12-09T11:38:05.000Z",
      "Tags": []
    }
  ]
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ピアリングアタッチメント」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTransitGatewayPeeringAttachments](#)」の「」を参照してください。

describe-transit-gateway-policy-tables

次のコード例は、describe-transit-gateway-policy-tables を使用する方法を示しています。

AWS CLI

トランジットゲートウェイポリシーテーブルを記述するには

次のdescribe-transit-gateway-policy-tables例では、指定された Transit Gateway ポリシーテーブルについて説明します。

```
aws ec2 describe-transit-gateway-policy-tables \
  --transit-gateway-policy-table-ids tgw-ptb-0a16f134b78668a81
```

出力:

```
{
  "TransitGatewayPolicyTables": [
    {
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",
      "State": "available",
      "CreationTime": "2023-11-28T16:36:43+00:00",
      "Tags": []
    }
  ]
}
```

詳細については、[「Transit Gateway ユーザーガイド」の「Transit Gateway ポリシーテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTransitGatewayPolicyTables](#)」の「」を参照してください。

describe-transit-gateway-route-tables

次の例は、describe-transit-gateway-route-tables を使用する方法を説明しています。

AWS CLI

トランジットゲートウェイルートテーブルを記述するには

次のdescribe-transit-gateway-route-tables例では、トランジットゲートウェイルートテーブルの詳細を表示します。

```
aws ec2 describe-transit-gateway-route-tables
```

出力:

```
{
  "TransitGatewayRouteTables": [
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0ca78a549EXAMPLE",
      "TransitGatewayId": "tgw-0bc994abffEXAMPLE",
      "State": "available",

```

```
    "DefaultAssociationRouteTable": true,
    "DefaultPropagationRouteTable": true,
    "CreationTime": "2018-11-28T14:24:49.000Z",
    "Tags": []
  },
  {
    "TransitGatewayRouteTableId": "tgw-rtb-0e8f48f148EXAMPLE",
    "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
    "State": "available",
    "DefaultAssociationRouteTable": true,
    "DefaultPropagationRouteTable": true,
    "CreationTime": "2018-11-28T14:24:00.000Z",
    "Tags": []
  }
]
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブルの表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTransitGatewayRouteTables](#)」の「」を参照してください。

describe-transit-gateway-vpc-attachments

次のコード例は、describe-transit-gateway-vpc-attachments を使用する方法を示しています。

AWS CLI

Transit Gateway アタッチメントを記述するには VPC

次のdescribe-transit-gateway-vpc-attachments例では、Transit Gateway VPCアタッチメントの詳細を表示します。

```
aws ec2 describe-transit-gateway-vpc-attachments
```

出力:

```
{
  "TransitGatewayVpcAttachments": [
```



```
{
  "TransitGatewayAttachmentId": "tgw-attach-0a08e88308EXAMPLE",
  "TransitGatewayId": "tgw-0043d72bb4EXAMPLE",
  "VpcId": "vpc-0f501f7ee8EXAMPLE",
  "VpcOwnerId": "111122223333",
  "State": "available",
  "SubnetIds": [
    "subnet-045d586432EXAMPLE",
    "subnet-0a0ad478a6EXAMPLE"
  ],
  "CreationTime": "2019-02-13T11:04:02.000Z",
  "Options": {
    "DnsSupport": "enable",
    "Ipv6Support": "disable"
  },
  "Tags": [
    {
      "Key": "Name",
      "Value": "attachment name"
    }
  ]
}
]
```

詳細については、「Transit Gateways ガイド」の [VPC「添付ファイルを表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTransitGatewayVpcAttachments](#)」の「」を参照してください。

describe-transit-gateways

次のコード例は、describe-transit-gateways を使用する方法を示しています。

AWS CLI

トランジットゲートウェイを記述するには

次のdescribe-transit-gateways例では、トランジットゲートウェイの詳細を取得します。

```
aws ec2 describe-transit-gateways
```

出力:

```
{
  "TransitGateways": [
    {
      "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
      "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0262a0e521EXAMPLE",
      "State": "available",
      "OwnerId": "111122223333",
      "Description": "MyTGW",
      "CreationTime": "2019-07-10T14:02:12.000Z",
      "Options": {
        "AmazonSideAsn": 64516,
        "AutoAcceptSharedAttachments": "enable",
        "DefaultRouteTableAssociation": "enable",
        "AssociationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
        "DefaultRouteTablePropagation": "enable",
        "PropagationDefaultRouteTableId": "tgw-rtb-018774adf3EXAMPLE",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
      },
      "Tags": []
    },
    {
      "TransitGatewayId": "tgw-0fb8421e2dEXAMPLE",
      "TransitGatewayArn": "arn:aws:ec2:us-east-2:111122223333:transit-
gateway/tgw-0fb8421e2da853bf3",
      "State": "available",
      "OwnerId": "111122223333",
      "CreationTime": "2019-03-15T22:57:33.000Z",
      "Options": {
        "AmazonSideAsn": 65412,
        "AutoAcceptSharedAttachments": "disable",
        "DefaultRouteTableAssociation": "enable",
        "AssociationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
        "DefaultRouteTablePropagation": "enable",
        "PropagationDefaultRouteTableId": "tgw-rtb-06a241a3d8EXAMPLE",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
      },
      "Tags": [
        {
          "Key": "Name",
```

```

    "Value": "TGW1"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTransitGateways](#)」の「」を参照してください。

describe-verified-access-endpoints

次のコード例は、describe-verified-access-endpoints を使用する方法を示しています。

AWS CLI

Verified Access エンドポイントを記述するには

次のdelete-verified-access-endpoints例では、指定された Verified Access エンドポイントについて説明します。

```
aws ec2 describe-verified-access-endpoints \
  --verified-access-endpoint-ids vae-066fac616d4d546f2
```

出力:

```
{
  "VerifiedAccessEndpoints": [
    {
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
      "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
      "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
      "ApplicationDomain": "example.com",
      "EndpointType": "network-interface",
      "AttachmentType": "vpc",
      "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
      "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
      "SecurityGroupIds": [

```

```
        "sg-004915970c4c8f13a"
    ],
    "NetworkInterfaceOptions": {
        "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
        "Protocol": "https",
        "Port": 443
    },
    "Status": {
        "Code": "active"
    },
    "Description": "",
    "CreationTime": "2023-08-25T20:54:43",
    "LastUpdatedTime": "2023-08-25T22:17:26",
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-va-endpoint"
        }
    ]
}
]
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access エンドポイント」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVerifiedAccessEndpoints](#)」の「」を参照してください。

describe-verified-access-groups

次の例は、describe-verified-access-groups を使用する方法を説明しています。

AWS CLI

Verified Access グループを記述するには

次のdescribe-verified-access-groups例では、指定された Verified Access グループについて説明します。

```
aws ec2 describe-verified-access-groups \
  --verified-access-group-ids vagr-0dbe967baf14b7235
```

出力:

```
{
  "VerifiedAccessGroups": [
    {
      "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
      "Description": "Testing Verified Access",
      "Owner": "123456789012",
      "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
      "CreationTime": "2023-08-25T19:55:19",
      "LastUpdatedTime": "2023-08-25T22:17:25",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-va-group"
        }
      ]
    }
  ]
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access グループ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVerifiedAccessGroups](#)」の「」を参照してください。

describe-verified-access-instance-logging-configurations

次の例は、describe-verified-access-instance-logging-configurations を使用する方法を説明しています。

AWS CLI

Verified Access インスタンスのログ記録設定を記述するには

次のdescribe-verified-access-instance-logging-configurations例では、指定された Verified Access インスタンスのログ記録設定について説明します。

```
aws ec2 describe-verified-access-instance-logging-configurations \
```

```
--verified-access-instance-ids vai-0ce000c0b7643abea
```

出力:

```
{
  "LoggingConfigurations": [
    {
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
      "AccessLogs": {
        "S3": {
          "Enabled": false
        },
        "CloudWatchLogs": {
          "Enabled": true,
          "DeliveryStatus": {
            "Code": "success"
          },
          "LogGroup": "my-log-group"
        },
        "KinesisDataFirehose": {
          "Enabled": false
        },
        "LogVersion": "ocsf-1.0.0-rc.2",
        "IncludeTrustContext": false
      }
    }
  ]
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access ログ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVerifiedAccessInstanceLoggingConfigurations](#)」の「」を参照してください。

describe-verified-access-instances

次のコード例は、describe-verified-access-instances を使用する方法を示しています。

AWS CLI

Verified Access インスタンスを記述するには

次のdescribe-verified-access-instances例では、指定された Verified Access インスタンスについて説明します。

```
aws ec2 describe-verified-access-instances \  
--verified-access-instance-ids vai-0ce000c0b7643abea
```

出力:

```
{  
  "VerifiedAccessInstances": [  
    {  
      "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
      "Description": "Testing Verified Access",  
      "VerifiedAccessTrustProviders": [  
        {  
          "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
          "TrustProviderType": "user",  
          "UserTrustProviderType": "iam-identity-center"  
        }  
      ],  
      "CreationTime": "2023-08-25T18:27:56",  
      "LastUpdatedTime": "2023-08-25T19:03:32",  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "my-ava-instance"  
        }  
      ]  
    }  
  ]  
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access インスタンス」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVerifiedAccessInstances](#)」の「」を参照してください。

describe-verified-access-trust-providers

次の例は、describe-verified-access-trust-providers を使用する方法を説明しています。

AWS CLI

Verified Access 信頼プロバイダーを記述するには

次のdescribe-verified-access-trust-providers例では、指定された Verified Access 信頼プロバイダーについて説明します。

```
aws ec2 describe-verified-access-trust-providers \  
  --verified-access-trust-provider-ids vatp-0bb32de759a3e19e7
```

出力:

```
{  
  "VerifiedAccessTrustProviders": [  
    {  
      "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
      "Description": "Testing Verified Access",  
      "TrustProviderType": "user",  
      "UserTrustProviderType": "iam-identity-center",  
      "PolicyReferenceName": "idc",  
      "CreationTime": "2023-08-25T19:00:38",  
      "LastUpdatedTime": "2023-08-25T19:03:32",  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "my-va-trust-provider"  
        }  
      ]  
    }  
  ]  
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access の信頼プロバイダー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVerifiedAccessTrustProviders](#)」の「」を参照してください。

describe-volume-attribute

次のコード例は、describe-volume-attribute を使用する方法を示しています。

AWS CLI

ボリューム属性を記述するには

このコマンド例では、ID を持つボリュームの `autoEnableIo` 属性について説明します `vol-049df61146c4d7901`。

コマンド:

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --  
attribute autoEnableIO
```

出力:

```
{  
  "AutoEnableIO": {  
    "Value": false  
  },  
  "VolumeId": "vol-049df61146c4d7901"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVolumeAttribute](#)」の「」を参照してください。

describe-volume-status

次の例は、`describe-volume-status` を使用する方法を説明しています。

AWS CLI

単一ボリュームのステータスを記述するには

このコマンド例では、ボリュームのステータスについて説明します `vol-1234567890abcdef0`。

コマンド:

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

出力:

```
{
  "VolumeStatuses": [
    {
      "VolumeStatus": {
        "Status": "ok",
        "Details": [
          {
            "Status": "passed",
            "Name": "io-enabled"
          },
          {
            "Status": "not-applicable",
            "Name": "io-performance"
          }
        ]
      },
      "AvailabilityZone": "us-east-1a",
      "VolumeId": "vol-1234567890abcdef0",
      "Actions": [],
      "Events": []
    }
  ]
}
```

障害のあるボリュームのステータスを記述するには

この例では、障害が発生したすべてのボリュームのステータスについて説明します。この例では、ボリュームに障害はありません。

コマンド:

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

出力:

```
{
  "VolumeStatuses": []
}
```

ステータスチェックに失敗したボリュームがある場合 (ステータスが損なわれている場合)、[「Amazon EC2ユーザーガイド」の「障害のあるボリュームの操作」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVolumeStatus](#)」の「」を参照してください。

describe-volumes-modifications

次の例は、describe-volumes-modifications を使用する方法を説明しています。

AWS CLI

ボリュームの変更ステータスを記述するには

次のdescribe-volumes-modifications例では、指定されたボリュームのボリューム変更ステータスについて説明します。

```
aws ec2 describe-volumes-modifications \  
  --volume-ids vol-1234567890abcdef0
```

出力:

```
{  
  "VolumeModification": {  
    "TargetSize": 150,  
    "TargetVolumeType": "io1",  
    "ModificationState": "optimizing",  
    "VolumeId": " vol-1234567890abcdef0",  
    "TargetIops": 100,  
    "StartTime": "2019-05-17T11:27:19.000Z",  
    "Progress": 70,  
    "OriginalVolumeType": "io1",  
    "OriginalIops": 100,  
    "OriginalSize": 100  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVolumesModifications](#)」の「」を参照してください。

describe-volumes

次の例は、describe-volumes を使用する方法を説明しています。

AWS CLI

例 1: ボリュームを記述するには

次のdescribe-volumes例では、現在のリージョンで指定されたボリュームについて説明します。

```
aws ec2 describe-volumes \  
  --volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

出力:

```
{  
  "Volumes": [  
    {  
      "AvailabilityZone": "us-east-1a",  
      "Attachments": [  
        {  
          "AttachTime": "2013-12-18T22:35:00.000Z",  
          "InstanceId": "i-1234567890abcdef0",  
          "VolumeId": "vol-049df61146c4d7901",  
          "State": "attached",  
          "DeleteOnTermination": true,  
          "Device": "/dev/sda1"  
        }  
      ],  
      "Encrypted": true,  
      "KmsKeyId": "arn:aws:kms:us-east-2a:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513eEXAMPLE",  
      "VolumeType": "gp2",  
      "VolumeId": "vol-049df61146c4d7901",  
      "State": "in-use",  
      "Iops": 100,  
      "SnapshotId": "snap-1234567890abcdef0",  
      "CreateTime": "2019-12-18T22:35:00.084Z",  
      "Size": 8  
    },  
    {  
      "AvailabilityZone": "us-east-1a",  
      "Attachments": [],  
      "Encrypted": false,  
      "VolumeType": "gp2",  
      "VolumeId": "vol-1234567890abcdef0",
```

```
        "State": "available",
        "Iops": 300,
        "SnapshotId": "",
        "CreateTime": "2020-02-27T00:02:41.791Z",
        "Size": 100
    }
]
}
```

例 2: 特定のインスタンスにアタッチされているボリュームを記述するには

次のdescribe-volumes例では、指定されたインスタンスにアタッチされ、インスタンスが終了したときに削除するように設定されたすべてのボリュームについて説明します。

```
aws ec2 describe-volumes \
  --region us-east-1 \
  --filters Name=attachment.instance-id,Values=i-1234567890abcdef0 Name=attachment.delete-on-termination,Values=true
```

describe-volumes の出力例については、例 1 を参照してください。

例 3: 特定のアベイラビリティゾーンで使用可能なボリュームを記述するには

次のdescribe-volumes例では、ステータスでavailable、指定されたアベイラビリティゾーンにあるすべてのボリュームについて説明します。

```
aws ec2 describe-volumes \
  --filters Name=status,Values=available Name=availability-zone,Values=us-east-1a
```

describe-volumes の出力例については、例 1 を参照してください。

例 4: タグに基づいてボリュームを記述するには

次のdescribe-volumes例では、タグキーNameと で始まる値を持つすべてのボリュームについて説明しますTest。その後、出力は、ボリュームIDsのタグと のみを表示するクエリでフィルタリングされます。

```
aws ec2 describe-volumes \
  --filters Name=tag:Name,Values=Test* \
  --query "Volumes[*].{ID:VolumeId,Tag:Tags}"
```

出力:

```
[
  {
    "Tag": [
      {
        "Value": "Test2",
        "Key": "Name"
      }
    ],
    "ID": "vol-1234567890abcdef0"
  },
  {
    "Tag": [
      {
        "Value": "Test1",
        "Key": "Name"
      }
    ],
    "ID": "vol-049df61146c4d7901"
  }
]
```

タグフィルターを使用するその他の例については、「Amazon EC2ユーザーガイド」の[「タグの操作」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVolumes](#)」の「」を参照してください。

describe-vpc-attribute

次のコード例は、describe-vpc-attribute を使用する方法を示しています。

AWS CLI

enableDnsSupport 属性を記述するには

この例では、enableDnsSupport 属性について説明します。この属性は、でDNS解決が有効になっているかどうかを示しますVPC。この属性が の場合true、Amazon DNSサーバーはインスタンスのDNSホスト名を対応する IP アドレスに解決します。それ以外の場合は解決しません。

コマンド:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

出力:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsSupport": {
    "Value": true
  }
}
```

enableDnsHostnames 属性を記述するには

この例では、enableDnsHostnames 属性について説明します。この属性は、インスタスが VPC get DNS hostnames で起動されたかどうかを示します。この属性が true の場合、インスタスは DNS ホスト名 VPC を取得します。それ以外の場合は取得しません。

コマンド:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

出力:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsHostnames": {
    "Value": true
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpcAttribute](#)」の「」を参照してください。

describe-vpc-classic-link-dns-support

次の例は、describe-vpc-classic-link-dns-support を使用する方法を説明しています。

AWS CLI

のサポートを記述 ClassicLink DNSするには VPCs

この例では、ClassicLink DNSすべての のサポートステータスについて説明しますVPCs。

コマンド:

```
aws ec2 describe-vpc-classic-link-dns-support
```

出力:

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-88888888",
      "ClassicLinkDnsSupported": true
    },
    {
      "VpcId": "vpc-1a2b3c4d",
      "ClassicLinkDnsSupported": false
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpcClassicLinkDnsSupport](#)」の「」を参照してください。

describe-vpc-classic-link

次の例は、describe-vpc-classic-link を使用する方法を説明しています。

AWS CLI

ClassicLink のステータスを記述するには VPCs

この例では、vpc-88888888 ClassicLink のステータスを一覧表示します。

コマンド:

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

出力:

```
{
  "Vpcs": [
```



```
{
  "ClassicLinkEnabled": true,
  "VpcId": "vpc-88888888",
  "Tags": [
    {
      "Value": "classiclinkvpc",
      "Key": "Name"
    }
  ]
}
```

この例では、Classiclink で有効になっている のみを一覧表示VPCsします (のフィルター値は `is-classic-link-enabled` 設定されています `true`)。

コマンド:

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-enabled,Values=true"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpcClassicLink](#)」の「」を参照してください。

describe-vpc-endpoint-connection-notifications

次の例は、`describe-vpc-endpoint-connection-notifications` を使用する方法を説明しています。

AWS CLI

エンドポイント接続通知を記述するには

次の `describe-vpc-endpoint-connection-notifications` 例では、エンドポイント接続通知をすべて説明しています。

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

出力:

```
{
  "ConnectionNotificationSet": [
```

```
{
  "ConnectionNotificationState": "Enabled",
  "ConnectionNotificationType": "Topic",
  "ConnectionEvents": [
    "Accept",
    "Reject",
    "Delete",
    "Connect"
  ],
  "ConnectionNotificationId": "vpce-nfn-04bcb952bc8af7abc",
  "ConnectionNotificationArn": "arn:aws:sns:us-
east-1:123456789012:VpceNotification",
  "VpcEndpointId": "vpce-0324151a02f327123"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpcEndpointConnections](#)」の「」を参照してください。

describe-vpc-endpoint-connections

次の例は、describe-vpc-endpoint-connections を使用する方法を説明しています。

AWS CLI

VPCエンドポイント接続を記述するには

この例では、エンドポイントサービスへのインターフェイスエンドポイント接続について説明し、結果をフィルタリングして `PendingAcceptance` の状態にあるエンドポイントを表示します。

コマンド:

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

出力:

```
{
  "VpcEndpointConnections": [
    {
      "VpcEndpointId": "vpce-0abed31004e618123",
```

```
    "ServiceId": "vpce-svc-0abced088d20def56",
    "CreationTimestamp": "2017-11-30T10:00:24.350Z",
    "VpcEndpointState": "pendingAcceptance",
    "VpcEndpointOwner": "123456789012"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVpcEndpointConnections](#)」の「」を参照してください。

describe-vpc-endpoint-service-configurations

次のコード例は、describe-vpc-endpoint-service-configurations を使用する方法を示しています。

AWS CLI

エンドポイントサービス設定を記述するには

次のdescribe-vpc-endpoint-service-configurations例では、エンドポイントサービスの設定について説明します。

```
aws ec2 describe-vpc-endpoint-service-configurations
```

出力:

```
{
  "ServiceConfigurations": [
    {
      "ServiceType": [
        {
          "ServiceType": "GatewayLoadBalancer"
        }
      ],
      "ServiceId": "vpce-svc-012d33a1c4321cab",
      "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-012d33a1c4321cab",
      "ServiceState": "Available",
      "AvailabilityZones": [
        "us-east-1d"
      ],
    }
  ]
}
```

```
    "AcceptanceRequired": false,
    "ManagesVpcEndpoints": false,
    "GatewayLoadBalancerArns": [
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
gwy/GWLBService/123210844e429123"
    ],
    "Tags": []
  },
  {
    "ServiceType": [
      {
        "ServiceType": "Interface"
      }
    ],
    "ServiceId": "vpce-svc-123cab125efa123",
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123cab125efa123",
    "ServiceState": "Available",
    "AvailabilityZones": [
      "us-east-1a"
    ],
    "AcceptanceRequired": true,
    "ManagesVpcEndpoints": false,
    "NetworkLoadBalancerArns": [
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
net/NLBforService/1238753950b25123"
    ],
    "BaseEndpointDnsNames": [
      "vpce-svc-123cab125efa123.us-east-1.vpce.amazonaws.com"
    ],
    "PrivateDnsName": "example.com",
    "PrivateDnsNameConfiguration": {
      "State": "failed",
      "Type": "TXT",
      "Value": "vpce:qUAth3FdeABCaPuiXabc",
      "Name": "_1d367jvbg34znqvyefrj"
    },
    "Tags": []
  }
]
}
```

詳細については、「Amazon ユーザーガイド」の[VPC「エンドポイントサービス」](#)を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpcEndpointServiceConfigurations](#)」の「」を参照してください。

describe-vpc-endpoint-service-permissions

次の例は、describe-vpc-endpoint-service-permissions を使用する方法を説明しています。

AWS CLI

エンドポイントサービスのアクセス許可を記述するには

この例では、指定されたエンドポイントサービスのアクセス許可について説明します。

コマンド:

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
```

出力:

```
{
  "AllowedPrincipals": [
    {
      "PrincipalType": "Account",
      "Principal": "arn:aws:iam::123456789012:root"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpcEndpointServicePermissions](#)」の「」を参照してください。

describe-vpc-endpoint-services

次の例は、describe-vpc-endpoint-services を使用する方法を説明しています。

AWS CLI

例 1: すべてのVPCエンドポイントサービスを記述するには

次のdescribe-vpc-endpoint-services 「」例は、AWS リージョンのすべてのVPCエンドポイントサービスを一覧表示します。

```
aws ec2 describe-vpc-endpoint-services
```

出力:

```
{
  "ServiceDetails": [
    {
      "ServiceType": [
        {
          "ServiceType": "Gateway"
        }
      ],
      "AcceptanceRequired": false,
      "ServiceName": "com.amazonaws.us-east-1.dynamodb",
      "VpcEndpointPolicySupported": true,
      "Owner": "amazon",
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ],
      "BaseEndpointDnsNames": [
        "dynamodb.us-east-1.amazonaws.com"
      ]
    },
    {
      "ServiceType": [
        {
          "ServiceType": "Interface"
        }
      ],
      "PrivateDnsName": "ec2.us-east-1.amazonaws.com",
      "ServiceName": "com.amazonaws.us-east-1.ec2",
      "VpcEndpointPolicySupported": false,
      "Owner": "amazon",
      "AvailabilityZones": [
        "us-east-1a",
```

```
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
    ],
    "AcceptanceRequired": false,
    "BaseEndpointDnsNames": [
        "ec2.us-east-1.vpce.amazonaws.com"
    ]
},
{
    "ServiceType": [
        {
            "ServiceType": "Interface"
        }
    ],
    "PrivateDnsName": "ssm.us-east-1.amazonaws.com",
    "ServiceName": "com.amazonaws.us-east-1.ssm",
    "VpcEndpointPolicySupported": true,
    "Owner": "amazon",
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e"
    ],
    "AcceptanceRequired": false,
    "BaseEndpointDnsNames": [
        "ssm.us-east-1.vpce.amazonaws.com"
    ]
}
],
"ServiceNames": [
    "com.amazonaws.us-east-1.dynamodb",
    "com.amazonaws.us-east-1.ec2",
    "com.amazonaws.us-east-1.ec2messages",
    "com.amazonaws.us-east-1.elasticloadbalancing",
    "com.amazonaws.us-east-1.kinesis-streams",
    "com.amazonaws.us-east-1.s3",
    "com.amazonaws.us-east-1.ssm"
]
```

```
}
```

詳細については、「ユーザーガイド」の[「使用可能な AWS サービス名を表示する」](#)を参照してください。 AWS PrivateLink

例 2: エンドポイントサービスの詳細を記述するには

次のdescribe-vpc-endpoint-services「」の例は、Amazon S3 インターフェイスエンドポイントサービスの詳細を一覧表示します。

```
aws ec2 describe-vpc-endpoint-services \
  --filter "Name=service-type,Values=Interface" Name=service-
  name,Values=com.amazonaws.us-east-1.s3
```

出力:

```
{
  "ServiceDetails": [
    {
      "ServiceName": "com.amazonaws.us-east-1.s3",
      "ServiceId": "vpce-svc-081d84efcdEXAMPLE",
      "ServiceType": [
        {
          "ServiceType": "Interface"
        }
      ],
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ],
      "Owner": "amazon",
      "BaseEndpointDnsNames": [
        "s3.us-east-1.vpce.amazonaws.com"
      ],
      "VpcEndpointPolicySupported": true,
      "AcceptanceRequired": false,
      "ManagesVpcEndpoints": false,
      "Tags": []
    }
  ]
}
```



```
    ],  
    "ServiceNames": [  
        "com.amazonaws.us-east-1.s3"  
    ]  
}
```

詳細については、「ユーザーガイド」の「[利用可能な AWS サービス名を表示する](#)」を参照してください。 AWS PrivateLink

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVpcEndpointServices](#)」の「」を参照してください。

describe-vpc-endpoints

次のコード例は、describe-vpc-endpoints を使用する方法を示しています。

AWS CLI

VPCエンドポイントを記述するには

次のdescribe-vpc-endpoints例では、すべてのVPCエンドポイントの詳細を表示します。

```
aws ec2 describe-vpc-endpoints
```

出力:

```
{  
  "VpcEndpoints": [  
    {  
      "PolicyDocument": "{\n\"Version\": \"2008-10-17\",\n\"Statement\": [\n{\n\"Effect\": \"Allow\",\n\"Principal\": \"*\",\n\"Action\": \"*\",\n\"Resource\": \"*\"}\n]}\n",  
      "VpcId": "vpc-aabb1122",  
      "NetworkInterfaceIds": [],  
      "SubnetIds": [],  
      "PrivateDnsEnabled": true,  
      "State": "available",  
      "ServiceName": "com.amazonaws.us-east-1.dynamodb",  
      "RouteTableIds": [  
        "rtb-3d560345"  
      ],  
      "Groups": [],  
      "VpcEndpointId": "vpce-032a826a",  
      "VpcEndpointType": "Gateway",  
    }  
  ]  
}
```

```

    "CreationTimestamp": "2017-09-05T20:41:28Z",
    "DnsEntries": [],
    "OwnerId": "123456789012"
  },
  {
    "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*\n\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \"*\", \n      \"Resource\n\": \"*\">\n    ]\n  ]\n}",
    "VpcId": "vpc-1a2b3c4d",
    "NetworkInterfaceIds": [
      "eni-2ec2b084",
      "eni-1b4a65cf"
    ],
    "SubnetIds": [
      "subnet-d6fcaa8d",
      "subnet-7b16de0c"
    ],
    "PrivateDnsEnabled": false,
    "State": "available",
    "ServiceName": "com.amazonaws.us-east-1.elasticloadbalancing",
    "RouteTableIds": [],
    "Groups": [
      {
        "GroupName": "default",
        "GroupId": "sg-54e8bf31"
      }
    ],
    "VpcEndpointId": "vpce-0f89a33420c1931d7",
    "VpcEndpointType": "Interface",
    "CreationTimestamp": "2017-09-05T17:55:27.583Z",
    "DnsEntries": [
      {
        "HostedZoneId": "Z7HUB22UULQXV",
        "DnsName": "vpce-0f89a33420c1931d7-
bluzidnv.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
      },
      {
        "HostedZoneId": "Z7HUB22UULQXV",
        "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1b.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
      },
      {
        "HostedZoneId": "Z7HUB22UULQXV",

```

```
        "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1a.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
    },
    "OwnerId": "123456789012"
},
{
    "VpcEndpointId": "vpce-aabbaabbaabbaabba",
    "VpcEndpointType": "GatewayLoadBalancer",
    "VpcId": "vpc-111122223333aabbc",
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-123123a1c43abc123",
    "State": "available",
    "SubnetIds": [
        "subnet-0011aabbcc2233445"
    ],
    "RequesterManaged": false,
    "NetworkInterfaceIds": [
        "eni-01010120203030405"
    ],
    "CreationTimestamp": "2020-11-11T08:06:03.522Z",
    "Tags": [],
    "OwnerId": "123456789012"
}
]
}
```

詳細については、「Amazon ユーザーガイド」の[VPC「エンドポイント」](#)を参照してください。
VPC

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeVpcEndpoints](#)」の「」を参照してください。

describe-vpc-peering-connections

次のコード例は、describe-vpc-peering-connections を使用する方法を示しています。

AWS CLI

VPCピアリング接続を記述するには

この例では、すべてのVPCピアリング接続について説明します。

コマンド:

```
aws ec2 describe-vpc-peering-connections
```

出力:

```
{
  "VpcPeeringConnections": [
    {
      "Status": {
        "Message": "Active",
        "Code": "active"
      },
      "Tags": [
        {
          "Value": "Peering-1",
          "Key": "Name"
        }
      ],
      "AcceptorVpcInfo": {
        "OwnerId": "111122223333",
        "VpcId": "vpc-1a2b3c4d",
        "CidrBlock": "10.0.1.0/28"
      },
      "VpcPeeringConnectionId": "pcx-11122233",
      "RequesterVpcInfo": {
        "PeeringOptions": {
          "AllowEgressFromLocalVpcToRemoteClassicLink": false,
          "AllowEgressFromLocalClassicLinkToRemoteVpc": false
        },
        "OwnerId": "444455556666",
        "VpcId": "vpc-123abc45",
        "CidrBlock": "192.168.0.0/16"
      }
    },
    {
      "Status": {
        "Message": "Pending Acceptance by 444455556666",
        "Code": "pending-acceptance"
      },
      "Tags": [],
      "RequesterVpcInfo": {
        "PeeringOptions": {
```

```
        "AllowEgressFromLocalVpcToRemoteClassicLink": false,
        "AllowEgressFromLocalClassicLinkToRemoteVpc": false
    },
    "OwnerId": "444455556666",
    "VpcId": "vpc-11aa22bb",
    "CidrBlock": "10.0.0.0/28"
},
"VpcPeeringConnectionId": "pcx-abababab",
"ExpirationTime": "2014-04-03T09:12:43.000Z",
"AccepterVpcInfo": {
    "OwnerId": "444455556666",
    "VpcId": "vpc-33cc44dd"
}
}
]
}
```

特定のVPCピアリング接続を記述するには

この例では、保留中の受け入れ状態にあるすべてのVPCピアリング接続について説明します。

コマンド:

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

この例では、「Owner=Finance」というタグを持つすべてのVPCピアリング接続について説明します。

コマンド:

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

この例では、指定された vpc-1a2b3c4d VPCに対してリクエストしたすべてのVPCピアリング接続について説明します。

コマンド:

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpcPeeringConnections](#)」の「」を参照してください。

describe-vpcs

次の例は、describe-vpcs を使用する方法を説明しています。

AWS CLI

例 1: すべての を記述するには VPCs

次のdescribe-vpcs例では、 の詳細を取得しますVPCs。

```
aws ec2 describe-vpcs
```

出力:

```
{
  "Vpcs": [
    {
      "CidrBlock": "30.1.0.0/16",
      "DhcpOptionsId": "dopt-19edf471",
      "State": "available",
      "VpcId": "vpc-0e9801d129EXAMPLE",
      "OwnerId": "111122223333",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-062c64cfafEXAMPLE",
          "CidrBlock": "30.1.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": false,
      "Tags": [
        {
          "Key": "Name",
          "Value": "Not Shared"
        }
      ]
    }
  ],
}
```

```

    {
      "CidrBlock": "10.0.0.0/16",
      "DhcpOptionsId": "dopt-19edf471",
      "State": "available",
      "VpcId": "vpc-06e4ab6c6cEXAMPLE",
      "OwnerId": "222222222222",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
          "CidrBlock": "10.0.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": false,
      "Tags": [
        {
          "Key": "Name",
          "Value": "Shared VPC"
        }
      ]
    }
  ]
}

```

例 2: 指定された を記述するには VPC

次のdescribe-vpcs例では、指定された の詳細を取得しますVPC。

```

aws ec2 describe-vpcs \
  --vpc-ids vpc-06e4ab6c6cEXAMPLE

```

出力:

```

{
  "Vpcs": [
    {
      "CidrBlock": "10.0.0.0/16",
      "DhcpOptionsId": "dopt-19edf471",
      "State": "available",
      "VpcId": "vpc-06e4ab6c6cEXAMPLE",

```

```
    "OwnerId": "111122223333",
    "InstanceTenancy": "default",
    "CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
        "CidrBlock": "10.0.0.0/16",
        "CidrBlockState": {
          "State": "associated"
        }
      }
    ],
    "IsDefault": false,
    "Tags": [
      {
        "Key": "Name",
        "Value": "Shared VPC"
      }
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVpcs](#)」の「」を参照してください。

describe-vpn-connections

次の例は、describe-vpn-connections を使用する方法を説明しています。

AWS CLI

例 1: VPN接続を記述するには

次のdescribe-vpn-connections例では、Site-to-SiteすべてのVPN接続について説明します。

```
aws ec2 describe-vpn-connections
```

出力:

```
{
  "VpnConnections": [
```



```
{
  "CustomerGatewayConfiguration": "...configuration information...",
  "CustomerGatewayId": "cgw-01234567abcde1234",
  "Category": "VPN",
  "State": "available",
  "Type": "ipsec.1",
  "VpnConnectionId": "vpn-1122334455aabbccd",
  "TransitGatewayId": "tgw-00112233445566aab",
  "Options": {
    "EnableAcceleration": false,
    "StaticRoutesOnly": true,
    "LocalIpv4NetworkCidr": "0.0.0.0/0",
    "RemoteIpv4NetworkCidr": "0.0.0.0/0",
    "TunnelInsideIpVersion": "ipv4"
  },
  "Routes": [],
  "Tags": [
    {
      "Key": "Name",
      "Value": "CanadaVPN"
    }
  ],
  "VgwTelemetry": [
    {
      "AcceptedRouteCount": 0,
      "LastStatusChange": "2020-07-29T10:35:11.000Z",
      "OutsideIpAddress": "203.0.113.3",
      "Status": "DOWN",
      "StatusMessage": ""
    },
    {
      "AcceptedRouteCount": 0,
      "LastStatusChange": "2020-09-02T09:09:33.000Z",
      "OutsideIpAddress": "203.0.113.5",
      "Status": "UP",
      "StatusMessage": ""
    }
  ]
}
```

詳細については、「ユーザーガイド」の「[AWS Site-to-Site VPNの仕組み](#)」を参照してください。AWS Site-to-Site VPN

例 2: 使用可能なVPN接続を記述するには

次のdescribe-vpn-connections例では、の状態VPNの接続について説明します Site-to-Siteavailable。

```
aws ec2 describe-vpn-connections \  
  --filters "Name=state,Values=available"
```

詳細については、「ユーザーガイド」の「[AWS Site-to-Site VPNの仕組み](#)」を参照してください。AWS Site-to-Site VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVpnConnections](#)」の「」を参照してください。

describe-vpn-gateways

次の例は、describe-vpn-gateways を使用する方法を説明しています。

AWS CLI

仮想プライベートゲートウェイを記述するには

この例では、仮想プライベートゲートウェイについて説明します。

コマンド:

```
aws ec2 describe-vpn-gateways
```

出力:

```
{  
  "VpnGateways": [  
    {  
      "State": "available",  
      "Type": "ipsec.1",  
      "VpnGatewayId": "vgw-f211f09b",  
      "VpcAttachments": [  
        {
```

```
        "State": "attached",
        "VpcId": "vpc-98eb5ef5"
      }
    ]
  },
  {
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": [
      {
        "State": "attaching",
        "VpcId": "vpc-a01106c2"
      }
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeVpnGateways](#)」の「」を参照してください。

detach-classic-link-vpc

次のコード例は、`detach-classic-link-vpc` を使用する方法を示しています。

AWS CLI

から EC2-Classical インスタンスのリンクを解除 (デタッチ) するには VPC

この例では、インスタンス `i-0598c7d356eba48d7` を `vpc-88888888` VPC からリンク解除します。

コマンド:

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-id vpc-88888888
```

出力:

```
{
  "Return": true
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachClassicLinkVpc](#)」の「」を参照してください。

detach-internet-gateway

次の例は、detach-internet-gateway を使用する方法を説明しています。

AWS CLI

からインターネットゲートウェイをデタッチするには VPC

次のdetach-internet-gateway例では、指定されたインターネットゲートウェイを特定の からデタッチしますVPC。

```
aws ec2 detach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「Amazon VPCユーザーガイド」の「[インターネットゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachInternetGateway](#)」の「」を参照してください。

detach-network-interface

次のコード例は、detach-network-interface を使用する方法を示しています。

AWS CLI

インスタンスからネットワークインターフェイスをデタッチするには

この例では、指定されたインスタンスから指定されたネットワークインターフェイスをデタッチします。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachNetworkInterface](#)」の「」を参照してください。

detach-verified-access-trust-provider

次のコード例は、`detach-verified-access-trust-provider` を使用する方法を示しています。

AWS CLI

インスタンスから信頼プロバイダーをデタッチするには

次の`detach-verified-access-trust-provider`例では、指定された Verified Access 信頼プロバイダーを指定された Verified Access インスタンスからデタッチします。

```
aws ec2 detach-verified-access-trust-provider \  
  --verified-access-instance-id vai-0ce000c0b7643abea \  
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

出力:

```
{  
  "VerifiedAccessTrustProvider": {  
    "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",  
    "Description": "Testing Verified Access",  
    "TrustProviderType": "user",  
    "UserTrustProviderType": "iam-identity-center",  
    "PolicyReferenceName": "idc",  
    "CreationTime": "2023-08-25T19:00:38",  
    "LastUpdatedTime": "2023-08-25T19:00:38"  
  },  
  "VerifiedAccessInstance": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "Testing Verified Access",  
    "VerifiedAccessTrustProviders": [],  
    "CreationTime": "2023-08-25T18:27:56",  
    "LastUpdatedTime": "2023-08-25T18:27:56"  
  }  
}
```

```
}
```

詳細については、[「Verified Access ユーザーガイド」](#)の[「Verified Access インスタンス」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachVerifiedAccessTrustProvider](#)」の「」を参照してください。

detach-volume

次の例は、detach-volume を使用する方法を説明しています。

AWS CLI

インスタンスからボリュームをデタッチするには

この例では、ボリューム (vol-049df61146c4d7901) をアタッチされているインスタンスからデタッチします。

コマンド:

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

出力:

```
{
  "AttachTime": "2014-02-27T19:23:06.000Z",
  "InstanceId": "i-1234567890abcdef0",
  "VolumeId": "vol-049df61146c4d7901",
  "State": "detaching",
  "Device": "/dev/sdb"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachVolume](#)」の「」を参照してください。

detach-vpn-gateway

次の例は、detach-vpn-gateway を使用する方法を説明しています。

AWS CLI

仮想プライベートゲートウェイを からデタッチするには VPC

この例では、指定された仮想プライベートゲートウェイを指定された からデタッチしますVPC。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachVpnGateway](#)」の「」を参照してください。

disable-address-transfer

次のコード例は、`disable-address-transfer` を使用する方法を示しています。

AWS CLI

Elastic IP アドレス転送を無効にするには

次の`disable-address-transfer`例では、指定された Elastic IP アドレスの Elastic IP アドレス転送を無効にします。

```
aws ec2 disable-address-transfer \  
  --allocation-id eipalloc-09ad461b0d03f6aaf
```

出力:

```
{  
  "AddressTransfer": {  
    "PublicIp": "100.21.184.216",  
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",  
    "AddressTransferStatus": "disabled"  
  }  
}
```

詳細については、「Amazon VPCユーザーガイド」の「[Elastic IP アドレスを転送する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableAddressTransfer](#)」の「」を参照してください。

disable-aws-network-performance-metric-subscription

次の例は、`disable-aws-network-performance-metric-subscription` を使用する方法を説明しています。

AWS CLI

メトリクスサブスクリプションを無効にするには

次の`disable-aws-network-performance-metric-subscription`例では、指定された送信元リージョンと送信先リージョン間の集約ネットワークレイテンシーのモニタリングを無効にします。

```
aws ec2 disable-aws-network-performance-metric-subscription \  
  --source us-east-1 \  
  --destination eu-west-1 \  
  --metric aggregate-latency \  
  --statistic p50
```

出力:

```
{  
  "Output": true  
}
```

詳細については、「Infrastructure Performance ユーザーガイド」の[「サブスクリプションの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableAwsNetworkPerformanceMetricSubscription](#)」の「」を参照してください。

disable-ebs-encryption-by-default

次の例は、`disable-ebs-encryption-by-default` を使用する方法を説明しています。

AWS CLI

デフォルトではEBS暗号化を無効にするには

次のdisable-ebs-encryption-by-default例では、現在のリージョンの AWS アカウントのEBS暗号化をデフォルトで無効にします。

```
aws ec2 disable-ebs-encryption-by-default
```

出力:

```
{
  "EbsEncryptionByDefault": false
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableEbsEncryptionByDefault](#)」の「」を参照してください。

disable-fast-launch

次の例は、disable-fast-launch を使用する方法を説明しています。

AWS CLI

イメージの高速起動を中止するには

次のdisable-fast-launch例では、指定された の高速起動を中止しAMI、事前プロビジョニングされた既存のスナップショットをクリーンアップします。

```
aws ec2 disable-fast-launch \
  --image-id ami-01234567890abcdef
```

出力:

```
{
  "ImageId": "ami-01234567890abcdef",
  "ResourceType": "snapshot",
  "SnapshotConfiguration": {},
  "LaunchTemplate": {
    "LaunchTemplateId": "lt-01234567890abcdef",
    "LaunchTemplateName": "EC2FastLaunchDefaultResourceCreation-a8c6215d-94e6-441b-9272-dbd1f87b07e2",
    "Version": "1"
  },
}
```

```
"MaxParallelLaunches": 6,  
"OwnerId": "0123456789123",  
"State": "disabling",  
"StateTransitionReason": "Client.UserInitiated",  
"StateTransitionTime": "2022-01-27T22:47:29.265000+00:00"  
}
```

起動を高速化AMIするための Windows の設定の詳細については、「Amazon EC2ユーザーガイド」の「[起動を高速化AMIするための の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableFastLaunch](#)」の「」を参照してください。

disable-fast-snapshot-restores

次のコード例は、`disable-fast-snapshot-restores` を使用する方法を示しています。

AWS CLI

高速スナップショット復元を無効にするには

次の`disable-fast-snapshot-restores`例では、指定されたアベイラビリティーゾーン内の指定されたスナップショットの高速スナップショット復元を無効にします。

```
aws ec2 disable-fast-snapshot-restores \  
  --availability-zones us-east-2a \  
  --source-snapshot-ids snap-1234567890abcdef0
```

出力:

```
{  
  "Successful": [  
    {  
      "SnapshotId": "snap-1234567890abcdef0"  
      "AvailabilityZone": "us-east-2a",  
      "State": "disabling",  
      "StateTransitionReason": "Client.UserInitiated",  
      "OwnerId": "123456789012",  
      "EnablingTime": "2020-01-25T23:57:49.602Z"  
    }  
  ],  
}
```

```
"Unsuccessful": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableFastSnapshotRestores](#)」の「」を参照してください。

disable-image-block-public-access

次の例は、`disable-image-block-public-access` を使用する方法を説明しています。

AWS CLI

指定されたリージョンAMIsでのパブリックアクセスのブロックを無効にするには

次の`disable-image-block-public-access`例では、指定されたリージョンのアカウントレベルでAMIsのパブリックアクセスのブロックを無効にします。

```
aws ec2 disable-image-block-public-access \
  --region us-east-1
```

出力:

```
{
  "ImageBlockPublicAccessState": "unblocked"
}
```

詳細については、「[Amazon ユーザーガイド](#)」の「[へのパブリックアクセスのブロックAMIs](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableImageBlockPublicAccess](#)」の「」を参照してください。

disable-image-deprecation

次のコード例は、`disable-image-deprecation` を使用する方法を示しています。

AWS CLI

の非推奨をキャンセルするには AMI

次のdisable-image-deprecation例では、の非推奨をキャンセルしAMI、describe-images出力から DeprecationTimeフィールドを削除します。この手順を実行するには、AMI所有者である必要があります。

```
aws ec2 disable-image-deprecation \  
  --image-id ami-1234567890abcdef0
```

出力:

```
{  
  "RequestID": "11aabb229-4eac-35bd-99ed-be587EXAMPLE",  
  "Return": "true"  
}
```

詳細については、「Amazon EC2ユーザーガイド」のAMI「<<https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami>>を廃止する」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DisableImageDeprecation](#)」の「」を参照してください。

disable-image

次の例は、disable-image を使用する方法を説明しています。

AWS CLI

を無効にするには AMI

次のdisable-image例では、指定された を無効にしますAMI。

```
aws ec2 disable-image \  
  --image-id ami-1234567890abcdef0
```

出力:

```
{  
  "Return": "true"  
}
```

詳細については、「Amazon EC2ユーザーガイド」の「[を無効にするAMI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableImage](#)」の「」を参照してください。

disable-ipam-organization-admin-account

次の例は、`disable-ipam-organization-admin-account` を使用する方法を説明しています。

AWS CLI

委任されたIPAM管理者を無効にするには

特定のシナリオでは、AWS Organizations IPAMと統合します。これを行うと、AWS Organizations 管理アカウントはAWS Organizations メンバーアカウントをIPAM管理者として委任します。

この例では、管理アカウントを委任したAWS Organizations IPAM 管理アカウントであり、そのアカウントをIPAM管理者として無効にします。

このリクエストを行う`--region`ときは、に任意のAWSリージョンを使用できます。最初に管理者を委任したリージョン、IPAMが作成されたリージョン、またはIPAM運用リージョンを使用する必要はありません。委任された管理者アカウントを無効にすると、いつでも再有効化することも、新しいアカウントをIPAM管理者として委任することもできます。

次の`disable-ipam-organization-admin-account`例では、AWSアカウントの委任されたIPAM管理者を無効にします。

```
aws ec2 disable-ipam-organization-admin-account \  
  --delegated-admin-account-id 320805250157 \  
  --region ap-south-1
```

出力:

```
{  
  "Success": true  
}
```

詳細については、「[Amazon VPCIPAMユーザーガイド](#)」のAWS「[組織内のアカウントIPAMとの統合](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableIpamOrganizationAdminAccount](#)」の「」を参照してください。

disable-serial-console-access

次のコード例は、`disable-serial-console-access` を使用する方法を示しています。

AWS CLI

アカウントのEC2シリアルコンソールへのアクセスを無効にするには

次の`disable-serial-console-access`例では、シリアルコンソールへのアカウントアクセスを無効にします。

```
aws ec2 disable-serial-console-access
```

出力:

```
{
  "SerialConsoleAccessEnabled": false
}
```

詳細については、「Amazon ユーザーガイド」の[EC2「シリアルコンソール」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableSerialConsoleAccess](#)」の「」を参照してください。

disable-snapshot-block-public-access

次の例は、`disable-snapshot-block-public-access` を使用する方法を説明しています。

AWS CLI

スナップショットのパブリックアクセスのブロックを無効にするには

次の`disable-snapshot-block-public-access`例では、スナップショットのパブリックアクセスのブロックを無効にして、スナップショットのパブリック共有を許可します。

```
aws ec2 disable-snapshot-block-public-access
```

出力:

```
{
```

```
"State": "unblocked"
}
```

詳細については、「Amazon EBSユーザーガイド」の[「スナップショットのパブリックアクセスをブロックする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableSnapshotBlockPublicAccess](#)」の「」を参照してください。

disable-transit-gateway-route-table-propagation

次の例は、`disable-transit-gateway-route-table-propagation` を使用する方法を説明しています。

AWS CLI

トランジットゲートウェイアタッチメントを無効にして、指定された伝播ルートテーブルにルートを伝播するには

次の`disable-transit-gateway-route-table-propagation`例では、指定されたアタッチメントを無効にして、指定された伝播ルートテーブルにルートを伝播します。

```
aws ec2 disable-transit-gateway-route-table-propagation \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

出力:

```
{
  "Propagation": {
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
    "ResourceId": "vpc-4d7de228",
    "ResourceType": "vpc",
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
    "State": "disabled"
  }
}
```

詳細については、[「Transit Gateways ガイド」](#)の「[Transit Gateway ルートテーブル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisableTransitGatewayRouteTablePropagation](#)」の「」を参照してください。

disable-vgw-route-propagation

次のコード例は、`disable-vgw-route-propagation` を使用する方法を示しています。

AWS CLI

ルート伝播を無効にするには

この例では、指定された仮想プライベートゲートウェイが静的ルートを指定されたルートテーブルに伝播することを無効にします。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DisableVgwRoutePropagation](#)」の「」を参照してください。

disable-vpc-classic-link-dns-support

次のコード例は、`disable-vpc-classic-link-dns-support` を使用する方法を示しています。

AWS CLI

のサポートを無効にする ClassicLink DNSには VPC

この例では、ClassicLink DNS のサポートを無効にします `vpc-88888888`。

コマンド:

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

出力:

```
{  
  "Return": true
```



```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableVpcClassicLinkDnsSupport](#)」の「」を参照してください。

disable-vpc-classic-link

次のコード例は、`disable-vpc-classic-link` を使用する方法を示しています。

AWS CLI

ClassicLink で を無効にするには VPC

この例では、`vpc-8888888` ClassicLink に対して を無効にします。

コマンド:

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

出力:

```
{  
  "Return": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableVpcClassicLink](#)」の「」を参照してください。

disassociate-address

次の例は、`disassociate-address` を使用する方法を説明しています。

AWS CLI

EC2-Classic で Elastic IP アドレスの関連付けを解除するには

この例では、EC2-Classic のインスタンスから Elastic IP アドレスの関連付けを解除します。コマンドが成功した場合、出力は返りません。

コマンド:

詳細については、クライアント管理者ガイドの「[ターゲットネットワーク](#)」を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateClientVpnTargetNetwork](#)」の「」を参照してください。

disassociate-iam-instance-profile

次の例は、disassociate-iam-instance-profile を使用する方法を説明しています。

AWS CLI

IAMインスタンスプロファイルの関連付けを解除するには

この例では、IAMインスタンスプロファイルと関連付け ID の関連付けを解除します `iip-
assoc-05020b59952902f5f`。

コマンド:

```
aws ec2 disassociate-iam-instance-profile --association-id iip-  
assoc-05020b59952902f5f
```

出力:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-123456789abcde123",  
    "State": "disassociating",  
    "AssociationId": "iip-assoc-05020b59952902f5f",  
    "IamInstanceProfile": {  
      "Id": "AIPAI5IVIHMFFYY2DKV5Y",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateIamInstanceProfile](#)」の「」を参照してください。

disassociate-instance-event-window

次のコード例は、disassociate-instance-event-window を使用する方法を示しています。

AWS CLI

例 1: イベントウィンドウから 1 つ以上のインスタンスの関連付けを解除するには

次のdisassociate-instance-event-window例では、イベントウィンドウから 1 つ以上のインスタンスの関連付けを解除します。instance-event-window-id パラメータを指定して、イベントウィンドウを指定します。インスタンスの関連付けを解除するには、association-targetパラメータを指定し、パラメータ値には 1 つ以上のインスタンスを指定しますIDs。

```
aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

出力:

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

イベントウィンドウの制約については、「Amazon EC2ユーザーガイド」の「スケジュールされたイベント」セクションの[「考慮事項」](#)を参照してください。

例 2: イベントウィンドウからインスタスタグの関連付けを解除するには

次のdisassociate-instance-event-window例では、イベントウィンドウからインスタスタグの関連付けを解除します。instance-event-window-id パラメータを指定して、イベントウィンドウを指定します。インスタスタグの関連付けを解除するには、association-target パラメータを使用し、その値に 1 つ以上のタグを指定します。

```
aws ec2 disassociate-instance-event-window \  
--region us-east-1 \  
--instance-event-window-id iew-0abcdef1234567890 \  
--association-target "InstanceTags=[{Key=k2, Value=v2}, {Key=k1, Value=v1}]"
```

出力:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

イベントウィンドウの制約については、「Amazon EC2ユーザーガイド」の「スケジュールされたイベント」セクションの[「考慮事項」](#)を参照してください。

例 3: イベントウィンドウから Dedicated Host の関連付けを解除するには

次のdisassociate-instance-event-window例では、Dedicated Host をイベントウィンドウから関連付け解除します。instance-event-window-id パラメータを指定して、イベントウィンドウを指定します。Dedicated Host の関連付けを解除するには、association-targetパラメータを指定し、パラメータ値には 1 つ以上の Dedicated Host を指定しますIDs。

```
aws ec2 disassociate-instance-event-window \  
--region us-east-1 \  
--instance-event-window-id iew-0abcdef1234567890 \  
--association-target DedicatedHostIds=h-029fa35a02b99801d
```

出力:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",
```

```
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの「[考慮事項](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateInstanceEventWindow](#)」の「」を参照してください。

disassociate-ipam-resource-discovery

次のコード例は、disassociate-ipam-resource-discovery を使用する方法を示しています。

AWS CLI

からリソース検出の関連付けを解除するには IPAM

この例では、IPAM委任された管理者アカウントであり、 からIPAMリソース検出の関連付けを解除しますIPAM。describe コマンドを実行し、 "ResourceDiscoveryStatus": "not-found" と の関連付けを解除IPAMして、他の関連付けのためのスペースを確保したいことに気づきました。

次のdisassociate-ipam-resource-discovery例では、 AWS アカウントのIPAMリソース検出の関連付けを解除します。

```
aws ec2 disassociate-ipam-resource-discovery \
  --ipam-resource-discovery-association-id ipam-res-disco-assoc-04382a6346357cf82 \
  --region us-east-1
```

出力:

```
{
```

```
"IpamResourceDiscoveryAssociation": {
  "OwnerId": "320805250157",
  "IpamResourceDiscoveryAssociationId": "ipam-res-disco-
assoc-04382a6346357cf82",
  "IpamResourceDiscoveryAssociationArn":
  "arn:aws:ec2::320805250157:ipam-resource-discovery-association/ipam-res-disco-
assoc-04382a6346357cf82",
  "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
  "IpamId": "ipam-005f921c17ebd5107",
  "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
  "IpamRegion": "us-east-1",
  "IsDefault": false,
  "ResourceDiscoveryStatus": "not-found",
  "State": "disassociate-in-progress"
}
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の[「組織外のアカウントIPAMとの統合」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateIpamResourceDiscovery](#)」の「」を参照してください。

disassociate-nat-gateway-address

次のコード例は、disassociate-nat-gateway-address を使用する方法を示しています。

AWS CLI

パブリックNATゲートウェイから Elastic IP アドレスの関連付けを解除するには

次のdisassociate-nat-gateway-address例では、指定された Elastic IP アドレスと指定されたパブリックNATゲートウェイの関連付けを解除します。

```
aws ec2 disassociate-nat-gateway-address \
  --nat-gateway-id nat-1234567890abcdef0 \
  --association-ids eipassoc-0f96bdca17EXAMPLE
```

出力:

```
{
```

```
"NatGatewayId": "nat-1234567890abcdef0",
"NatGatewayAddresses": [
  {
    "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
    "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
    "PrivateIp": "10.0.0.74",
    "PublicIp": "3.211.231.218",
    "AssociationId": "eipassoc-0f96bdca17EXAMPLE",
    "IsPrimary": false,
    "Status": "disassociating"
  }
]
```

詳細については、「Amazon VPCユーザーガイド」の[NAT「ゲートウェイ」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DisassociateNatGatewayAddress](#)」の「」を参照してください。

disassociate-route-table

次のコード例は、disassociate-route-table を使用する方法を示しています。

AWS CLI

ルートテーブルの関連付けを解除するには

この例では、指定されたサブネットから指定されたルートテーブルの関連付けを解除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- API 詳細については、AWS CLI「コマンドリファレンス[DisassociateRouteTable](#)」の「」を参照してください。

disassociate-subnet-cidr-block

次のコード例は、disassociate-subnet-cidr-block を使用する方法を示しています。

AWS CLI

サブネットからIPv6CIDRブロックの関連付けを解除するには

この例では、IPv6CIDRブロックの関連付け ID を使用して、サブネットからCIDRブロックの関連付けを解除します。

コマンド:

```
aws ec2 disassociate-subnet-cidr-block --association-id subnet-cidr-assoc-3aa54053
```

出力:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
    "AssociationId": "subnet-cidr-assoc-3aa54053",
    "Ipv6CidrBlockState": {
      "State": "disassociating"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateSubnetCidrBlock](#)」の「」を参照してください。

disassociate-transit-gateway-multicast-domain

次の例は、disassociate-transit-gateway-multicast-domain を使用する方法を説明しています。

AWS CLI

マルチキャストドメインからサブネットの関連付けを解除するには

次のdisassociate-transit-gateway-multicast-domain例では、指定されたマルチキャストドメインからサブネットの関連付けを解除します。

```
aws ec2 disassociate-transit-gateway-multicast-domain \
  --transit-gateway-attachment-id tgw-attach-070e571cd1EXAMPLE \
```

```
--subnet-id subnet-000de86e3bEXAMPLE \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

出力:

```
{  
  "Associations": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",  
    "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",  
    "ResourceId": "vpc-7EXAMPLE",  
    "ResourceType": "vpc",  
    "Subnets": [  
      {  
        "SubnetId": "subnet-000de86e3bEXAMPLE",  
        "State": "disassociating"  
      }  
    ]  
  }  
}
```

詳細については、「トランジットゲートウェイガイド」の[「マルチキャストの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateTransitGatewayMulticastDomain](#)」の「」を参照してください。

disassociate-transit-gateway-route-table

次のコード例は、disassociate-transit-gateway-route-table を使用する方法を示しています。

AWS CLI

トランジットゲートウェイルートテーブルとリソースアタッチメントの関連付けを解除するには

次のdisassociate-transit-gateway-route-table例では、指定されたアタッチメントとトランジットゲートウェイルートテーブルの関連付けを解除します。

```
aws ec2 disassociate-transit-gateway-route-table \  
--transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \  
--transit-gateway-attachment-id tgw-attach-08e0bc912cEXAMPLE
```

出力:

```
{
  "Association": {
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
    "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
    "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "ResourceType": "direct-connect-gateway",
    "State": "disassociating"
  }
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateTransitGatewayRouteTable](#)」の「」を参照してください。

disassociate-vpc-cidr-block

次のコード例は、disassociate-vpc-cidr-block を使用する方法を示しています。

AWS CLI

からIPv6CIDRブロックの関連付けを解除するには VPC

この例では、IPv6CIDRブロックの関連付け ID VPCを使用して、 からCIDRブロックの関連付けを解除します。

コマンド:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-eca54085
```

出力:

```
{
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
    "AssociationId": "vpc-cidr-assoc-eca54085",
    "Ipv6CidrBlockState": {
      "State": "disassociating"
    }
  }
}
```

```
    }  
  },  
  "VpcId": "vpc-a034d6c4"  
}
```

からIPv4CIDRブロックの関連付けを解除するには VPC

この例では、IPv4CIDRブロックと の関連付けを解除しますVPC。

コマンド:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-0287ac6b
```

出力:

```
{  
  "CidrBlockAssociation": {  
    "AssociationId": "vpc-cidr-assoc-0287ac6b",  
    "CidrBlock": "172.18.0.0/16",  
    "CidrBlockState": {  
      "State": "disassociating"  
    }  
  },  
  "VpcId": "vpc-27621243"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateVpcCidrBlock](#)」の「」を参照してください。

enable-address-transfer

次のコード例は、enable-address-transfer を使用する方法を示しています。

AWS CLI

Elastic IP アドレス転送を有効にするには

次のenable-address-transfer例では、指定されたアカウントへの指定された Elastic IP アドレスの Elastic IP アドレス転送を有効にします。

```
aws ec2 enable-address-transfer \  

```

```
--allocation-id eipalloc-09ad461b0d03f6aaf \  
--transfer-account-id 123456789012
```

出力:

```
{  
  "AddressTransfer": {  
    "PublicIp": "100.21.184.216",  
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",  
    "TransferAccountId": "123456789012",  
    "TransferOfferExpirationTimestamp": "2023-02-22T20:51:01.000Z",  
    "AddressTransferStatus": "pending"  
  }  
}
```

詳細については、「Amazon VPCユーザーガイド」の「[Elastic IP アドレスを転送する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableAddressTransfer](#)」の「」を参照してください。

enable-aws-network-performance-metric-subscription

次のコード例は、enable-aws-network-performance-metric-subscription を使用する方法を示しています。

AWS CLI

メトリクスサブスクリプションを有効にするには

次のenable-aws-network-performance-metric-subscription例では、指定された送信元リージョンと送信先リージョン間のネットワークレイテンシーの集計をモニタリングできません。

```
aws ec2 enable-aws-network-performance-metric-subscription \  
--source us-east-1 \  
--destination eu-west-1 \  
--metric aggregate-latency \  
--statistic p50
```

出力:

```
{
  "Output": true
}
```

詳細については、「Infrastructure Performance ユーザーガイド」の[「サブスクリプションの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableAwsNetworkPerformanceMetricSubscription](#)」の「」を参照してください。

enable-efs-encryption-by-default

次のコード例は、enable-efs-encryption-by-default を使用する方法を示しています。

AWS CLI

EFS暗号化をデフォルトで有効にするには

次のenable-efs-encryption-by-default例では、現在のリージョンの AWS アカウントでデフォルトでEFS暗号化を有効にします。

```
aws efs enable-efs-encryption-by-default
```

出力:

```
{
  "EfsEncryptionByDefault": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableEfsEncryptionByDefault](#)」の「」を参照してください。

enable-fast-launch

次の例は、enable-fast-launch を使用する方法を説明しています。

AWS CLI

イメージの高速起動を開始するには

次のenable-fast-launch例では、指定された で高速起動を開始しAMI、起動する並列インスタンスの最大数を 6 に設定します。の事前プロビジョニングに使用するリソースのタイプAMIは snapshot に設定されます。これはデフォルト値でもあります。

```
aws ec2 enable-fast-launch \  
  --image-id ami-01234567890abcdef \  
  --max-parallel-launches 6 \  
  --resource-type snapshot
```

出力:

```
{  
  "ImageId": "ami-01234567890abcdef",  
  "ResourceType": "snapshot",  
  "SnapshotConfiguration": {  
    "TargetResourceCount": 10  
  },  
  "LaunchTemplate": {},  
  "MaxParallelLaunches": 6,  
  "OwnerId": "0123456789123",  
  "State": "enabling",  
  "StateTransitionReason": "Client.UserInitiated",  
  "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"  
}
```

起動を高速化AMIするための Windows の設定の詳細については、「Amazon EC2ユーザーガイド」の「[起動を高速化AMIするための の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableFastLaunch](#)」の「」を参照してください。

enable-fast-snapshot-restores

次のコード例は、enable-fast-snapshot-restores を使用する方法を示しています。

AWS CLI

高速スナップショット復元を有効にするには

次のenable-fast-snapshot-restores例では、指定されたアベイラビリティーゾーンで指定されたスナップショットの高速スナップショット復元を有効にします。

```
aws ec2 enable-fast-snapshot-restores \  
  --availability-zones us-east-2a us-east-2b \  
  --source-snapshot-ids snap-1234567890abcdef0
```

出力:

```
{  
  "Successful": [  
    {  
      "SnapshotId": "snap-1234567890abcdef0"  
      "AvailabilityZone": "us-east-2a",  
      "State": "enabling",  
      "StateTransitionReason": "Client.UserInitiated",  
      "OwnerId": "123456789012",  
      "EnablingTime": "2020-01-25T23:57:49.602Z"  
    },  
    {  
      "SnapshotId": "snap-1234567890abcdef0"  
      "AvailabilityZone": "us-east-2b",  
      "State": "enabling",  
      "StateTransitionReason": "Client.UserInitiated",  
      "OwnerId": "123456789012",  
      "EnablingTime": "2020-01-25T23:57:49.596Z"  
    }  
  ],  
  "Unsuccessful": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableFastSnapshotRestores](#)」の「」を参照してください。

enable-image-block-public-access

次のコード例は、enable-image-block-public-access を使用する方法を示しています。

AWS CLI

指定されたリージョンAMIsでのパブリックアクセスのブロックを有効にするには

次のenable-image-block-public-access例ではAMIs、指定されたリージョンのアカウントレベルでのパブリックアクセスのブロックを有効にします。


```
aws ec2 enable-image-block-public-access \  
  --region us-east-1 \  
  --image-block-public-access-state block-new-sharing
```

出力:

```
{  
  "ImageBlockPublicAccessState": "block-new-sharing"  
}
```

詳細については、「Amazon ユーザーガイド」の「[へのパブリックアクセスのブロックAMIs](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableImageBlockPublicAccess](#)」の「」を参照してください。

enable-image-deprecation

次のコード例は、enable-image-deprecation を使用する方法を示しています。

AWS CLI

例 1: を非推奨にするには AMI

次のenable-image-deprecation例では、特定の日時AMIのを廃止しています。秒の値を指定すると、Amazon は秒を最も近い分にEC2四捨五入します。この手順を実行するには、AMI所有者である必要があります。

```
aws ec2 enable-image-deprecation \  
  --image-id ami-1234567890abcdef0 \  
  --deprecate-at "2022-10-15T13:17:12.000Z"
```

出力:

```
{  
  "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",  
  "Return": "true"  
}
```

詳細については、「Amazon EC2ユーザーガイド」のAMI「<<https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami>>の廃止」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableImageDeprecation](#)」の「」を参照してください。

enable-image

次の例は、enable-image を使用する方法を説明しています。

AWS CLI

を有効にするには AMI

次のenable-image例では、指定された を有効にしますAMI。

```
aws ec2 enable-image \  
  --image-id ami-1234567890abcdef0
```

出力:

```
{  
  "Return": "true"  
}
```

詳細については、「Amazon ユーザーガイド」の「[を無効にするAMI](#)」を参照してください。

EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableImage](#)」の「」を参照してください。

enable-ipam-organization-admin-account

次の例は、enable-ipam-organization-admin-account を使用する方法を説明しています。

AWS CLI

AWS Organizations と統合し、メンバーアカウントをIPAMアカウントとして委任するには

次のenable-ipam-organization-admin-account例では、AWS Organizations IPAM と統合し、メンバーアカウントをIPAMアカウントとして委任します。

```
aws ec2 enable-ipam-organization-admin-account \  
  --delegated-admin-account-id 320805250157
```

出力:

```
{
  "Success": true
}
```

詳細については、「[Amazon VPCIPAMユーザーガイド](#)」の [AWS 「組織IPAMとの統合」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスEnableIpamOrganizationAdminAccount](#)」の「」を参照してください。

enable-reachability-analyzer-organization-sharing

次のコード例は、enable-reachability-analyzer-organization-sharing を使用する方法を示しています。

AWS CLI

Reachability Analyzer の信頼されたアクセスを有効にするには

次のenable-reachability-analyzer-organization-sharing例では、Reachability Analyzer の信頼されたアクセスを有効にします。

```
aws ec2 enable-reachability-analyzer-organization-sharing
```

このコマンドでは何も出力されません。

詳細については、「[Reachability Analyzer ユーザーガイド](#)」の [「クロスアカウント分析」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスEnableReachabilityAnalyzerOrganizationSharing](#)」の「」を参照してください。

enable-serial-console-access

次の例は、enable-serial-console-access を使用する方法を説明しています。

AWS CLI

アカウントのシリアルコンソールへのアクセスを有効にするには

次のenable-serial-console-access例では、シリアルコンソールへのアカウントアクセスを有効にします。

```
aws ec2 enable-serial-console-access
```

出力:

```
{
  "SerialConsoleAccessEnabled": true
}
```

詳細については、「Amazon ユーザーガイド」の[EC2「シリアルコンソール」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableSerialConsoleAccess](#)」の「」を参照してください。

enable-snapshot-block-public-access

次の例は、enable-snapshot-block-public-access を使用する方法を説明しています。

AWS CLI

スナップショットのパブリックアクセスのブロックを有効にするには

次のenable-snapshot-block-public-access例では、スナップショットのすべてのパブリック共有をブロックします。

```
aws ec2 enable-snapshot-block-public-access \
  --state block-all-sharing
```

出力:

```
{
  "State": "block-all-sharing"
}
```

詳細については、「Amazon EBSユーザーガイド」の[「スナップショットのパブリックアクセスのブロック」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableSnapshotBlockPublicAccess](#)」の「」を参照してください。

enable-transit-gateway-route-table-propagation

次のコード例は、enable-transit-gateway-route-table-propagation を使用する方法を示しています。

AWS CLI

トランジットゲートウェイアタッチメントが指定された伝播ルートテーブルにルートを伝播できるようにするには

次のenable-transit-gateway-route-table-propagation例では、指定されたアタッチメントが指定された伝播ルートテーブルにルートを伝播できるようにします。

```
aws ec2 enable-transit-gateway-route-table-propagation \  
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

出力:

```
{  
  "Propagation": {  
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",  
    "ResourceId": "vpc-4d7de228",  
    "ResourceType": "vpc",  
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",  
    "State": "disabled"  
  }  
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableTransitGatewayRouteTablePropagation](#)」の「」を参照してください。

enable-vgw-route-propagation

次の例は、enable-vgw-route-propagation を使用する方法を説明しています。

AWS CLI

ルート伝播を有効にするには

この例では、指定された仮想プライベートゲートウェイが静的ルートを指定されたルートテーブルに伝達できるようにします。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableVgwRoutePropagation](#)」の「」を参照してください。

enable-volume-io

次のコード例は、enable-volume-io を使用する方法を示しています。

AWS CLI

ボリュームの I/O を有効にするには

この例では、ボリュームの I/O を有効にします `vol-1234567890abcdef0`。

コマンド:

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

出力:

```
{  
  "Return": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableVolumelo](#)」の「」を参照してください。

enable-vpc-classic-link-dns-support

次のコード例は、enable-vpc-classic-link-dns-support を使用する方法を示しています。

AWS CLI

のサポートを有効にする ClassicLink DNSには VPC

この例では、のサポートを有効にします ClassicLink DNSvpc-88888888。

コマンド:

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableVpcClassicLinkDnsSupport](#)」の「」を参照してください。

enable-vpc-classic-link

次のコード例は、enable-vpc-classic-link を使用する方法を示しています。

AWS CLI

VPCの を有効にするには ClassicLink

この例では、の vpc-88888888 を有効にします ClassicLink。

コマンド:

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableVpcClassicLink](#)」の「」を参照してください。

export-client-vpn-client-certificate-revocation-list

次のコード例は、`export-client-vpn-client-certificate-revocation-list` を使用する方法を示しています。

AWS CLI

クライアント証明書失効リストをエクスポートするには

次の`export-client-vpn-client-certificate-revocation-list`例では、指定されたクライアントVPNエンドポイントのクライアント証明書失効リストをエクスポートします。この例では、読みやすくするために、出力はテキスト形式で返されます。

```
aws ec2 export-client-vpn-client-certificate-revocation-list \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --output text
```

出力:

```
-----BEGIN X509 CRL-----  
MIICiTCCAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMC  
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAcTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWx1eHAd  
BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI1MjA0NTIxWjCBiDELMaKGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAcTC0lBTSBDb25z  
b2x1MRIwEAYDVQQDEw1UZXN0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb25lQGFT  
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzSzwY6786m86gpE  
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb  
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStb  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=  
-----END X509 CRL-----  
STATUS      pending
```

詳細については、クライアントAWS VPN管理者ガイドの「クライアント[証明書失効リスト](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ExportClientVpnClientCertificateRevocationList](#)」の「」を参照してください。

export-client-vpn-client-configuration

次のコード例は、`export-client-vpn-client-configuration` を使用する方法を示しています。

AWS CLI

クライアント設定をエクスポートするには

次の`export-client-vpn-client-configuration`例では、指定されたクライアントVPNエンドポイントのクライアント設定をエクスポートします。この例では、読みやすくするために、出力はテキスト形式で返されます。

```
aws ec2 export-client-vpn-client-configuration \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --output text
```

出力:

```
client  
dev tun  
proto udp  
remote cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-south-1.amazonaws.com 443  
remote-random-hostname  
resolv-retry infinite  
nobind  
persist-key  
persist-tun  
remote-cert-tls server  
cipher AES-256-GCM  
verb 3  
<ca>  
-----BEGIN CERTIFICATE-----  
MIICiTCCAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC  
VVMxCzAJBgNVBAgTAldBMRAdDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd  
BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI1MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAdDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z  
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb251QGFT  
YXpvbi5jb20wZG8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
```

```
Ibb30hjZncvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----
</ca>
reneg-sec 0
```

詳細については、[「クライアント管理者ガイド」の「クライアントVPNエンドポイント」](#)を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[ExportClientVpnClientConfiguration](#)」の「」を参照してください。

export-image

次の例は、`export-image` を使用する方法を説明しています。

AWS CLI

から VM をエクスポートするには AMI

次の`export-image`例では、指定された AMI を指定された形式で指定されたバケットにエクスポートします。

```
aws ec2 export-image \
  --image-id ami-1234567890abcdef0 \
  --disk-image-format VMDK \
  --s3-export-location S3Bucket=my-export-bucket,S3Prefix=exports/
```

出力:

```
{
  "DiskImageFormat": "vmdk",
  "ExportImageTaskId": "export-ami-1234567890abcdef0"
  "ImageId": "ami-1234567890abcdef0",
  "RoleName": "vmimport",
  "Progress": "0",
  "S3ExportLocation": {
    "S3Bucket": "my-export-bucket",
    "S3Prefix": "exports/"
  },
  "Status": "active",
```

```
"StatusMessage": "validating"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ExportImage](#)」の「」を参照してください。

get-associated-ipv6-pool-cidrs

次の例は、get-associated-ipv6-pool-cidrs を使用する方法を説明しています。

AWS CLI

IPv6 アドレスプールの関連付けを取得するには

次のget-associated-ipv6-pool-cidrs例では、指定されたIPv6アドレスプールの関連付けを取得します。

```
aws ec2 get-associated-ipv6-pool-cidrs \
  --pool-id ipv6pool-ec2-012345abc12345abc
```

出力:

```
{
  "Ipv6CidrAssociations": [
    {
      "Ipv6Cidr": "2001:db8:1234:1a00::/56",
      "AssociatedResource": "vpc-111111222222333ab"
    }
  ]
}
```

- API 詳細については、「コマンドリファレンス」の [GetAssociatedIpv6PoolCidrs](#) を参照してください。AWS CLI

get-aws-network-performance-data

次の例は、get-aws-network-performance-data を使用する方法を説明しています。

AWS CLI

ネットワークパフォーマンスデータを取得するには

次のget-aws-network-performance-data例では、指定された期間内の指定されたリージョン間のネットワークパフォーマンスに関するデータを取得します。

```
aws ec2 get-aws-network-performance-data \  
  --start-time 2022-10-26T12:00:00.000Z \  
  --end-time 2022-10-26T12:30:00.000Z \  
  --data-queries Id=my-query,Source=us-east-1,Destination=eu-west-1,Metric=aggregate-latency,Statistic=p50,Period=five-minutes
```

出力:

```
{  
  "DataResponses": [  
    {  
      "Id": "my-query",  
      "Source": "us-east-1",  
      "Destination": "eu-west-1",  
      "Metric": "aggregate-latency",  
      "Statistic": "p50",  
      "Period": "five-minutes",  
      "MetricPoints": [  
        {  
          "StartDate": "2022-10-26T12:00:00+00:00",  
          "EndDate": "2022-10-26T12:05:00+00:00",  
          "Value": 62.44349,  
          "Status": "OK"  
        },  
        {  
          "StartDate": "2022-10-26T12:05:00+00:00",  
          "EndDate": "2022-10-26T12:10:00+00:00",  
          "Value": 62.483498,  
          "Status": "OK"  
        },  
        {  
          "StartDate": "2022-10-26T12:10:00+00:00",  
          "EndDate": "2022-10-26T12:15:00+00:00",  
          "Value": 62.51248,  
          "Status": "OK"  
        },  
        {  
          "StartDate": "2022-10-26T12:15:00+00:00",  
          "EndDate": "2022-10-26T12:20:00+00:00",  
          "Value": 62.635475,  
          "Status": "OK"  
        }  
      ]  
    }  
  ]  
}
```

```
        "Status": "OK"
      },
      {
        "StartDate": "2022-10-26T12:20:00+00:00",
        "EndDate": "2022-10-26T12:25:00+00:00",
        "Value": 62.733974,
        "Status": "OK"
      },
      {
        "StartDate": "2022-10-26T12:25:00+00:00",
        "EndDate": "2022-10-26T12:30:00+00:00",
        "Value": 62.773975,
        "Status": "OK"
      },
      {
        "StartDate": "2022-10-26T12:30:00+00:00",
        "EndDate": "2022-10-26T12:35:00+00:00",
        "Value": 62.75349,
        "Status": "OK"
      }
    ]
  }
]
```

詳細については、「[インフラストラクチャパフォーマンスユーザーガイド](#)」の「[ネットワークパフォーマンスのモニタリング](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetAwsNetworkPerformanceData](#)」の「」を参照してください。

get-capacity-reservation-usage

次のコード例は、get-capacity-reservation-usage を使用する方法を示しています。

AWS CLI

AWS アカウント間でキャパシティ予約の使用状況を表示するには

次のget-capacity-reservation-usage例は、指定されたキャパシティ予約の使用状況情報を表示します。

```
aws ec2 get-capacity-reservation-usage \
```

```
--capacity-reservation-id cr-1234abcd56EXAMPLE
```

出力:

```
{
  "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
  "InstanceUsages": [
    {
      "UsedInstanceCount": 1,
      "AccountId": "123456789012"
    }
  ],
  "AvailableInstanceCount": 4,
  "TotalInstanceCount": 5,
  "State": "active",
  "InstanceType": "t2.medium"
}
```

詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「共有キャパシティ予約使用状況の表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCapacityReservationUsage](#)」の「」を参照してください。

get-coip-pool-usage

次のコード例は、get-coip-pool-usage を使用する方法を示しています。

AWS CLI

顧客所有の IP アドレスプールの使用状況を取得するには

次のget-coip-pool-usage例では、指定されたカスタマー所有の IP アドレスプールの使用状況の詳細を取得します。

```
aws ec2 get-coip-pool-usage \  
  --pool-id ipv4pool-coip-123a45678bEXAMPLE
```

出力:

```
{
```

```
"CoipPoolId": "ipv4pool-coip-123a45678bEXAMPLE",
"CoipAddressUsages": [
  {
    "CoIp": "0.0.0.0"
  },
  {
    "AllocationId": "eipalloc-123ab45c6dEXAMPLE",
    "AwsAccountId": "123456789012",
    "CoIp": "0.0.0.0"
  },
  {
    "AllocationId": "eipalloc-123ab45c6dEXAMPLE",
    "AwsAccountId": "123456789111",
    "CoIp": "0.0.0.0"
  }
],
"LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
}
```

CoIP の詳細については、AWS Outposts ユーザーガイドの「[カスタマー所有の IP アドレス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCoipPoolUsage](#)」の「」を参照してください。

get-console-output

次のコード例は、get-console-output を使用する方法を示しています。

AWS CLI

例 1: コンソール出力を取得するには

次のget-console-output例では、指定された Linux インスタンスのコンソール出力を取得します。

```
aws ec2 get-console-output \
  --instance-id i-1234567890abcdef0
```

出力:

```
{
```

```
"InstanceId": "i-1234567890abcdef0",
"Timestamp": "2013-07-25T21:23:53.000Z",
"Output": "..."}
}
```

詳細については、「Amazon ユーザーガイド」の「[インスタンスコンソール出力](#)」を参照してください。 EC2

例 2: 最新のコンソール出力を取得するには

次のget-console-output例では、指定された Linux インスタンスの最新のコンソール出力を取得します。

```
aws ec2 get-console-output \
  --instance-id i-1234567890abcdef0 \
  --latest \
  --output text
```

出力:

```
i-1234567890abcdef0 [ 0.000000] Command line: root=LABEL=/ console=tty1
console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point
registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
...
Cloud-init v. 0.7.6 finished at Wed, 09 May 2018 19:01:13 +0000. Datasource
DataSourceEc2. Up 21.50 seconds
Amazon Linux AMI release 2018.03
Kernel 4.14.26-46.32.amzn1.x
```

詳細については、「Amazon ユーザーガイド」の「[インスタンスコンソール出力](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConsoleOutput](#)」の「」を参照してください。

get-console-screenshot

次のコード例は、get-console-screenshot を使用する方法を示しています。

AWS CLI

実行中のインスタンスのスクリーンショットを取得するには

次の`get-console-screenshot`例では、指定されたインスタンスのスクリーンショットを `.jpg` 形式で取得します。スクリーンショットは Base64-encodedされた文字列として返されます。

```
aws ec2 get-console-screenshot \  
  --instance-id i-1234567890abcdef0
```

出力:

```
{  
  "ImageData": "997987/8kgj49ikjhewkww0008084EXAMPLE",  
  "InstanceId": "i-1234567890abcdef0"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConsoleScreenshot](#)」の「」を参照してください。

get-default-credit-specification

次の例は、`get-default-credit-specification` を使用する方法を説明しています。

AWS CLI

デフォルトのクレジットオプションを記述するには

次の`get-default-credit-specification`例では、T2 インスタンスのデフォルトのクレジットオプションについて説明します。

```
aws ec2 get-default-credit-specification \  
  --instance-family t2
```

出力:

```
{  
  "InstanceFamilyCreditSpecification": {  
    "InstanceFamily": "t2",  
    "CpuCredits": "standard"  
  }
```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDefaultCreditSpecification](#)」の「」を参照してください。

get-ebs-default-kms-key-id

次の例は、get-ebs-default-kms-key-id を使用する方法を説明しています。

AWS CLI

EBS暗号化CMKのデフォルトを記述するには

次のget-ebs-default-kms-key-id例では、AWS アカウントのEBS暗号化CMKのデフォルトについて説明します。

```
aws ec2 get-ebs-default-kms-key-id
```

出力にはEBS暗号化CMKのデフォルトが表示されます。これは、エイリアス CMKで AWS 管理される ですalias/aws/ebs。

```
{  
  "KmsKeyId": "alias/aws/ebs"  
}
```

次の出力は、EBS暗号化CMKのカスタムを示しています。

```
{  
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEbsDefaultKmsKeyId](#)」の「」を参照してください。

get-ebs-encryption-by-default

次の例は、get-ebs-encryption-by-default を使用する方法を説明しています。

AWS CLI

EBS暗号化がデフォルトで有効になっているかどうかを記述するには

次の`get-ebs-encryption-by-default`例は、現在のリージョンの AWS アカウントでEBS暗号化がデフォルトで有効になっているかどうかを示しています。

```
aws ec2 get-ebs-encryption-by-default
```

次の出力は、EBS暗号化がデフォルトで無効になっていることを示しています。

```
{
  "EbsEncryptionByDefault": false
}
```

次の出力は、EBS暗号化がデフォルトで有効であることを示します。

```
{
  "EbsEncryptionByDefault": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEbsEncryptionByDefault](#)」の「」を参照してください。

get-flow-logs-integration-template

次の例は、`get-flow-logs-integration-template` を使用する方法を説明しています。

AWS CLI

VPCフローログと Amazon Athena の統合を自動化する CloudFormation テンプレートを作成するには

次の`get-flow-logs-integration-template`例では、VPCフローログと Amazon Athena の統合を自動化する CloudFormation テンプレートを作成します。

Linux :

```
aws ec2 get-flow-logs-integration-template \
  --flow-log-id fl-1234567890abcdef0 \
  --config-delivery-s3-destination-arn arn:aws:s3:::DOC-EXAMPLE-BUCKET \
```

--integrate-services

```
AthenaIntegrations='[{IntegrationResultS3DestinationArn=arn:aws:s3::DOC-EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
}{IntegrationResultS3DestinationArn=arn:aws:s3::DOC-EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

Windows :

```
aws ec2 get-flow-logs-integration-template ^
--flow-log-id f1-1234567890abcdef0 ^
--config-delivery-s3-destination-arn arn:aws:s3::DOC-EXAMPLE-BUCKET ^
--integrate-
services AthenaIntegrations=[{IntegrationResultS3DestinationArn=arn:aws:s3::DOC-
EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
}{IntegrationResultS3DestinationArn=arn:aws:s3::DOC-EXAMPLE-
BUCKET,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

出力:

```
{
  "Result": "https://DOC-EXAMPLE-BUCKET.s3.us-east-2.amazonaws.com/
VPCFlowLogsIntegrationTemplate_f1-1234567890abcdef0_Wed%20Jul
%2021%2000%3A57%3A56%20UTC%202021.yml"
}
```

CloudFormation テンプレートの使用については、AWS CloudFormation 「ユーザーガイド」の [AWS CloudFormation 「テンプレートの使用」](#) を参照してください。

Amazon Athena とフローログの使用については、「Amazon Amazon Virtual Private Cloud ユーザーガイド」の [「Amazon Athena を使用したフローログのクエリ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetFlowLogsIntegrationTemplate](#)」の「」を参照してください。

get-groups-for-capacity-reservation

次のコード例は、get-groups-for-capacity-reservation を使用する方法を示しています。

AWS CLI

キャパシティー予約でリソースグループを一覧表示するには

次のget-groups-for-capacity-reservation例では、指定されたキャパシティ予約が追加されたリソースグループを一覧表示します。

```
aws ec2 get-groups-for-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

出力:

```
{
  "CapacityReservationsGroup": [
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/my-
resource-group",
      "OwnerId": "123456789012"
    }
  ]
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[キャパシティ予約の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroupsForCapacityReservation](#)」の「」を参照してください。

get-host-reservation-purchase-preview

次のコード例は、get-host-reservation-purchase-preview を使用する方法を示しています。

AWS CLI

Dedicated Host Reservation の購入プレビューを取得するには

この例では、アカウント内の指定された Dedicated Host の指定された Dedicated Host 予約のコストをプレビューします。

コマンド:

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324 --
host-id-set h-013abcd2a00cbd123
```

出力:

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
    {
      "HourlyPrice": "1.499",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "HostIdSet": [
        "h-013abcd2a00cbd123"
      ],
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetHostReservationPurchasePreview](#)」の「」を参照してください。

get-image-block-public-access-state

次のコード例は、`get-image-block-public-access-state` を使用する方法を示しています。

AWS CLI

指定されたリージョンAMIでのブロックパブリックアクセス状態を取得するには

次の`get-image-block-public-access-state`例では、指定されたリージョンAMIのアカウントレベルでのブロックパブリックアクセス状態を取得します。

```
aws ec2 get-image-block-public-access-state \
  --region us-east-1
```

出力:

```
{
  "ImageBlockPublicAccessState": "block-new-sharing"
}
```

詳細については、「Amazon ユーザーガイド」の「[へのパブリックアクセスをブロックAMIsする](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[GetImageBlockPublicAccessState](#)」の「」を参照してください。

get-instance-types-from-instance-requirements

次のコード例は、get-instance-types-from-instance-requirements を使用方法を示しています。

AWS CLI

指定された属性に一致するインスタンスタイプをプレビューするには

次のget-instance-types-from-instance-requirements例では、まず、--generate-cli-skeletonパラメータを使用して指定できるすべての属性のリストを生成し、そのリストをJSONファイルに保存します。次に、JSONファイルは、一致するインスタンスタイプをプレビューする属性をカスタマイズするために使用されます。

可能な属性をすべて生成し、出力をJSONファイルに直接保存するには、次のコマンドを使用します。

```
aws ec2 get-instance-types-from-instance-requirements \
  --region us-east-1 \
  --generate-cli-skeleton input > attributes.json
```

出力:

```
{
  "DryRun": true,
  "ArchitectureTypes": [
    "x86_64_mac"
  ],
  "VirtualizationTypes": [
    "paravirtual"
  ],
  "InstanceRequirements": {
    "VCpuCount": {
      "Min": 0,
      "Max": 0
    }
  },
}
```

```
"MemoryMiB": {
  "Min": 0,
  "Max": 0
},
"CpuManufacturers": [
  "intel"
],
"MemoryGiBPerVCpu": {
  "Min": 0.0,
  "Max": 0.0
},
"ExcludedInstanceTypes": [
  ""
],
"InstanceGenerations": [
  "current"
],
"SpotMaxPricePercentageOverLowestPrice": 0,
"OnDemandMaxPricePercentageOverLowestPrice": 0,
"BareMetal": "included",
"BurstablePerformance": "excluded",
"RequireHibernateSupport": true,
"NetworkInterfaceCount": {
  "Min": 0,
  "Max": 0
},
"LocalStorage": "required",
"LocalStorageTypes": [
  "hdd"
],
"TotalLocalStorageGB": {
  "Min": 0.0,
  "Max": 0.0
},
"BaselineEbsBandwidthMbps": {
  "Min": 0,
  "Max": 0
},
"AcceleratorTypes": [
  "inference"
],
"AcceleratorCount": {
  "Min": 0,
  "Max": 0
}
```



```

    },
    "AcceleratorManufacturers": [
      "xilinx"
    ],
    "AcceleratorNames": [
      "t4"
    ],
    "AcceleratorTotalMemoryMiB": {
      "Min": 0,
      "Max": 0
    }
  },
  "MaxResults": 0,
  "NextToken": ""
}

```

JSON ファイルを設定しま

す。ArchitectureTypes、VirtualizationTypes、VCpuCount、および MemoryMiB の値を指定する必要があります。他の属性は省略できます。省略すると、デフォルト値が使用されます。各属性とそのデフォルト値の説明については、`get-instance-types-from-instance-requirements` <<https://docs.aws.amazon.com/cli/latest/reference/ec2/get-instance-types-from-instance-requirements.html>>」を参照してください。

で指定された属性を持つインスタンスタイプをプレビューします `attributes.json`。 `--cli-input-json` パラメータを使用して、JSONファイルの名前とパスを指定します。次のリクエストでは、出力はテーブルとしてフォーマットされます。

```

aws ec2 get-instance-types-from-instance-requirements \
  --cli-input-json file://attributes.json \
  --output table

```

`attributes.json` ファイルの内容。

```

{
  "ArchitectureTypes": [
    "x86_64"
  ],
  "VirtualizationTypes": [
    "hvm"
  ],

```

```

    "InstanceRequirements": {
      "VCpuCount": {
        "Min": 4,
        "Max": 6
      },
      "MemoryMiB": {
        "Min": 2048
      },
      "InstanceGenerations": [
        "current"
      ]
    }
  }
}

```

出力:

```

-----
|GetInstanceTypesFromInstanceRequirements|
+-----+
||           InstanceTypes           ||
|+-----+|
||           InstanceType           ||
|+-----+|
|| c4.xlarge                         ||
|| c5.xlarge                         ||
|| c5a.xlarge                       ||
|| c5ad.xlarge                      ||
|| c5d.xlarge                       ||
|| c5n.xlarge                       ||
|| d2.xlarge                        ||
...

```

属性ベースのインスタンスタイプの選択の詳細については、Amazon EC2ユーザーガイドの[「属性ベースのインスタンスタイプの選択の仕組み」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceTypesFromInstanceRequirements](#)」の「」を参照してください。

get-instance-uefi-data

次のコード例は、get-instance-uefi-data を使用する方法を示しています。

AWS CLI

インスタンスからUEFIデータを取得するには

次のget-instance-uefi-data例では、インスタンスからUEFIデータを取得します。出力が空の場合、インスタンスにはUEFIデータは含まれません。

```
aws ec2 get-instance-uefi-data \  
  --instance-id i-0123456789example
```

出力:

```
{  
  "InstanceId": "i-0123456789example",  
  "UefiData": "QU1aTlVFRkkf+uLXAAAAAHj5a7fZ9+3dBzxXb/.  
  <snipped>  
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD4L/J/A0Dshho="
```

詳細については、「Amazon ユーザーガイド」の[UEFI「セキュアブート」](#)を参照してください。
EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceUefiData](#)」の「」を参照してください。

get-ipam-address-history

次のコード例は、get-ipam-address-history を使用する方法を示しています。

AWS CLI

の履歴を取得するには CIDR

次のget-ipam-address-history例では、 の履歴を取得しますCIDR。

(Linux):

```
aws ec2 get-ipam-address-history \  
  --cidr 10.0.0.0/16 \  
  --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \  
  --start-time 2021-12-08T01:00:00.000Z \  
  --end-time 2021-12-10T01:00:00.000Z
```

(Windows):

```
aws ec2 get-ipam-address-history ^  
--cidr 10.0.0.0/16 ^  
--ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^  
--start-time 2021-12-08T01:00:00.000Z ^  
--end-time 2021-12-10T01:00:00.000Z
```

出力:

```
{  
  "HistoryRecords": [  
    {  
      "ResourceOwnerId": "123456789012",  
      "ResourceRegion": "us-west-1",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-06cbefa9ee907e1c0",  
      "ResourceCidr": "10.0.0.0/16",  
      "ResourceName": "Demo",  
      "ResourceComplianceStatus": "unmanaged",  
      "ResourceOverlapStatus": "overlapping",  
      "VpcId": "vpc-06cbefa9ee907e1c0",  
      "SampledStartTime": "2021-12-08T19:54:57.675000+00:00"  
    },  
    {  
      "ResourceOwnerId": "123456789012",  
      "ResourceRegion": "us-east-2",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-042702f474812c9ad",  
      "ResourceCidr": "10.0.0.0/16",  
      "ResourceName": "test",  
      "ResourceComplianceStatus": "unmanaged",  
      "ResourceOverlapStatus": "overlapping",  
      "VpcId": "vpc-042702f474812c9ad",  
      "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"  
    },  
    {  
      "ResourceOwnerId": "123456789012",  
      "ResourceRegion": "us-east-2",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-042b8a44f64267d67",  
      "ResourceCidr": "10.0.0.0/16",  
      "ResourceName": "tester",
```

```

    "ResourceComplianceStatus": "unmanaged",
    "ResourceOverlapStatus": "overlapping",
    "VpcId": "vpc-042b8a44f64267d67",
    "SampledStartTime": "2021-12-08T19:54:59.019000+00:00"
  }
]
}

```

詳細については、「Amazon VPCIPAMユーザーガイド」の「[IP アドレスの履歴を表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIpamAddressHistory](#)」の「」を参照してください。

get-ipam-discovered-accounts

次の例は、get-ipam-discovered-accounts を使用する方法を説明しています。

AWS CLI

によって検出されたアカウントを表示するには IPAM

このシナリオでは、が検出するリソースを所有する AWS アカウントを表示したい IPAM 委任管理者 IPAM です。

--discovery-region は、モニタリング対象のアカウントステータスを表示する IPAM 運用リージョンです。例えば、3 つの IPAM 運用リージョンがある場合、このリクエストを 3 回実行して、それらの特定の各リージョンの検出に固有のタイムスタンプを表示することができます。

次の get-ipam-discovered-accounts 例では、IPAM が検出するリソースを所有する AWS アカウントを一覧表示します。

```

aws ec2 get-ipam-discovered-accounts \
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
  --discovery-region us-east-1

```

出力:

```

{
  "IpamDiscoveredAccounts": [
    {

```

```
    "AccountId": "149977607591",
    "DiscoveryRegion": "us-east-1",
    "LastAttemptedDiscoveryTime": "2024-02-09T19:04:31.379000+00:00",
    "LastSuccessfulDiscoveryTime": "2024-02-09T19:04:31.379000+00:00"
  }
]
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の[「組織外のアカウントIPAMとの統合」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIpamDiscoveredAccounts](#)」の「」を参照してください。

get-ipam-discovered-public-addresses

次の例は、get-ipam-discovered-public-addresses を使用する方法を説明しています。

AWS CLI

検出されたパブリック IP アドレスを表示するには

この例では、委任IPAMされた管理者であり、によって検出されたリソースの IP アドレスを表示できますIPAM。でリソース検出 ID を取得できます[describe-ipam-resource-discoveries](#)。

次のget-ipam-discovered-public-addresses例は、リソース検出で検出されたパブリック IP アドレスを示しています。

```
aws ec2 get-ipam-discovered-public-addresses \
  --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
  --address-region us-east-1 \
  --region us-east-1
```

出力:

```
{
  "IpamDiscoveredPublicAddresses": [
    {
      "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
      "AddressRegion": "us-east-1",
      "Address": "54.208.155.7",
      "AddressOwnerId": "320805250157",
```

```
"AssociationStatus": "associated",
"AddressType": "ec2-public-ip",
"VpcId": "vpc-073b294916198ce49",
"SubnetId": "subnet-0b6c8a8839e9a4f15",
"NetworkInterfaceId": "eni-081c446b5284a5e06",
"NetworkInterfaceDescription": "",
"InstanceId": "i-07459a6fca5b35823",
"Tags": {},
"NetworkBorderGroup": "us-east-1c",
"SecurityGroups": [
  {
    "GroupName": "launch-wizard-2",
    "GroupId": "sg-0a489dd6a65c244ce"
  }
],
"SampleTime": "2024-04-05T15:13:59.228000+00:00"
},
{
  "IpamResourceDiscoveryId": "ipam-res-disco-0f4ef577a9f37a162",
  "AddressRegion": "us-east-1",
  "Address": "44.201.251.218",
  "AddressOwnerId": "470889052923",
  "AssociationStatus": "associated",
  "AddressType": "ec2-public-ip",
  "VpcId": "vpc-6c31a611",
  "SubnetId": "subnet-062f47608b99834b1",
  "NetworkInterfaceId": "eni-024845359c2c3ae9b",
  "NetworkInterfaceDescription": "",
  "InstanceId": "i-04ef786d9c4e03f41",
  "Tags": {},
  "NetworkBorderGroup": "us-east-1a",
  "SecurityGroups": [
    {
      "GroupName": "launch-wizard-32",
      "GroupId": "sg-0ed1a426e96a68374"
    }
  ],
  "SampleTime": "2024-04-05T15:13:59.145000+00:00"
}
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の[「パブリック IP インサイトを表示する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetIpamDiscoveredPublicAddresses](#)」の「」を参照してください。

get-ipam-discovered-resource-cidrs

次の例は、get-ipam-discovered-resource-cidrs を使用する方法を説明しています。

AWS CLI

によってCIDRs検出された IP アドレスを表示するには IPAM

この例では、 が検出するCIDRsリソースの IP アドレスに関連する詳細を表示したいIPAM委任管理者IPAMです。

このリクエストを完了するには：

選択したリソース検出は、に関連付ける必要がありますIPAM。 --resource-regionは、リソースが作成された AWS リージョンです。

次のget-ipam-discovered-resource-cidrs例では、IPAMが検出するリソースの IP アドレスを一覧表示します。

```
aws ec2 get-ipam-discovered-resource-cidrs \
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
  --resource-region us-east-1
```

出力:

```
{
  {
    "IpamDiscoveredResourceCidrs": [
      {
        "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
        "ResourceRegion": "us-east-1",
        "ResourceId": "vpc-0c974c95ca7ceef4a",
        "ResourceOwnerId": "149977607591",
        "ResourceCidr": "172.31.0.0/16",
        "ResourceType": "vpc",
        "ResourceTags": [],
        "IpUsage": 0.375,
        "VpcId": "vpc-0c974c95ca7ceef4a",
```



```
    "SampleTime": "2024-02-09T19:15:16.529000+00:00"
  },
  {
    "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
    "ResourceRegion": "us-east-1",
    "ResourceId": "subnet-07fe028119082a8c1",
    "ResourceOwnerId": "149977607591",
    "ResourceCidr": "172.31.0.0/20",
    "ResourceType": "subnet",
    "ResourceTags": [],
    "IpUsage": 0.0012,
    "VpcId": "vpc-0c974c95ca7ceef4a",
    "SampleTime": "2024-02-09T19:15:16.529000+00:00"
  },
  {
    "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",
    "ResourceRegion": "us-east-1",
    "ResourceId": "subnet-0a96893763984cc4e",
    "ResourceOwnerId": "149977607591",
    "ResourceCidr": "172.31.64.0/20",
    "ResourceType": "subnet",
    "ResourceTags": [],
    "IpUsage": 0.0012,
    "VpcId": "vpc-0c974c95ca7ceef4a",
    "SampleTime": "2024-02-09T19:15:16.529000+00:00"
  }
}
}
```

詳細については、「Amazon ユーザーガイド」の「[リソース別のCIDR使用状況のモニタリング](#)」を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIpamDiscoveredResourceCidrs](#)」の「」を参照してください。

get-ipam-pool-allocations

次の例は、get-ipam-pool-allocations を使用する方法を説明しています。

AWS CLI

IPAM プールから CIDRsの割り当てを取得するには

次のget-ipam-pool-allocations例では、IPAMプールから CIDRs の割り当てを取得します。

(Linux):

```
aws ec2 get-ipam-pool-allocations \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --filters Name=ipam-pool-allocation-id,Values=ipam-pool-  
alloc-0e6186d73999e47389266a5d6991e6220
```

(Windows):

```
aws ec2 get-ipam-pool-allocations ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --filters Name=ipam-pool-allocation-id,Values=ipam-pool-  
alloc-0e6186d73999e47389266a5d6991e6220
```

出力:

```
{  
  "IpamPoolAllocations": [  
    {  
      "Cidr": "10.0.0.0/16",  
      "IpamPoolAllocationId": "ipam-pool-  
alloc-0e6186d73999e47389266a5d6991e6220",  
      "ResourceType": "custom",  
      "ResourceOwner": "123456789012"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIpamPoolAllocations](#)」の「」を参照してください。

get-ipam-pool-cidrs

次のコード例は、get-ipam-pool-cidrs を使用する方法を示しています。

AWS CLI

IPAM プールにCIDRsプロビジョニングされた を取得するには

次のget-ipam-pool-cidrs例では、 をIPAMプールにCIDRsプロビジョニングします。

(Linux):

```
aws ec2 get-ipam-pool-cidrs \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --filters 'Name=cidr,Values=10.*'
```

(Windows):

```
aws ec2 get-ipam-pool-cidrs ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --filters Name=cidr,Values=10.*
```

出力:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "10.0.0.0/24",  
    "State": "provisioned"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIpamPoolCidrs](#)」の「」を参照してください。

get-ipam-resource-cidrs

次のコード例は、get-ipam-resource-cidrs を使用する方法を示しています。

AWS CLI

リソースCIDRsに割り当てられた を取得するには

次のget-ipam-resource-cidrs例では、リソースCIDRsに割り当てられた を取得します。

(Linux):

```
aws ec2 get-ipam-resource-cidrs \  
  --ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \  
  --filters Name=management-state,Values=unmanaged
```

(Windows):

```
aws ec2 get-ipam-resource-cidrs ^  
--ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^  
--filters Name=management-state,Values=unmanaged
```

出力:

```
{  
  "IpamResourceCidrs": [  
    {  
      "IpamId": "ipam-08440e7a3acde3908",  
      "IpamScopeId": "ipam-scope-02fc38cd4c48e7d38",  
      "ResourceRegion": "us-east-2",  
      "ResourceOwnerId": "123456789012",  
      "ResourceId": "vpc-621b8709",  
      "ResourceName": "Default AWS VPC",  
      "ResourceCidr": "172.33.0.0/16",  
      "ResourceType": "vpc",  
      "ResourceTags": [  
        {  
          "Key": "Environment",  
          "Value": "Test"  
        },  
        {  
          "Key": "Name",  
          "Value": "Default AWS VPC"  
        }  
      ],  
      "IpUsage": 0.0039,  
      "ComplianceStatus": "unmanaged",  
      "ManagementState": "unmanaged",  
      "OverlapStatus": "nonoverlapping",  
      "VpcId": "vpc-621b8709"  
    }  
  ]  
}
```

詳細については、「Amazon ユーザーガイド」の [「リソース別のCIDR使用状況のモニタリング」](#) を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス [GetIpamResourceCidrs](#)」の「」を参照してください。

get-launch-template-data

次のコード例は、get-launch-template-data を使用する方法を示しています。

AWS CLI

起動テンプレートのインスタンスデータを取得するには

この例では、指定されたインスタンスに関するデータを取得し、--query オプションを使用して内のコンテンツを返します LaunchTemplateData。出力をベースとして使用して、新しい起動テンプレートや起動テンプレートのバージョンを作成できます。

コマンド:

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query  
'LaunchTemplateData'
```

出力:

```
{  
  "Monitoring": {},  
  "ImageId": "ami-8c1be5f6",  
  "BlockDeviceMappings": [  
    {  
      "DeviceName": "/dev/xvda",  
      "Ebs": {  
        "DeleteOnTermination": true  
      }  
    }  
  ],  
  "EbsOptimized": false,  
  "Placement": {  
    "Tenancy": "default",  
    "GroupName": "",  
    "AvailabilityZone": "us-east-1a"  
  },  
  "InstanceType": "t2.micro",  
  "NetworkInterfaces": [  
    {  
      "Description": "",  
      "NetworkInterfaceId": "eni-35306abc",  
      "PrivateIpAddresses": [  
        {
```

```

        "Primary": true,
        "PrivateIpAddress": "10.0.0.72"
      }
    ],
    "SubnetId": "subnet-7b16de0c",
    "Groups": [
      "sg-7c227019"
    ],
    "Ipv6Addresses": [
      {
        "Ipv6Address": "2001:db8:1234:1a00::123"
      }
    ],
    "PrivateIpAddress": "10.0.0.72"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLaunchTemplateData](#)」の「」を参照してください。

get-managed-prefix-list-associations

次の例は、get-managed-prefix-list-associations を使用する方法を説明しています。

AWS CLI

プレフィックスリストの関連付けを取得するには

次のget-managed-prefix-list-associations例では、指定されたプレフィックスリストに関連付けられているリソースを取得します。

```
aws ec2 get-managed-prefix-list-associations \
  --prefix-list-id pl-0123456abcabc1
```

出力:

```
{
  "PrefixListAssociations": [
    {
      "ResourceId": "sg-0abc123456abc12345",
      "ResourceOwner": "123456789012"
    }
  ]
}
```

```
    }
  ]
}
```

詳細については、「Amazon ユーザーガイド」の [「マネージドプレフィックスリスト」](#) を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス [GetManagedPrefixListAssociations](#)」の「」を参照してください。

get-managed-prefix-list-entries

次の例は、get-managed-prefix-list-entries を使用する方法を説明しています。

AWS CLI

プレフィックスリストのエントリを取得するには

以下は、指定されたプレフィックスリストのエントリ get-managed-prefix-list-entries を取得します。

```
aws ec2 get-managed-prefix-list-entries \
  --prefix-list-id pl-0123456abcabc1
```

出力:

```
{
  "Entries": [
    {
      "Cidr": "10.0.0.0/16",
      "Description": "vpc-a"
    },
    {
      "Cidr": "10.2.0.0/16",
      "Description": "vpc-b"
    }
  ]
}
```

詳細については、「Amazon ユーザーガイド」の [「マネージドプレフィックスリスト」](#) を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[GetManagedPrefixListEntries](#)」の「」を参照してください。

get-network-insights-access-scope-analysis-findings

次のコード例は、get-network-insights-access-scope-analysis-findings を使用する方法を示しています。

AWS CLI

Network Insights アクセススコープ分析の結果を取得するには

次のget-network-insights-access-scope-analysis-findings例では、AWS アカウントで選択したスコープ分析結果を取得します。

```
aws ec2 get-network-insights-access-scope-analysis-findings \
  --region us-east-1 \
  --network-insights-access-scope-analysis-id nis \
  --nis-123456789111
```

出力:

```
{
  "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
  "AnalysisFindings": [
    {
      "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
      "NetworkInsightsAccessScopeId": "nis-123456789111",
      "FindingComponents": [
        {
          "SequenceNumber": 1,
          "Component": {
            "Id": "eni-02e3d42d5cceca67d",
            "Arn": "arn:aws:ec2:us-east-1:936459623503:network-
interface/eni-02e3d32d9cceca17d"
          },
          "OutboundHeader": {
            "DestinationAddresses": [
              "0.0.0.0/5",
              "11.0.0.0/8",
              "12.0.0.0/6",
              "128.0.0.0/3",
```



```

        "16.0.0.0/4",
        "160.0.0.0/5",
        "168.0.0.0/6",
        "172.0.0.0/12"
        "8.0.0.0/7"
    ],
    "DestinationPortRanges": [
        {
            "From": 0,
            "To": 65535
        }
    ],
    "Protocol": "6",
    "SourceAddresses": [
        "10.0.2.253/32"
    ],
    "SourcePortRanges": [
        {
            "From": 0,
            "To": 65535
        }
    ]
}, [etc]
]
}
]
}
}

```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetNetworkInsightsAccessScopeAnalysisFindings](#)」の「」を参照してください。

get-network-insights-access-scope-content

次の例は、get-network-insights-access-scope-content を使用する方法を説明していません。

AWS CLI

Network Insights アクセススコープコンテンツを取得するには

次のget-network-insights-access-scope-content例では、AWS アカウントで選択したスコープ分析 ID の内容を取得します。

```
aws ec2 get-network-insights-access-scope-content \  
  --region us-east-1 \  
  --network-insights-access-scope-id nis-123456789222
```

出力:

```
{  
  "NetworkInsightsAccessScopeContent": {  
    "NetworkInsightsAccessScopeId": "nis-123456789222",  
    "MatchPaths": [  
      {  
        "Source": {  
          "ResourceStatement": {  
            "ResourceTypes": [  
              "AWS::EC2::NetworkInterface"  
            ]  
          }  
        },  
        "Destination": {  
          "ResourceStatement": {  
            "ResourceTypes": [  
              "AWS::EC2::InternetGateway"  
            ]  
          }  
        }  
      }  
    ]  
  }  
}
```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetNetworkInsightsAccessScopeContent](#)」の「」を参照してください。

get-password-data

次のコード例は、get-password-data を使用する方法を示しています。

AWS CLI

暗号化されたパスワードを取得するには

この例では、暗号化されたパスワードを取得します。

コマンド:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0
```

出力:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-07T22:18:38.000Z",
  "PasswordData": "gSlJFq+VpcZXqy+iktXMF6NyxQ4qCrT4+ga0uN0enX1MmgXPTj7XEXAMPLE
UQ+YeFfb+L1U4C4AKv652Ux1iRB3CPTYp7WmU3TUnhsuBd+p6LVk7T21KUm160Xbk6WPW1VYYm/TRPB1
e1DQ7PY4an/DgZT4mwcpRFigzhniQgDDe01InvSDcwoUTwNs0Y1S8ouri2W4n5GNlriM3Q0AnNVe1Vz/
53TkDtxbNoU606M1gK9zUWSxqEgwvbV2j8c5rP0WCuaMWSF14ziDu4bd7q+4RSyi8NUsVWnKZ4aEZffu
DPGzKrF5yLlf3etP2L4ZR6CvG7K1hx7VK0QVN32Dajw=="
}
```

復号されたパスワードを取得するには

この例では、復号されたパスワードを取得します。

コマンド:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0 --priv-launch-key C:
\Keys\MyKeyPair.pem
```

出力:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-30T23:18:05.000Z",
  "PasswordData": "&ViJ652e*u"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetPasswordData](#)」の「」を参照してください。

get-reserved-instances-exchange-quote

次のコード例は、`get-reserved-instances-exchange-quote` を使用する方法を示しています。

AWS CLI

コンバーティブルリザーブドインスタンスを交換するための見積りを取得するには

この例では、指定されたコンバーティブルリザーブドインスタンスのエクステンジ情報を取得します。

コマンド:

```
aws ec2 get-reserved-instances-exchange-quote --reserved-  
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-  
configurations OfferingId=6fea5434-b379-434c-b07b-a7abexample
```

出力:

```
{  
  "CurrencyCode": "USD",  
  "ReservedInstanceValueSet": [  
    {  
      "ReservedInstanceId": "7b8750c3-397e-4da4-bbcb-a45ebexample",  
      "ReservationValue": {  
        "RemainingUpfrontValue": "0.000000",  
        "HourlyPrice": "0.027800",  
        "RemainingTotalValue": "730.556200"  
      }  
    }  
  ],  
  "PaymentDue": "424.983828",  
  "TargetConfigurationValueSet": [  
    {  
      "TargetConfiguration": {  
        "InstanceCount": 5,  
        "OfferingId": "6fea5434-b379-434c-b07b-a7abexample"  
      }  
    },  
    "ReservationValue": {  
      "RemainingUpfrontValue": "424.983828",  
      "HourlyPrice": "0.016000",  
      "RemainingTotalValue": "845.447828"  
    }  
  ]  
}
```

```
    }
  }
],
"IsValidExchange": true,
"OutputReservedInstancesWillExpireAt": "2020-10-01T13:03:39Z",
"ReservedInstanceValueRollup": {
  "RemainingUpfrontValue": "0.000000",
  "HourlyPrice": "0.027800",
  "RemainingTotalValue": "730.556200"
},
"TargetConfigurationValueRollup": {
  "RemainingUpfrontValue": "424.983828",
  "HourlyPrice": "0.016000",
  "RemainingTotalValue": "845.447828"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetReservedInstancesExchangeQuote](#)」の「」を参照してください。

get-security-groups-for-vpc

次の例は、get-security-groups-for-vpc を使用する方法を説明しています。

AWS CLI

指定された のネットワークインターフェイスに関連付けることができるセキュリティグループを表示するにはVPC。

次のget-security-groups-for-vpc例は、 のネットワークインターフェイスに関連付けることができるセキュリティグループを示していますVPC。

```
aws ec2 get-security-groups-for-vpc \
  --vpc-id vpc-6c31a611 \
  --region us-east-1
```

出力:

```
{
  "SecurityGroupForVpcs": [
    {
```

```
    "Description": "launch-wizard-36 created 2022-08-29T15:59:35.338Z",
    "GroupName": "launch-wizard-36",
    "OwnerId": "470889052923",
    "GroupId": "sg-007e0c3027ee885f5",
    "Tags": [],
    "PrimaryVpcId": "vpc-6c31a611"
  },
  {
    "Description": "launch-wizard-18 created 2024-01-19T20:22:27.527Z",
    "GroupName": "launch-wizard-18",
    "OwnerId": "470889052923",
    "GroupId": "sg-0147193bef51c9eef",
    "Tags": [],
    "PrimaryVpcId": "vpc-6c31a611"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSecurityGroupsForVpc](#)」の「」を参照してください。

get-serial-console-access-status

次のコード例は、get-serial-console-access-status を使用する方法を示しています。

AWS CLI

シリアルコンソールへのアカウントアクセスのステータスを表示するには

次のget-serial-console-access-status例では、アカウントでシリアルコンソールアクセスが有効になっているかどうかを決定します。

```
aws ec2 get-serial-console-access-status
```

出力:

```
{
  "SerialConsoleAccessEnabled": true
}
```

詳細については、「Amazon ユーザーガイド」の[EC2「シリアルコンソール」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSerialConsoleAccessStatus](#)」の「」を参照してください。

get-snapshot-block-public-access-state

次の例は、`get-snapshot-block-public-access-state` を使用する方法を説明しています。

AWS CLI

スナップショットのブロックパブリックアクセスの現在の状態を取得するには

次の`get-snapshot-block-public-access-state`例では、スナップショットのパブリックアクセスブロックの現在の状態を取得します。

```
aws ec2 get-snapshot-block-public-access-state
```

出力:

```
{
  "State": "block-all-sharing"
}
```

詳細については、「Amazon EBSユーザーガイド」の[「スナップショットのパブリックアクセスをブロックする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSnapshotBlockPublicAccessState](#)」の「」を参照してください。

get-spot-placement-scores

次のコード例は、`get-spot-placement-scores` を使用する方法を示しています。

AWS CLI

指定された要件のスポットプレイスメントスコアを計算するには

次の`get-spot-placement-scores`例では、まず、パラメータを使用してスポットプレイスメントスコア設定に指定できるすべてのパラメータのリストを生成し`--generate-cli-skeleton`、そのリストを JSON ファイルに保存します。次に、JSON ファイルは、スポットプレイスメントスコアの計算に使用する要件を設定するために使用されます。

スポットプレースメントスコア設定に指定できるすべてのパラメータを生成し、出力をJSONファイルに直接保存します。

```
aws ec2 get-spot-placement-scores \  
  --region us-east-1 \  
  --generate-cli-skeleton input > attributes.json
```

出力:

```
{  
  "InstanceTypes": [  
    ""  
  ],  
  "TargetCapacity": 0,  
  "TargetCapacityUnitType": "vcpu",  
  "SingleAvailabilityZone": true,  
  "RegionNames": [  
    ""  
  ],  
  "InstanceRequirementsWithMetadata": {  
    "ArchitectureTypes": [  
      "x86_64_mac"  
    ],  
    "VirtualizationTypes": [  
      "hvm"  
    ],  
    "InstanceRequirements": {  
      "VCpuCount": {  
        "Min": 0,  
        "Max": 0  
      },  
      "MemoryMiB": {  
        "Min": 0,  
        "Max": 0  
      },  
      "CpuManufacturers": [  
        "amd"  
      ],  
      "MemoryGiBPerVCpu": {  
        "Min": 0.0,  
        "Max": 0.0  
      },  
      "ExcludedInstanceTypes": [  

```



```
    ""
  ],
  "InstanceGenerations": [
    "previous"
  ],
  "SpotMaxPricePercentageOverLowestPrice": 0,
  "OnDemandMaxPricePercentageOverLowestPrice": 0,
  "BareMetal": "excluded",
  "BurstablePerformance": "excluded",
  "RequireHibernateSupport": true,
  "NetworkInterfaceCount": {
    "Min": 0,
    "Max": 0
  },
  "LocalStorage": "included",
  "LocalStorageTypes": [
    "hdd"
  ],
  "TotalLocalStorageGB": {
    "Min": 0.0,
    "Max": 0.0
  },
  "BaselineEbsBandwidthMbps": {
    "Min": 0,
    "Max": 0
  },
  "AcceleratorTypes": [
    "fpga"
  ],
  "AcceleratorCount": {
    "Min": 0,
    "Max": 0
  },
  "AcceleratorManufacturers": [
    "amd"
  ],
  "AcceleratorNames": [
    "vu9p"
  ],
  "AcceleratorTotalMemoryMiB": {
    "Min": 0,
    "Max": 0
  }
}
```

```
  },
  "DryRun": true,
  "MaxResults": 0,
  "NextToken": ""
}
```

JSON ファイルを設定します。TargetCapacity の値を指定する必要があります。各パラメータとそのデフォルト値の説明については、「スポットプレイスメントスコアの計算 (AWS CLI) <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/spot-placement-score.html#calculate-sps-cli>>」を参照してください。

で指定された要件のスポットプレイスメントスコアを計算します attributes.json。--cli-input-json パラメータを使用して、JSON ファイルの名前とパスを指定します。

```
aws ec2 get-spot-placement-scores \
  --region us-east-1 \
  --cli-input-json file://attributes.json
```

SingleAvailabilityZone が に設定されている場合、false または が省略されている場合は出力します (省略されている場合、デフォルトで になります false)。リージョンのスコア付きリストが返されます。

```
"Recommendation": [
  {
    "Region": "us-east-1",
    "Score": 7
  },
  {
    "Region": "us-west-1",
    "Score": 5
  },
  ...
]
```

SingleAvailabilityZone が に設定されている場合は出力します true。SingleAvailability ゾーンのスコア付きリストが返されます。

```
"Recommendation": [
  {
    "Region": "us-east-1",
    "AvailabilityZoneId": "use1-az1"
  }
]
```

```
    "Score": 8
  },
  {
    "Region": "us-east-1",
    "AvailabilityZoneId": "usw2-az3"
    "Score": 6
  },
  ...
```

スポットプレースメントスコアの計算や設定例の詳細については、「Amazon EC2ユーザーガイド」の「[スポットプレースメントスコアの計算](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSpotPlacementScores](#)」の「」を参照してください。

get-subnet-cidr-reservations

次の例は、get-subnet-cidr-reservations を使用する方法を説明しています。

AWS CLI

サブネットCIDR予約に関する情報を取得するには

次のget-subnet-cidr-reservations例では、指定されたサブネットCIDR予約に関する情報を表示します。

```
aws ec2 get-subnet-cidr-reservations \
  --subnet-id subnet-03c51e2e6cEXAMPLE
```

出力:

```
{
  "SubnetIpv4CidrReservations": [
    {
      "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
      "SubnetId": "subnet-03c51e2e6cEXAMPLE",
      "Cidr": "10.1.0.16/28",
      "ReservationType": "prefix",
      "OwnerId": "123456789012"
    }
  ],
}
```

```
"SubnetIpv6CidrReservations": []
}
```

詳細については、「Amazon ユーザーガイド」の「[サブネットCIDR予約](#)」を参照してください。
VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSubnetCidrReservations](#)」の「」を参照してください。

get-transit-gateway-attachment-propagations

次のコード例は、get-transit-gateway-attachment-propagations を使用する方法を示しています。

AWS CLI

指定されたリソースアタッチメントがルートを伝達するルートテーブルを一覧表示するには

次のget-transit-gateway-attachment-propagations例は、指定されたリソースアタッチメントがルートを伝達するルートテーブルを一覧表示します。

```
aws ec2 get-transit-gateway-attachment-propagations \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE
```

出力:

```
{
  "TransitGatewayAttachmentPropagations": [
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0882c61b97EXAMPLE",
      "State": "enabled"
    }
  ]
}
```

詳細については、「[Transit Gateways ガイド](#)」の「[Transit Gateway ルートテーブル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTransitGatewayAttachmentPropagations](#)」の「」を参照してください。

get-transit-gateway-multicast-domain-associations

次の例は、get-transit-gateway-multicast-domain-associations を使用する方法を説明しています。

AWS CLI

Transit Gateway マルチキャストドメインの関連付けに関する情報を表示するには

次のget-transit-gateway-multicast-domain-associations例では、指定されたマルチキャストドメインの関連付けを返します。

```
aws ec2 get-transit-gateway-multicast-domain-associations \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

出力:

```
{  
  "MulticastDomainAssociations": [  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8EXAMPLE",  
      "ResourceId": "vpc-01128d2c24EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnet": {  
        "SubnetId": "subnet-000de86e3bEXAMPLE",  
        "State": "associated"  
      }  
    },  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",  
      "ResourceId": "vpc-7EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnet": {  
        "SubnetId": "subnet-4EXAMPLE",  
        "State": "associated"  
      }  
    },  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",  
      "ResourceId": "vpc-7EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnet": {  
        "SubnetId": "subnet-5EXAMPLE",
```

```
        "State": "associated"
      }
    },
    {
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
      "ResourceId": "vpc-7EXAMPLE",
      "ResourceType": "vpc",
      "Subnet": {
        "SubnetId": "subnet-aEXAMPLE",
        "State": "associated"
      }
    },
    {
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
      "ResourceId": "vpc-7EXAMPLE",
      "ResourceType": "vpc",
      "Subnet": {
        "SubnetId": "subnet-fEXAMPLE",
        "State": "associated"
      }
    }
  ]
}
```

詳細については、「Transit Gateways ガイド」の「[マルチキャストドメインの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetTransitGatewayMulticastDomainAssociations](#)」の「」を参照してください。

get-transit-gateway-prefix-list-references

次のコード例は、get-transit-gateway-prefix-list-references を使用する方法を示しています。

AWS CLI

トランジットゲートウェイルートテーブルでプレフィックスリストリファレンスを取得するには

次のget-transit-gateway-prefix-list-references例では、指定された Transit Gateway ルートテーブルのプレフィックスリストリファレンスを取得し、特定のプレフィックスリストの ID でフィルタリングします。

```
aws ec2 get-transit-gateway-prefix-list-references \  
--transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \  
--filters Name=prefix-list-id,Values=pl-1111112222222333
```

出力:

```
{  
  "TransitGatewayPrefixListReferences": [  
    {  
      "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",  
      "PrefixListId": "pl-1111112222222333",  
      "PrefixListOwnerId": "123456789012",  
      "State": "available",  
      "Blackhole": false,  
      "TransitGatewayAttachment": {  
        "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",  
        "ResourceType": "vpc",  
        "ResourceId": "vpc-112233445566aabbcc"  
      }  
    }  
  ]  
}
```

詳細については、「Transit Gateways ガイド」の [「プレフィックスリストリファレンス」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetTransitGatewayPrefixListReferences](#)」の「」を参照してください。

get-transit-gateway-route-table-associations

次のコード例は、get-transit-gateway-route-table-associations を使用する方法を示しています。

AWS CLI

指定された Transit Gateway ルートテーブルの関連付けに関する情報を取得するには

次のget-transit-gateway-route-table-associations例は、指定された Transit Gateway ルートテーブルの関連付けに関する情報を示しています。

```
aws ec2 get-transit-gateway-route-table-associations \  

```

```
--transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

出力:

```
{
  "Associations": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
      "ResourceId": "vpc-4d7de228",
      "ResourceType": "vpc",
      "State": "associating"
    }
  ]
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetTransitGatewayRouteTableAssociations](#)」の「」を参照してください。

get-transit-gateway-route-table-propagations

次の例は、get-transit-gateway-route-table-propagations を使用する方法を説明しています。

AWS CLI

指定された Transit Gateway ルートテーブルのルートテーブル伝播に関する情報を表示するには

次のget-transit-gateway-route-table-propagations例では、指定されたルートテーブルのルートテーブル伝播を返します。

```
aws ec2 get-transit-gateway-route-table-propagations \
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE
```

出力:

```
{
  "TransitGatewayRouteTablePropagations": [
```



```
{
  "TransitGatewayAttachmentId": "tgw-attach-01f8100bc7EXAMPLE",
  "ResourceId": "vpc-3EXAMPLE",
  "ResourceType": "vpc",
  "State": "enabled"
},
{
  "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
  "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
  "ResourceType": "direct-connect-gateway",
  "State": "enabled"
},
{
  "TransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
  "ResourceId": "8384da05-13ce-4a91-aada-5a1baEXAMPLE",
  "ResourceType": "direct-connect-gateway",
  "State": "enabled"
}
]
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTransitGatewayRouteTablePropagations](#)」の「」を参照してください。

get-verified-access-endpoint-policy

次のコード例は、get-verified-access-endpoint-policy を使用する方法を示しています。

AWS CLI

エンドポイントの検証済みアクセスポリシーを取得するには

次のget-verified-access-endpoint-policy例では、指定されたエンドポイントの検証済みアクセスポリシーを取得します。

```
aws ec2 get-verified-access-endpoint-policy \
  --verified-access-endpoint-id vae-066fac616d4d546f2
```

出力:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

詳細については、[「Verified Access ユーザーガイド」](#)の「Verified Access ポリシー」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVerifiedAccessEndpointPolicy](#)」の「」を参照してください。

get-verified-access-group-policy

次のコード例は、get-verified-access-group-policy を使用する方法を示しています。

AWS CLI

グループの Verified Access ポリシーを取得するには

次のget-verified-access-group-policy例では、指定されたグループの Verified Access ポリシーを取得します。

```
aws ec2 get-verified-access-group-policy \
  --verified-access-group-id vagr-0dbe967baf14b7235
```

出力:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

詳細については、[「Verified Access ユーザーガイド」](#)の「Verified Access グループ」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVerifiedAccessGroupPolicy](#)」の「」を参照してください。

get-vpn-connection-device-sample-configuration

次のコード例は、`get-vpn-connection-device-sample-configuration` を使用する方法を示しています。

AWS CLI

サンプル設定ファイルをダウンロードするには

次の`get-vpn-connection-device-sample-configuration`例では、指定されたサンプル設定ファイルをダウンロードします。サンプル設定ファイルを使用してゲートウェイデバイスを一覧表示するには、`get-vpn-connection-device-types` コマンドを呼び出します。

```
aws ec2 get-vpn-connection-device-sample-configuration \  
  --vpn-connection-id vpn-123456789abc01234 \  
  --vpn-connection-device-type-id 5fb390ba
```

出力:

```
{  
  "VpnConnectionDeviceSampleConfiguration": "contents-of-the-sample-configuration-file"  
}
```

詳細については、AWS Site-to-Site VPN「[ユーザーガイド](#)」の「[設定ファイルのダウンロード](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス](#)」の「[GetVpnConnectionDeviceSampleConfiguration](#)」の「」を参照してください。

get-vpn-connection-device-types

次の例は、`get-vpn-connection-device-types` を使用する方法を説明しています。

AWS CLI

サンプル設定ファイルを使用してゲートウェイデバイスを一覧表示するには

次の`get-vpn-connection-device-types`例では、サンプル設定ファイルを持つ Palo Alto Networks のゲートウェイデバイスを一覧表示します。

```
aws ec2 get-vpn-connection-device-types \  
  --query "VpnConnectionDeviceTypes[?Vendor=='Palo Alto Networks']"
```

出力:

```
[  
  {  
    "VpnConnectionDeviceTypeId": "754a6372",  
    "Vendor": "Palo Alto Networks",  
    "Platform": "PA Series",  
    "Software": "PANOS 4.1.2+"  
  },  
  {  
    "VpnConnectionDeviceTypeId": "9612cbed",  
    "Vendor": "Palo Alto Networks",  
    "Platform": "PA Series",  
    "Software": "PANOS 4.1.2+ (GUI)"  
  },  
  {  
    "VpnConnectionDeviceTypeId": "5fb390ba",  
    "Vendor": "Palo Alto Networks",  
    "Platform": "PA Series",  
    "Software": "PANOS 7.0+"  
  }  
]
```

詳細については、AWS Site-to-Site VPN「ユーザーガイド」の「[設定ファイルのダウンロード](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetVpnConnectionDeviceTypes](#)」の「」を参照してください。

import-client-vpn-client-certificate-revocation-list

次の例は、import-client-vpn-client-certificate-revocation-list を使用する方法を説明しています。

AWS CLI

クライアント証明書失効リストをインポートするには

次のimport-client-vpn-client-certificate-revocation-list例では、ローカルコンピュータ上のファイルの場所を指定して、クライアント証明書失効リストをクライアントVPNエンドポイントにインポートします。

```
aws ec2 import-client-vpn-client-certificate-revocation-list \  
  --certificate-revocation-list file:///path/to/crl.pem \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

出力:

```
{  
  "Return": true  
}
```

詳細については、クライアントAWS VPN管理者ガイドの「クライアント[証明書失効リスト](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportClientVpnClientCertificateRevocationList](#)」の「」を参照してください。

import-image

次の例は、import-image を使用する方法を説明しています。

AWS CLI

VM イメージファイルを としてインポートするには AMI

次のimport-image例では、指定された をインポートしますOVA。

```
aws ec2 import-image \  
  --disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.ova}"
```

出力:

```
{  
  "ImportTaskId": "import-ami-1234567890abcdef0",  
  "Progress": "2",
```

```
"SnapshotDetails": [  
  {  
    "DiskImageSize": 0.0,  
    "Format": "ova",  
    "UserBucket": {  
      "S3Bucket": "my-import-bucket",  
      "S3Key": "vms/my-server-vm.ova"  
    }  
  }  
],  
"Status": "active",  
"StatusMessage": "pending"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ImportImage](#)」の「」を参照してください。

import-key-pair

次の例は、import-key-pair を使用する方法を説明しています。

AWS CLI

パブリックキーをインポートするには

まず、選択したツールを使用してキーペアを生成します。例えば、次の ssh-keygen コマンドを使用します。

コマンド:

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

出力:

```
Generating public/private rsa key pair.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ec2-user/.ssh/my-key.  
Your public key has been saved in /home/ec2-user/.ssh/my-key.pub.  
...
```

この例では、指定されたパブリックキーをインポートします。

コマンド:

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

出力:

```
{
  "KeyName": "my-key",
  "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportKeyPair](#)」の「」を参照してください。

import-snapshot

次のコード例は、import-snapshot を使用する方法を示しています。

AWS CLI

スナップショットをインポートするには

次のimport-snapshot例では、指定されたディスクをスナップショットとしてインポートします。

```
aws ec2 import-snapshot \
  --description "My server VMDK" \
  --disk-container Format=VMDK,UserBucket={S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.vmdk}
```

出力:

```
{
  "Description": "My server VMDK",
  "ImportTaskId": "import-snap-1234567890abcdef0",
  "SnapshotTaskDetail": {
    "Description": "My server VMDK",
```

```
    "DiskImageSize": "0.0",
    "Format": "VMDK",
    "Progress": "3",
    "Status": "active",
    "StatusMessage": "pending"
    "UserBucket": {
      "S3Bucket": "my-import-bucket",
      "S3Key": "vms/my-server-vm.vmdk"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportSnapshot](#)」の「」を参照してください。

list-images-in-recycle-bin

次のコード例は、list-images-in-recycle-bin を使用する方法を示しています。

AWS CLI

ごみ箱内のイメージを一覧表示するには

次のlist-images-in-recycle-bin例では、ごみ箱に現在保持されているすべてのイメージを一覧表示します。

```
aws ec2 list-images-in-recycle-bin
```

出力:

```
{
  "Images": [
    {
      "RecycleBinEnterTime": "2022-03-14T15:35:08.000Z",
      "Description": "Monthly AMI One",
      "RecycleBinExitTime": "2022-03-15T15:35:08.000Z",
      "Name": "AMI_01",
      "ImageId": "ami-0111222333444abcd"
    }
  ]
}
```


詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[ごみ箱AMIsからの復旧](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListImagesInRecycleBin](#)」の「」を参照してください。

list-snapshots-in-recycle-bin

次の例は、list-snapshots-in-recycle-bin を使用する方法を説明しています。

AWS CLI

ごみ箱にスナップショットを表示するには

次のlist-snapshots-in-recycle-bin例では、スナップショット ID、スナップショットの説明、スナップショットが作成されたボリュームの ID、スナップショットが削除されてごみ箱に入った日時、保持期間が終了する日時など、ごみ箱内のスナップショットに関する情報を示します。

```
aws ec2 list-snapshots-in-recycle-bin \  
--snapshot-id snap-01234567890abcdef
```

出力:

```
{  
  "SnapshotRecycleBinInfo": [  
    {  
      "Description": "Monthly data backup snapshot",  
      "RecycleBinEnterTime": "2022-12-01T13:00:00.000Z",  
      "RecycleBinExitTime": "2022-12-15T13:00:00.000Z",  
      "VolumeId": "vol-abcdef09876543210",  
      "SnapshotId": "snap-01234567890abcdef"  
    }  
  ]  
}
```

Amazon 用ごみ箱の詳細についてはEBS、「Amazon EC2ユーザーガイド」の「[ごみ箱からスナップショットを復元する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSnapshotsInRecycleBin](#)」の「」を参照してください。

lock-snapshot

次のコード例は、lock-snapshot を使用する方法を示しています。

AWS CLI

例 1: ガバナンスモードでスナップショットをロックするには

次のlock-snapshot例では、指定されたスナップショットをガバナンスモードでロックします。

```
aws ec2 lock-snapshot \  
  --snapshot-id snap-0b5e733b4a8df6e0d \  
  --lock-mode governance \  
  --lock-duration 365
```

出力:

```
{  
  "SnapshotId": "snap-0b5e733b4a8df6e0d",  
  "LockState": "governance",  
  "LockDuration": 365,  
  "LockCreatedOn": "2024-05-05T00:56:06.208000+00:00",  
  "LockExpiresOn": "2025-05-05T00:56:06.208000+00:00",  
  "LockDurationStartTime": "2024-05-05T00:56:06.208000+00:00"  
}
```

詳細については、「Amazon ユーザーガイド」の [「スナップショットロック」](#) を参照してください。 EBS

例 2: コンプライアンスモードでスナップショットをロックするには

次のlock-snapshot例では、指定されたスナップショットをコンプライアンスモードでロックします。

```
aws ec2 lock-snapshot \  
  --snapshot-id snap-0163a8524c5b9901f \  
  --lock-mode compliance \  
  --cool-off-period 24 \  
  --lock-duration 365
```

出力:

```
{
  "SnapshotId": "snap-0b5e733b4a8df6e0d",
  "LockState": "compliance-cooloff",
  "LockDuration": 365,
  "CoolOffPeriod": 24,
  "CoolOffPeriodExpiresOn": "2024-05-06T01:02:20.527000+00:00",
  "LockCreatedOn": "2024-05-05T01:02:20.527000+00:00",
  "LockExpiresOn": "2025-05-05T01:02:20.527000+00:00",
  "LockDurationStartTime": "2024-05-05T01:02:20.527000+00:00"
}
```

詳細については、「Amazon ユーザーガイド」の「[スナップショットロック](#)」を参照してください。 EBS

- API 詳細については、AWS CLI 「コマンドリファレンス[LockSnapshot](#)」の「」を参照してください。

modify-address-attribute

次の例は、modify-address-attribute を使用する方法を説明しています。

AWS CLI

Elastic IP アドレスに関連付けられたドメイン名属性を変更するには

次のmodify-address-attribute例では、Elastic IP アドレスのドメイン名属性を変更します。

Linux :

```
aws ec2 modify-address-attribute \
  --allocation-id eipalloc-abcdef01234567890 \
  --domain-name example.com
```

Windows :

```
aws ec2 modify-address-attribute ^
  --allocation-id eipalloc-abcdef01234567890 ^
  --domain-name example.com
```

出力:

```
{
  "Addresses": [
    {
      "PublicIp": "192.0.2.0",
      "AllocationId": "eipalloc-abcdef01234567890",
      "PtrRecord": "example.net."
      "PtrRecordUpdate": {
        "Value": "example.com.",
        "Status": "PENDING"
      }
    }
  ]
}
```

保留中の変更をモニタリングし、Elastic IP アドレスの変更された属性を表示するには、[describe-addresses-attribute](#) AWS CLI コマンドリファレンスの「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyAddressAttribute](#)」の「」を参照してください。

modify-availability-zone-group

次のコード例は、`modify-availability-zone-group` を使用する方法を示しています。

AWS CLI

ゾーングループを有効にするには

次の`modify-availability-zone-group`例では、指定されたゾーングループを有効にします。

```
aws ec2 modify-availability-zone-group \
  --group-name us-west-2-lax-1 \
  --opt-in-status opted-in
```

出力:

```
{
  "Return": true
}
```

詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「リージョンとゾーン」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyAvailabilityZoneGroup](#)」の「」を参照してください。

modify-capacity-reservation-fleet

次の例は、modify-capacity-reservation-fleet を使用する方法を説明しています。

AWS CLI

例 1: キャパシティ予約フリートの合計ターゲットキャパシティを変更するには

次のmodify-capacity-reservation-fleet例では、指定されたキャパシティ予約フリートの合計ターゲット容量を変更します。キャパシティー予約フリートの総ターゲット容量を変更すると、フリートは、新しい総ターゲット容量を満たすように、自動的に新しいキャパシティー予約を作成したり、フリート内の既存のキャパシティー予約を変更またはキャンセルしたりします。フリートのステータスが modifying の間は、他の変更を試みることはできません。

```
aws ec2 modify-capacity-reservation-fleet \  
  --capacity-reservation-fleet-id crf-01234567890abcdef \  
  --total-target-capacity 160
```

出力:

```
{  
  "Return": true  
}
```

例 2: キャパシティ予約フリートの終了日を変更するには

次のmodify-capacity-reservation-fleet例では、指定されたキャパシティ予約フリートの終了日を変更します。フリートの終了日を変更すると、個々のキャパシティー予約の終了日もそれに応じて更新されます。フリートのステータスが modifying の間は、他の変更を試みることはできません。

```
aws ec2 modify-capacity-reservation-fleet \  
  --capacity-reservation-fleet-id crf-01234567890abcdef \  
  --end-date 2022-07-04T23:59:59.000Z
```

出力:

```
{
  "Return": true
}
```

キャパシティ予約フリートの詳細については、Amazon EC2ユーザーガイドの[「キャパシティ予約フリート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyCapacityReservationFleet](#)」の「」を参照してください。

modify-capacity-reservation

次のコード例は、modify-capacity-reservation を使用する方法を示しています。

AWS CLI

例 1: 既存のキャパシティ予約で予約されたインスタンスの数を変更するには

次のmodify-capacity-reservation例では、キャパシティ予約がキャパシティを予約するインスタンスの数を変更します。

```
aws ec2 modify-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE \
  --instance-count 5
```

出力:

```
{
  "Return": true
}
```

例 2: 既存のキャパシティ予約の終了日時を変更するには

次のmodify-capacity-reservation例では、既存のキャパシティ予約を指定された日時で終了するように変更します。

```
aws ec2 modify-capacity-reservation \
```

```
--capacity-reservation-id cr-1234abcd56EXAMPLE \  
--end-date-type Limited \  
--end-date 2019-08-31T23:59:59Z
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[キャパシティ予約の変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyCapacityReservation](#)」の「」を参照してください。

modify-client-vpn-endpoint

次のコード例は、modify-client-vpn-endpoint を使用する方法を示しています。

AWS CLI

クライアントVPNエンドポイントを変更するには

次のmodify-client-vpn-endpoint例では、指定されたクライアントVPNエンドポイントのクライアント接続ログ記録を有効にします。

```
aws ec2 modify-client-vpn-endpoint \  
--client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
--connection-log-options Enabled=true,CloudwatchLogGroup=ClientVPNLogs
```

出力:

```
{  
  "Return": true  
}
```

詳細については、「[クライアント管理者ガイド](#)」の「[クライアントVPNエンドポイント](#)」を参照してください。AWS VPN

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyClientVpnEndpoint](#)」の「」を参照してください。

modify-default-credit-specification

次のコード例は、modify-default-credit-specification を使用する方法を示しています。

AWS CLI

デフォルトのクレジットオプションを変更するには

次のmodify-default-credit-specification例では、T2 インスタンスのデフォルトのクレジットオプションを変更します。

```
aws ec2 modify-default-credit-specification \  
  --instance-family t2 \  
  --cpu-credits unlimited
```

出力:

```
{  
  "InstanceFamilyCreditSpecification": {  
    "InstanceFamily": "t2",  
    "CpuCredits": "unlimited"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDefaultCreditSpecification](#)」の「」を参照してください。

modify-ebs-default-kms-key-id

次のコード例は、modify-ebs-default-kms-key-id を使用する方法を示しています。

AWS CLI

EBS暗号化CMKのデフォルトを設定するには

次のmodify-ebs-default-kms-key-id例では、現在のリージョンの AWS アカウントの CMKEBS暗号化のデフォルトCMKとして指定された を設定します。

```
aws ec2 modify-ebs-default-kms-key-id \  
  --kms-key-id alias/my-cmk
```

出力:

```
{
```



```
"KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyEbsDefaultKmsKeyId](#)」の「」を参照してください。

modify-fleet

次のコード例は、modify-fleet を使用する方法を示しています。

AWS CLI

EC2フリートをスケールするには

次のmodify-fleet例では、指定されたEC2フリートのターゲット容量を変更します。指定された値が現在の容量より大きい場合、EC2フリートは追加のインスタンスを起動します。指定された値が現在の容量よりも小さい場合、EC2フリートは未処理のリクエストをキャンセルし、終了ポリシーが terminate の場合 terminate、EC2フリートは新しいターゲット容量を超えるインスタンスをすべて終了します。

```
aws ec2 modify-fleet \  
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
  --target-capacity-specification TotalTargetCapacity=5
```

出力:

```
{  
  "Return": true  
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の[EC2「フリートの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyFleet](#)」の「」を参照してください。

modify-fpga-image-attribute

次の例は、modify-fpga-image-attribute を使用する方法を説明しています。

AWS CLI

Amazon FPGAイメージの属性を変更するには

この例では、指定された のアカウント ID 123456789012のロードアクセス許可を追加します AFI。

コマンド:

```
aws ec2 modify-fpga-image-attribute --attribute LoadPermission --fpga-image-id afi-0d123e123bfc85abc --load-permission Add=[{UserId=123456789012}]
```

出力:

```
{
  "FpgaImageAttribute": {
    "FpgaImageId": "afi-0d123e123bfc85abc",
    "LoadPermissions": [
      {
        "UserId": "123456789012"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyFpgaImageAttribute](#)」の「」を参照してください。

modify-hosts

次の例は、modify-hosts を使用する方法を説明しています。

AWS CLI

例 1: Dedicated Host の自動配置を有効にするには

次のmodify-hosts例では、Dedicated Host の自動配置を有効にして、インスタンスタイプ設定に一致するターゲット外のインスタンス起動を受け入れます。

```
aws ec2 modify-hosts \  
  --host-id h-06c2f189b4EXAMPLE \  
  --
```

```
--auto-placement on
```

出力:

```
{
  "Successful": [
    "h-06c2f189b4EXAMPLE"
  ],
  "Unsuccessful": []
}
```

例 2: Dedicated Host のホストリカバリを有効にするには

次のmodify-hosts例では、指定された Dedicated Host のホストリカバリを有効にします。

```
aws ec2 modify-hosts \
  --host-id h-06c2f189b4EXAMPLE \
  --host-recovery on
```

出力:

```
{
  "Successful": [
    "h-06c2f189b4EXAMPLE"
  ],
  "Unsuccessful": []
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[専用ホストの自動配置の変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyHosts](#)」の「」を参照してください。

modify-id-format

次のコード例は、modify-id-format を使用する方法を示しています。

AWS CLI

リソースの長い ID 形式を有効にするには

次のmodify-id-format例では、instanceリソースタイプの長い ID 形式を有効にします。

```
aws ec2 modify-id-format \  
  --resource instance \  
  --use-long-ids
```

リソースの長い ID 形式を無効にするには

次のmodify-id-format例では、instanceリソースタイプの長い ID 形式を無効にします。

```
aws ec2 modify-id-format \  
  --resource instance \  
  --no-use-long-ids
```

次のmodify-id-format例では、オプトイン期間内のサポートされているすべてのリソースタイプで、より長い ID 形式を有効にします。

```
aws ec2 modify-id-format \  
  --resource all-current \  
  --use-long-ids
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyIdFormat](#)」の「」を参照してください。

modify-identity-id-format

次の例は、modify-identity-id-format を使用する方法を説明しています。

AWS CLI

IAMロールがIDsリソースに対してより長く使用できるようにするには

次のmodify-identity-id-format例では、AWS アカウントのIAMロールEC2Roleがinstanceリソースタイプに長い ID 形式を使用できるようにします。

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:role/EC2Role \  
  --resource instance \  
  --use-long-ids
```

IAMユーザーが IDs リソースをより長く使用できるようにするには

次のmodify-identity-id-format例では、AdminUser AWS アカウントのIAMユーザーがvolumeリソースタイプに長い ID 形式を使用できるようにします。

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource volume \  
  --use-long-ids
```

次のmodify-identity-id-format例では、AdminUser AWS アカウントのIAMユーザーが、オプトイン期間内のサポートされているすべてのリソースタイプで、より長い ID 形式を使用できるようにします。

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource all-current \  
  --use-long-ids
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyIdentityIdFormat](#)」の「」を参照してください。

modify-image-attribute

次のコード例は、modify-image-attribute を使用する方法を示しています。

AWS CLI

例 1: AMI公開するには

次のmodify-instance-attribute例では、指定された をAMIパブリックにします。

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{Group=all}]"
```

このコマンドでは何も出力されません。

例 2: AMIプライベートを作成するには

次のmodify-instance-attribute例では、指定された をAMIプライベートにします。

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Remove=[{Group=all}]"
```

このコマンドでは何も出力されません。

例 3: AWS アカウントに起動アクセス許可を付与するには

次のmodify-instance-attribute例では、指定された AWS アカウントに起動アクセス許可を付与します。

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{UserId=123456789012}]"
```

このコマンドでは何も出力されません。

例 4: AWS アカウントから起動許可を削除するには

次のmodify-instance-attribute例では、指定された AWS アカウントから起動許可を削除します。

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Remove=[{UserId=123456789012}]"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyImageAttribute](#)」の「」を参照してください。

modify-instance-attribute

次のコード例は、modify-instance-attribute を使用する方法を示しています。

AWS CLI

例 1: インスタンスタイプを変更するには

次のmodify-instance-attribute例では、指定されたインスタンスのインスタンスタイプを変更します。インスタンスは stopped の状態である必要があります。

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --instance-type "{\"Value\": \"m1.small\"}"
```

このコマンドでは何も出力されません。

例 2: インスタンスで拡張ネットワーキングを有効にするには

次のmodify-instance-attribute例では、指定されたインスタンスの拡張ネットワーキングを有効にします。インスタンスは stopped の状態である必要があります。

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --sriov-net-support simple
```

このコマンドでは何も出力されません。

例 3: sourceDestCheck 属性を変更するには

次のmodify-instance-attribute例では、指定されたインスタンスの sourceDestCheck 属性を に設定しますtrue。インスタンスは がある必要がありますVPC。

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-  
check "{\"Value\": true}"
```

このコマンドでは何も出力されません。

例 4: ルートボリュームの deleteOnTermination 属性を変更するには

次のmodify-instance-attribute例では、指定された Amazon EBS-backed インスタンスのルートボリュームの deleteOnTermination 属性を に設定しますfalse。デフォルトでは、この属性はルートボリュームtrue用です。

コマンド:

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\", \"Ebs\":  
{\"DeleteOnTermination\": false}}]"
```

このコマンドでは何も出力されません。

例 5: インスタンスにアタッチされたユーザーデータを変更するには

次のmodify-instance-attribute例では、UserData 指定されたインスタンスの UserData.txtとしてファイルの内容を追加します。

元のファイルの内容UserData.txt :

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

ファイルの内容は base64 でエンコードされている必要があります。最初のコマンドはテキストファイルを base64 に変換し、新しいファイルとして保存します。

コマンドの Linux/macOS バージョン :

```
base64 UserData.txt > UserData.base64.txt
```

このコマンドでは何も出力されません。

コマンドの Windows バージョン :

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >
UserData.base64.txt
```

出力:

```
Input Length = 67
Output Length = 152
CertUtil: -encode command completed successfully.
```

これで、次のCLIコマンドでそのファイルを参照できるようになりました。

```
aws ec2 modify-instance-attribute \  
  --instance-id=i-09b5a14dbca622e76 \  
  --attribute userData --value file://UserData.base64.txt
```


このコマンドでは何も出力されません。

詳細については、[「ユーザーガイド」の「ユーザーデータと AWS CLI EC2」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyInstanceAttribute](#)」の「」を参照してください。

modify-instance-capacity-reservation-attributes

次のコード例は、`modify-instance-capacity-reservation-attributes` を使用する方法を示しています。

AWS CLI

例 1: インスタンスのキャパシティ予約のターゲット設定を変更するには

次の`modify-instance-capacity-reservation-attributes`例では、停止したインスタンスを変更して、特定のキャパシティ予約をターゲットにします。

```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification  
  'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

出力:

```
{  
  "Return": true  
}
```

例 2: インスタンスのキャパシティ予約のターゲット設定を変更するには

次の`modify-instance-capacity-reservation-attributes`例では、指定されたキャパシティ予約をターゲットとする停止されたインスタンスを変更して、一致する属性 (インスタンスタイプ、プラットフォーム、アベイラビリティゾーン) とオープンインスタンスの一致基準を持つキャパシティ予約で起動します。

```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification  
  'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

```
--capacity-reservation-specification 'CapacityReservationPreference=open'
```

出力:

```
{
  "Return": true
}
```

詳細については、「Linux [インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド](#)」の「[インスタンスのキャパシティ予約設定の変更](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスModifyInstanceCapacityReservationAttributes](#)」の「」を参照してください。

modify-instance-credit-specification

次のコード例は、modify-instance-credit-specification を使用する方法を示しています。

AWS CLI

インスタンスCPUの使用に対するクレジットオプションを変更するには

この例では、指定されたリージョンで指定されたインスタンスCPUの使用に対するクレジットオプションを「無制限」に変更します。有効なクレジットオプションは、「standard」と「unlimited」です。

コマンド:

```
aws ec2 modify-instance-credit-specification --instance-credit-specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

出力:

```
{
  "SuccessfulInstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0"
    }
  ],
  "UnsuccessfulInstanceCreditSpecifications": []
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyInstanceCreditSpecification](#)」の「」を参照してください。

modify-instance-event-start-time

次の例は、`modify-instance-event-start-time` を使用する方法を説明しています。

AWS CLI

インスタンスのイベント開始時刻を変更するには

次の`modify-instance-event-start-time` コマンドは、指定されたインスタンスのイベント開始時刻を変更する方法を示しています。 `--instance-event-id` パラメータを使用してイベント ID を指定します。 `--not-before` パラメータを使用して、新しい日時を指定します。

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0
--instance-event-id instance-event-0abcdef1234567890 --not-
before 2019-03-25T10:00:00.000
```

出力:

```
"Event": {
  "InstanceEventId": "instance-event-0abcdef1234567890",
  "Code": "system-reboot",
  "Description": "scheduled reboot",
  "NotAfter": "2019-03-25T12:00:00.000Z",
  "NotBefore": "2019-03-25T10:00:00.000Z",
  "NotBeforeDeadline": "2019-04-22T21:00:00.000Z"
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「再起動が予定されているインスタンスの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyInstanceEventStartTime](#)」の「」を参照してください。

modify-instance-event-window

次の例は、`modify-instance-event-window` を使用する方法を説明しています。

AWS CLI

例 1: イベントウィンドウの時間範囲を変更するには

次のmodify-instance-event-window例では、イベントウィンドウの時間範囲を変更します。time-range パラメータにより時間範囲を変更します。また、cron-expression パラメータを指定することはできません。

```
aws ec2 modify-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --time-range StartWeekDay=monday, StartHour=2, EndWeekDay=wednesday, EndHour=8
```

出力:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "TimeRanges": [  
      {  
        "StartWeekDay": "monday",  
        "StartHour": 2,  
        "EndWeekDay": "wednesday",  
        "EndHour": 8  
      }  
    ],  
    "Name": "myEventWindowName",  
    "AssociationTarget": {  
      "InstanceIds": [  
        "i-0abcdef1234567890",  
        "i-0be35f9acb8ba01f0"  
      ],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating",  
    "Tags": [  
      {  
        "Key": "K1",  
        "Value": "V1"  
      }  
    ]  
  }  
}
```

```
}
```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの「[考慮事項](#)」を参照してください。

例 2: イベントウィンドウの一連の時間範囲を変更するには

次の `modify-instance-event-window` 例では、イベントウィンドウの時間範囲を変更します。 `time-range` パラメータにより時間範囲を変更します。また、 `cron-expression` パラメータを指定することはできません。

```
aws ec2 modify-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --time-range '[{"StartWeekDay": "monday", "StartHour": 2, "EndWeekDay":  
    "wednesday", "EndHour": 8},  
    {"StartWeekDay": "thursday", "StartHour": 2, "EndWeekDay": "friday",  
    "EndHour": 8}]'
```

出力:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "TimeRanges": [  
      {  
        "StartWeekDay": "monday",  
        "StartHour": 2,  
        "EndWeekDay": "wednesday",  
        "EndHour": 8  
      },  
      {  
        "StartWeekDay": "thursday",  
        "StartHour": 2,  
        "EndWeekDay": "friday",  
        "EndHour": 8  
      }  
    ],  
    "Name": "myEventWindowName",  
    "AssociationTarget": {  
      "InstanceIds": [  
        "i-0abcdef1234567890",  

```

```

        "i-0be35f9acb8ba01f0"
    ],
    "Tags": [],
    "DedicatedHostIds": []
  },
  "State": "creating",
  "Tags": [
    {
      "Key": "K1",
      "Value": "V1"
    }
  ]
}
}

```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの[「考慮事項」](#)を参照してください。

例 3: イベントウィンドウの cron 式を変更するには

次のmodify-instance-event-window例では、イベントウィンドウの cron 式を変更します。cron-expression パラメータにより cron 式を変更します。また、time-range パラメータを指定することはできません。

```

aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --cron-expression "* 21-23 * * 2,3"

```

出力:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [
        "i-0abcdef1234567890",
        "i-0be35f9acb8ba01f0"
      ],
      "Tags": [],
    }
  }
}

```

```
        "DedicatedHostIds": []
    },
    "State": "creating",
    "Tags": [
        {
            "Key": "K1",
            "Value": "V1"
        }
    ]
}
}
```

イベントウィンドウの制約については、「Amazon EC2 ユーザーガイド」の「スケジュールされたイベント」セクションの「[考慮事項](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyInstanceEventWindow](#)」の「」を参照してください。

modify-instance-maintenance-options

次の例は、modify-instance-maintenance-options を使用する方法を説明しています。

AWS CLI

例 1: インスタンスの復旧動作を無効にするには

次のmodify-instance-maintenance-options例では、実行中または停止しているインスタンスの簡易自動復旧を無効にします。

```
aws ec2 modify-instance-maintenance-options \
  --instance-id i-0abcdef1234567890 \
  --auto-recovery disabled
```

出力:

```
{
  "InstanceId": "i-0abcdef1234567890",
  "AutoRecovery": "disabled"
}
```

詳細については、「Linux インスタンス用 Amazon ユーザーガイド」の「インスタンスの[復旧](#)」を参照してください。 EC2

例 2: インスタンスの復旧動作をデフォルトに設定するには

次のmodify-instance-maintenance-options例では、自動復旧動作をデフォルトに設定し、サポートされているインスタンスタイプの簡易自動復旧を有効にします。

```
aws ec2 modify-instance-maintenance-options \  
  --instance-id i-0abcdef1234567890 \  
  --auto-recovery default
```

出力:

```
{  
  "InstanceId": "i-0abcdef1234567890",  
  "AutoRecovery": "default"  
}
```

詳細については、「Linux インスタンス用 Amazon ユーザーガイド」の「インスタンスの[復旧](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyInstanceMaintenanceOptions](#)」の「」を参照してください。

modify-instance-metadata-options

次の例は、modify-instance-metadata-options を使用する方法を説明しています。

AWS CLI

例 1: を有効にするには IMDSv2

次のmodify-instance-metadata-options例では、指定されたインスタンスIMDSv2での使用を設定します。

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-tokens required \  
  --http-endpoint enabled
```

出力:

```
{
```



```
"InstanceId": "i-1234567898abcdef0",
"InstanceMetadataOptions": {
  "State": "pending",
  "HttpTokens": "required",
  "HttpPutResponseHopLimit": 1,
  "HttpEndpoint": "enabled"
}
}
```

詳細については、「Linux [インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド](#)」の「[インスタンスメタデータとユーザーデータ](#)」を参照してください。

例 2: インスタンスメタデータを無効にするには

次のmodify-instance-metadata-options例では、指定されたインスタンスですべてのバージョンのインスタンスメタデータの使用を無効にします。

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
  --http-endpoint disabled
```

出力:

```
{
  "InstanceId": "i-1234567898abcdef0",
  "InstanceMetadataOptions": {
    "State": "pending",
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "disabled"
  }
}
```

詳細については、「Linux [インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド](#)」の「[インスタンスメタデータとユーザーデータ](#)」を参照してください。

例 3: インスタンスのインスタンスメタデータIPv6エンドポイントを有効にするには

次のmodify-instance-metadata-options例は、インスタンスメタデータサービスのIPv6エンドポイントを有効にする方法を示しています。

```
aws ec2 modify-instance-metadata-options \
```

```
--instance-id i-1234567898abcdef0 \  
--http-protocol-ipv6 enabled \  
--http-endpoint enabled
```

出力:

```
{  
  "InstanceId": "i-1234567898abcdef0",  
  "InstanceMetadataOptions": {  
    "State": "pending",  
    "HttpTokens": "required",  
    "HttpPutResponseHopLimit": 1,  
    "HttpEndpoint": "enabled",  
    "HttpProtocolIpv6": "enabled"  
  }  
}
```

デフォルトでは、IPv6エンドポイントは無効になっています。これは、インスタンスを IPv6 のみのサブネットで起動した場合でも当てはまります。の IPv6 エンドポイント IMDS は、Nitro System 上に構築されたインスタンスでのみアクセスできます。詳細については、「Linux [インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド](#)」の「[インスタンスメタデータとユーザーデータ](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス `ModifyInstanceMetadataOptions`](#)」の「」を参照してください。

modify-instance-placement

次のコード例は、`modify-instance-placement` を使用する方法を示しています。

AWS CLI

例 1: Dedicated Host とのインスタンスのアフィニティを削除するには

次の `modify-instance-placement` 例では、Dedicated Host とのインスタンスのアフィニティを削除し、そのインスタンスタイプをサポートするアカウントで使用可能な Dedicated Host でインスタンスを起動できるようにします。

```
aws ec2 modify-instance-placement \  
  --instance-id i-0e6ddf6187EXAMPLE \  
  --affinity default
```

出力:

```
{
  "Return": true
}
```

例 2: インスタンスと指定された Dedicated Host 間のアフィニティを確立するには

次のmodify-instance-placement例では、インスタンスと Dedicated Host 間の起動関係を確立します。インスタンスは、指定された Dedicated Host でのみ実行できます。

```
aws ec2 modify-instance-placement \
  --instance-id i-0e6ddf6187EXAMPLE \
  --affinity host \
  --host-id i-0e6ddf6187EXAMPLE
```

出力:

```
{
  "Return": true
}
```

詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスのテナンシーとアフィニティの変更」](#)を参照してください。

例 3: インスタンスをプレースメントグループに移動するには

次のmodify-instance-placement例では、インスタンスをプレースメントグループに移動し、インスタンスを停止し、インスタンスのプレースメントを変更してからインスタンスを再起動します。

```
aws ec2 stop-instances \
  --instance-ids i-0123a456700123456

aws ec2 modify-instance-placement \
  --instance-id i-0123a456700123456 \
  --group-name MySpreadGroup

aws ec2 start-instances \
  --instance-ids i-0123a456700123456
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の [「インスタンスのプレースメントグループの変更」](#) を参照してください。

例 4: プレースメントグループからインスタンスを削除するには

次の `modify-instance-placement` 例では、インスタンスを停止し、インスタンス配置を変更してからインスタンスを再起動することで、インスタンスをプレースメントグループから削除します。次の例では、プレースメントグループ名に空の文字列 (「」) を指定して、インスタンスがプレースメントグループ内に配置されないことを示します。

インスタンスを停止します。

```
aws ec2 stop-instances \  
  --instance-ids i-0123a456700123456
```

配置を変更します (Windows コマンドプロンプト、Linux、macOS):

```
aws ec2 modify-instance-placement \  
  --instance-id i-0123a456700123456 \  
  --group-name ""
```

配置を変更する (Windows PowerShell):

```
aws ec2 modify-instance-placement `\  
  --instance-id i-0123a456700123456 `\  
  --group-name ""
```

インスタンスを再起動します。

```
aws ec2 start-instances \  
  --instance-ids i-0123a456700123456
```

出力:

```
{  
  "Return": true  
}
```

詳細については、[「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスのテナンシーとアフィニティの変更」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyInstancePlacement](#)」の「」を参照してください。

modify-ipam-pool

次の例は、modify-ipam-pool を使用する方法を説明しています。

AWS CLI

IPAM プールを変更するには

次のmodify-ipam-pool例では、IPAMプールを変更します。

(Linux):

```
aws ec2 modify-ipam-pool \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --add-allocation-resource-tags "Key=Owner,Value=Build Team" \  
  --clear-allocation-default-netmask-length \  
  --allocation-min-netmask-length 14
```

(Windows):

```
aws ec2 modify-ipam-pool ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --add-allocation-resource-tags "Key=Owner,Value=Build Team" ^  
  --clear-allocation-default-netmask-length ^  
  --allocation-min-netmask-length 14
```

出力:

```
{  
  "IpamPool": {  
    "OwnerId": "123456789012",  
    "IpamPoolId": "ipam-pool-0533048da7d823723",  
    "IpamPoolArn": "arn:aws:ec2::123456789012:ipam-pool/ipam-  
pool-0533048da7d823723",  
    "IpamScopeArn": "arn:aws:ec2::123456789012:ipam-scope/ipam-  
scope-02fc38cd4c48e7d38",  
    "IpamScopeType": "private",  
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",  
    "IpamRegion": "us-east-1",
```

```
"Locale": "None",
"PoolDepth": 1,
"State": "modify-complete",
"AutoImport": true,
"AddressFamily": "ipv4",
"AllocationMinNetmaskLength": 14,
"AllocationMaxNetmaskLength": 26,
"AllocationResourceTags": [
  {
    "Key": "Environment",
    "Value": "Preprod"
  },
  {
    "Key": "Owner",
    "Value": "Build Team"
  }
]
}
```

詳細については、「Amazon ユーザーガイド」の「[プールの編集](#)」を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyIpamPool](#)」の「」を参照してください。

modify-ipam-resource-cidr

次のコード例は、modify-ipam-resource-cidr を使用する方法を示しています。

AWS CLI

リソースCIDRに割り当てられた を変更するには

次のmodify-ipam-resource-cidr例では、リソース を変更しますCIDR。

(Linux):

```
aws ec2 modify-ipam-resource-cidr \
  --current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 \
  --destination-ipam-scope-id ipam-scope-0da34c61fd189a141 \
  --resource-id vpc-010e1791024eb0af9 \
  --resource-cidr 10.0.1.0/24 \
```

```
--resource-region us-east-1 \  
--monitored
```

(Windows):

```
aws ec2 modify-ipam-resource-cidr ^  
--current-ipam-scope-id ipam-scope-02fc38cd4c48e7d38 ^  
--destination-ipam-scope-id ipam-scope-0da34c61fd189a141 ^  
--resource-id vpc-010e1791024eb0af9 ^  
--resource-cidr 10.0.1.0/24 ^  
--resource-region us-east-1 ^  
--monitored
```

出力:

```
{  
  "IpamResourceCidr": {  
    "IpamId": "ipam-08440e7a3acde3908",  
    "IpamScopeId": "ipam-scope-0da34c61fd189a141",  
    "IpamPoolId": "ipam-pool-0533048da7d823723",  
    "ResourceRegion": "us-east-1",  
    "ResourceOwnerId": "123456789012",  
    "ResourceId": "vpc-010e1791024eb0af9",  
    "ResourceCidr": "10.0.1.0/24",  
    "ResourceType": "vpc",  
    "ResourceTags": [  
      {  
        "Key": "Environment",  
        "Value": "Preprod"  
      },  
      {  
        "Key": "Owner",  
        "Value": "Build Team"  
      }  
    ],  
    "IpUsage": 0.0,  
    "ComplianceStatus": "noncompliant",  
    "ManagementState": "managed",  
    "OverlapStatus": "overlapping",  
    "VpcId": "vpc-010e1791024eb0af9"  
  }  
}
```

リソースの移動の詳細については、「Amazon VPCIPAMユーザーガイド」の「[スコープCIDRs間でリソースを移動する](#)」を参照してください。

モニタリング状態の変更の詳細については、「Amazon ユーザーガイド」の「[リソースのモニタリング状態の変更CIDRs](#)」を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyIpamResourceCidr](#)」の「」を参照してください。

modify-ipam-resource-discovery

次の例は、modify-ipam-resource-discovery を使用する方法を説明しています。

AWS CLI

リソース検出のオペレーションリージョンを変更するには

この例では、リソース検出のオペレーションリージョンを変更するIPAM委任された管理者です。

このリクエストを完了するには：

デフォルトのリソース検出を変更することはできません。また、リソース検出の所有者である必要があります。で取得できるリソース検出 ID が必要です[describe-ipam-resource-discoveries](#)。

次のmodify-ipam-resource-discovery例では、AWS アカウントのデフォルト以外のリソース検出を変更します。

```
aws ec2 modify-ipam-resource-discovery \  
  --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \  
  --add-operating-regions RegionName='us-west-1' \  
  --remove-operating-regions RegionName='us-east-2' \  
  --region us-east-1
```

出力:

```
{  
  "IpamResourceDiscovery": {  
    "OwnerId": "149977607591",  
    "IpamResourceDiscoveryId": "ipam-res-disco-0365d2977fc1672fe",  
    "IpamResourceDiscoveryArn": "arn:aws:ec2::149977607591:ipam-resource-  
discovery/ipam-res-disco-0365d2977fc1672fe",  
    "IpamResourceDiscoveryRegion": "us-east-1",  
    "Description": "Example",
```



```
    "OperatingRegions": [  
      {  
        "RegionName": "us-east-1"  
      },  
      {  
        "RegionName": "us-west-1"  
      }  
    ],  
    "IsDefault": false,  
    "State": "modify-in-progress"  
  }  
}
```

詳細については、「[Amazon VPCIPAMユーザーガイド](#)」の「[リソース検出の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスModifyIpamResourceDiscovery](#)」の「」を参照してください。

modify-ipam-scope

次の例は、modify-ipam-scope を使用する方法を説明しています。

AWS CLI

スコープの説明を変更するには

このシナリオでは、IPAMスコープの説明を変更するIPAM委任された管理者です。

このリクエストを完了するには、で取得できるスコープ ID が必要です [describe-ipam-scopes](#)。

次のmodify-ipam-scope例では、スコープの説明を更新します。

```
aws ec2 modify-ipam-scope \  
  --ipam-scope-id ipam-scope-0d3539a30b57dcdd1 \  
  --description example \  
  --region us-east-1
```

出力:

```
{  
  "IpamScope": {  
    "OwnerId": "320805250157",
```

```
    "IpamScopeId": "ipam-scope-0d3539a30b57dcdd1",
    "IpamScopeArn": "arn:aws:ec2::320805250157:ipam-scope/ipam-
scope-0d3539a30b57dcdd1",
    "IpamArn": "arn:aws:ec2::320805250157:ipam/ipam-005f921c17ebd5107",
    "IpamRegion": "us-east-1",
    "IpamScopeType": "public",
    "IsDefault": true,
    "Description": "example",
    "PoolCount": 1,
    "State": "modify-in-progress"
  }
}
```

スコープの詳細については、「Amazon VPC IPAM ユーザーガイド」の「[IPAMの仕組み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyIpamScope](#)」の「」を参照してください。

modify-ipam

次の例は、modify-ipam を使用する方法を説明しています。

AWS CLI

を変更するには IPAM

次のmodify-ipam例では、オペレーションリージョンIPAMを追加して を変更します。

(Linux):

```
aws ec2 modify-ipam \  
  --ipam-id ipam-08440e7a3acde3908 \  
  --add-operating-regions RegionName=us-west-2
```

(Windows):

```
aws ec2 modify-ipam ^  
  --ipam-id ipam-08440e7a3acde3908 ^  
  --add-operating-regions RegionName=us-west-2
```

出力:

```
{
  "Ipam": {
    "OwnerId": "123456789012",
    "IpamId": "ipam-08440e7a3acde3908",
    "IpamArn": "arn:aws:ec2::123456789012:ipam/ipam-08440e7a3acde3908",
    "IpamRegion": "us-east-1",
    "PublicDefaultScopeId": "ipam-scope-0b9eed026396dbc16",
    "PrivateDefaultScopeId": "ipam-scope-02fc38cd4c48e7d38",
    "ScopeCount": 3,
    "OperatingRegions": [
      {
        "RegionName": "us-east-1"
      },
      {
        "RegionName": "us-east-2"
      },
      {
        "RegionName": "us-west-1"
      },
      {
        "RegionName": "us-west-2"
      }
    ],
    "State": "modify-in-progress"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyIpam](#)」の「」を参照してください。

modify-launch-template

次の例は、modify-launch-template を使用する方法を説明しています。

AWS CLI

デフォルトの起動テンプレートのバージョンを変更するには

この例では、指定された起動テンプレートのバージョン 2 をデフォルトバージョンとして指定します。

コマンド:

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-version 2
```

出力:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "WebServers",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyLaunchTemplate](#)」の「」を参照してください。

modify-managed-prefix-list

次の例は、modify-managed-prefix-list を使用する方法を説明しています。

AWS CLI

プレフィックスリストを変更するには

次のmodify-managed-prefix-list例では、指定されたプレフィックスリストにエントリを追加します。

```
aws ec2 modify-managed-prefix-list \
  --prefix-list-id pl-0123456abcabcabc1 \
  --add-entries Cidr=10.1.0.0/16,Description=vpc-c \
  --current-version 1
```

出力:

```
{
  "PrefixList": {
    "PrefixListId": "pl-0123456abcabcabc1",
    "AddressFamily": "IPv4",
  }
}
```

```
    "State": "modify-in-progress",
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
    "PrefixListName": "vpc-cidrs",
    "MaxEntries": 10,
    "Version": 1,
    "OwnerId": "123456789012"
  }
}
```

詳細については、「Amazon ユーザーガイド」の「[マネージドプレフィックスリスト](#)」を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyManagedPrefixList](#)」の「」を参照してください。

modify-network-interface-attribute

次の例は、modify-network-interface-attribute を使用する方法を説明しています。

AWS CLI

ネットワークインターフェイスのアタッチメント属性を変更するには

この例では、指定されたネットワークインターフェイスの attachment 属性を変更します。

コマンド:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```

ネットワークインターフェイスの説明属性を変更するには

この例では、指定されたネットワークインターフェイスの description 属性を変更します。

コマンド:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
description "My description"
```

ネットワークインターフェイスの groupSet 属性を変更するには

この例では、指定されたネットワークインターフェイスの groupSet 属性を変更します。

コマンド:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --groups sg-903004f8 sg-1a2b3c4d
```

ネットワークインターフェイスの sourceDestCheck 属性を変更するには

この例では、指定されたネットワークインターフェイスの sourceDestCheck 属性を変更します。

コマンド:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-source-dest-check
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyNetworkInterfaceAttribute](#)」の「」を参照してください。

modify-private-dns-name-options

次の例は、modify-private-dns-name-options を使用する方法を説明しています。

AWS CLI

インスタンスホスト名のオプションを変更するには

次の modify-private-dns-name-options 例では、DNSA レコードを持つインスタンスホスト名の DNS クエリに回答するオプションを無効にします。

```
aws ec2 modify-private-dns-name-options \  
  --instance-id i-1234567890abcdef0 \  
  --no-enable-resource-name-dns-a-record
```

出力:

```
{  
  "Return": true  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EC2 インスタンスのホスト名タイプ」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyPrivateDnsNameOptions](#)」の「」を参照してください。

modify-reserved-instances

次の例は、modify-reserved-instances を使用する方法を説明しています。

AWS CLI

リザーブドインスタンスを変更するには

この例では、リザーブドインスタンスを同じリージョンの別のアベイラビリティゾーンに移動します。

コマンド:

```
aws ec2 modify-reserved-instances --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classical,InstanceCount=10
```

出力:

```
{
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"
}
```

リザーブドインスタンスのネットワークプラットフォームを変更するには

この例では、EC2-Classical リザーブドインスタンスを EC2- に変換します VPC。

コマンド:

```
aws ec2 modify-reserved-instances --reserved-instances-ids f127bd27-edb7-44c9-a0eb-0d7e09259af0 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-VPC,InstanceCount=5
```

出力:

```
{
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"
}
```

詳細については、「Amazon EC2ユーザーガイド」の「リザーブドインスタンスの変更」を参照してください。

リザーブドインスタンスのインスタンスサイズを変更するには

この例では、us-west-1c に 10 m1.small Linux/UNIX インスタンスを持つリザーブドインスタンスを変更して、8 m1.small インスタンスを 2 m1.large インスタンスにし、残りの 2 m1.small を同じアベイラビリティゾーンに 1 m1.medium インスタンスにします。コマンド:

```
aws ec2 modify-reserved-instances --reserved-instances-ids 1ba8e2e3-3556-4264-949e-63ee671405a9 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=2,InstanceType=m1.large AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=1,InstanceType=m1.medium
```

出力:

```
{
  "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-b3e3-1c6b11fa00b6"
}
```

詳細については、「Amazon EC2ユーザーガイド」の「予約のインスタンスサイズの変更」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyReservedInstances](#)」の「」を参照してください。

modify-security-group-rules

次の例は、modify-security-group-rules を使用する方法を説明しています。

AWS CLI

セキュリティグループルールを変更して、ルールの説明、IP プロトコル、CidrIpv4 アドレス範囲を更新するには

次のmodify-security-group-rules例では、説明、IP プロトコル、および指定されたセキュリティグループルールIPV4CIDRの範囲を更新します。security-group-rules パラメータを使用して、指定されたセキュリティグループルールの更新を入力します。はすべてのプロトコル-1を指定します。


```
aws ec2 modify-security-group-rules \  
  --group-id sg-1234567890abcdef0 \  
  --security-group-rules SecurityGroupId=sgr-  
abcdef01234567890,SecurityGroupRule='{Description=test,IpProtocol=-1,CidrIpv4=0.0.0.0/0}'
```

出力:

```
{  
  "Return": true  
}
```

セキュリティグループルールの詳細については、「Amazon ユーザーガイド」の「[セキュリティグループルール](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifySecurityGroupRules](#)」の「」を参照してください。

modify-snapshot-attribute

次の例は、modify-snapshot-attribute を使用する方法を説明しています。

AWS CLI

例 1: スナップショット属性を変更するには

次のmodify-snapshot-attribute例では、指定したスナップショットの createVolumePermission 属性を更新し、指定したユーザーのボリュームアクセス許可を削除します。

```
aws ec2 modify-snapshot-attribute \  
  --snapshot-id snap-1234567890abcdef0 \  
  --attribute createVolumePermission \  
  --operation-type remove \  
  --user-ids 123456789012
```

例 2: スナップショットを公開するには

次のmodify-snapshot-attribute例では、指定されたスナップショットをパブリックにします。

```
aws ec2 modify-snapshot-attribute \  
  --snapshot-id snap-1234567890abcdef0 \  
  --attribute public \  
  --operation-type add \  
  --user-ids 123456789012
```

```
--snapshot-id snap-1234567890abcdef0 \  
--attribute createVolumePermission \  
--operation-type add \  
--group-names all
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifySnapshotAttribute](#)」の「」を参照してください。

modify-snapshot-tier

次のコード例は、modify-snapshot-tier を使用する方法を示しています。

AWS CLI

例 1: スナップショットをアーカイブするには

次のmodify-snapshot-tier例では、指定されたスナップショットをアーカイブします。

```
aws ec2 modify-snapshot-tier \  
--snapshot-id snap-01234567890abcdef \  
--storage-tier archive
```

出力:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "TieringStartTime": "2021-09-15T16:44:37.574Z"  
}
```

TieringStartTime レスポンスパラメータは、アーカイブプロセスが開始された日時をUTC時間形式 (YYYY-MM-DDTHH:MM:) で示しますSSZ。

スナップショットのアーカイブの詳細については、[「Amazon ユーザーガイド」の「Amazon EBSスナップショットのアーカイブ」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifySnapshotTier](#)」の「」を参照してください。

modify-spot-fleet-request

次のコード例は、modify-spot-fleet-request を使用する方法を示しています。

AWS CLI

スポットフリートリクエストを変更するには

この例では、指定されたスポットフリートリクエストのターゲットキャパシティを更新します。

コマンド:

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

出力:

```
{
  "Return": true
}
```

このコマンド例では、結果としてスポットインスタンスを終了することなく、指定されたスポットフリートリクエストのターゲット容量を減少させます。

コマンド:

```
aws ec2 modify-spot-fleet-request --target-capacity 10 --excess-capacity-termination-policy NoTermination --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifySpotFleetRequest](#)」の「」を参照してください。

modify-subnet-attribute

次のコード例は、modify-subnet-attribute を使用する方法を示しています。

AWS CLI

サブネットのパブリックIPv4アドレス指定の動作を変更するには

この例では、サブネット-1a2b3c4d を変更して、このサブネットで起動されたすべてのインスタンスにパブリックIPv4アドレスが割り当てられるように指定します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-launch
```

サブネットのIPv6アドレス指定動作を変更するには

この例では、サブネット-1a2b3c4d を変更して、このサブネットで起動されたすべてのインスタンスにサブネットの範囲からのIPv6アドレスが割り当てられていることを指定します。

コマンド:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

詳細については、AWS 「Virtual Private Cloud ユーザーガイドVPC」の「の IP アドレス指定」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifySubnetAttribute](#)」の「」を参照してください。

modify-traffic-mirror-filter-network-services

次の例は、modify-traffic-mirror-filter-network-services を使用する方法を説明しています。

AWS CLI

Traffic Mirror フィルターにネットワークサービスを追加するには

次のmodify-traffic-mirror-filter-network-services例では、Amazon DNSネットワークサービスを指定されたフィルターに追加します。

```
aws ec2 modify-traffic-mirror-filter-network-services \  
  --traffic-mirror-filter-id tmf-04812ff784EXAMPLE \  
  --add-network-service amazon-dns
```

出力:

```
{
  "TrafficMirrorFilter": {
    "Tags": [
      {
        "Key": "Name",
        "Value": "Production"
      }
    ],
    "EgressFilterRules": [],
    "NetworkServices": [
      "amazon-dns"
    ],
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
    "IngressFilterRules": [
      {
        "SourceCidrBlock": "0.0.0.0/0",
        "RuleNumber": 1,
        "DestinationCidrBlock": "0.0.0.0/0",
        "Description": "TCP Rule",
        "Protocol": 6,
        "TrafficDirection": "ingress",
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "RuleAction": "accept",
        "TrafficMirrorFilterRuleId": "tmf-04812ff784EXAMPLE"
      }
    ]
  }
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーフィルターネットワークサービスの変更AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyTrafficMirrorFilterNetworkServices](#)」の「」を参照してください。

modify-traffic-mirror-filter-rule

次のコード例は、modify-traffic-mirror-filter-rule を使用する方法を示しています。

AWS CLI

トラフィックミラーフィルタールールを変更するには

次のmodify-traffic-mirror-filter-rule例では、指定されたトラフィックミラーフィルタールールの説明を変更します。

```
aws ec2 modify-traffic-mirror-filter-rule \  
  --traffic-mirror-filter-rule-id tmfr-0ca76e0e08EXAMPLE \  
  --description "TCP Rule"
```

出力:

```
{  
  "TrafficMirrorFilterRule": {  
    "TrafficMirrorFilterRuleId": "tmfr-0ca76e0e08EXAMPLE",  
    "TrafficMirrorFilterId": "tmf-0293f26e86EXAMPLE",  
    "TrafficDirection": "ingress",  
    "RuleNumber": 100,  
    "RuleAction": "accept",  
    "Protocol": 6,  
    "DestinationCidrBlock": "10.0.0.0/24",  
    "SourceCidrBlock": "10.0.0.0/24",  
    "Description": "TCP Rule"  
  }  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーフィルタールールの変更AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyTrafficMirrorFilterRule](#)」の「」を参照してください。

modify-traffic-mirror-session

次の例は、modify-traffic-mirror-session を使用する方法を説明しています。

AWS CLI

トラフィックミラーセッションを変更するには

次のmodify-traffic-mirror-session例では、トラフィックミラーセッションの説明とミラーリングするパケットの数を変更します。

```
aws ec2 modify-traffic-mirror-session \  
  --description "Change packet length" \  
  --packet-length 100
```

```
--traffic-mirror-session-id tms-08a33b1214EXAMPLE \  
--remove-fields "packet-length"
```

出力:

```
{  
  "TrafficMirrorSession": {  
    "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",  
    "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",  
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",  
    "NetworkInterfaceId": "eni-070203f901EXAMPLE",  
    "OwnerId": "111122223333",  
    "SessionNumber": 1,  
    "VirtualNetworkId": 7159709,  
    "Description": "Change packet length",  
    "Tags": []  
  }  
}
```

詳細については、[「トラフィックミラーリングガイド」の「トラフィックミラーリングセッションの変更」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyTrafficMirrorSession](#)」の「」を参照してください。

modify-transit-gateway-prefix-list-reference

次のコード例は、`modify-transit-gateway-prefix-list-reference` を使用方法を示しています。

AWS CLI

プレフィックスリストへの参照を変更するには

次の`modify-transit-gateway-prefix-list-reference`例では、トラフィックがルーティングされるアタッチメントを変更することで、指定されたルートテーブルのプレフィックスリストリファレンスを変更します。

```
aws ec2 modify-transit-gateway-prefix-list-reference \  
--transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \  
--prefix-list-id pl-11111122222222333 \  
--transit-gateway-attachment-id tgw-attach-aabbccddaabbccaab
```

出力:

```
{
  "TransitGatewayPrefixListReference": {
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-11111122222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "modifying",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}
```

詳細については、「Transit Gateways ガイド」の「[プレフィックスリストリファレンス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [Modify Transit Gateway Prefix List Reference](#)」の「」を参照してください。

modify-transit-gateway-vpc-attachment

次のコード例は、modify-transit-gateway-vpc-attachment を使用する方法を示しています。

AWS CLI

Transit Gateway アタッチメントを変更するには VPC

次のmodify-transit-gateway-vpc-attachment例では、指定された Transit Gateway VPC アタッチメントにサブネットを追加します。

```
aws ec2 modify-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE \
  --add-subnet-ids subnet-0e51f45802EXAMPLE
```

出力:

```
{
```



```
"TransitGatewayVpcAttachment": {
  "TransitGatewayAttachmentId": "tgw-attach-09fbd47ddfEXAMPLE",
  "TransitGatewayId": "tgw-0560315ccfEXAMPLE",
  "VpcId": "vpc-5eccc927",
  "VpcOwnerId": "111122223333",
  "State": "modifying",
  "SubnetIds": [
    "subnet-0e51f45802EXAMPLE",
    "subnet-1EXAMPLE"
  ],
  "CreationTime": "2019-08-08T16:47:38.000Z",
  "Options": {
    "DnsSupport": "enable",
    "Ipv6Support": "disable"
  }
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway の へのアタッチメント VPC」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyTransitGatewayVpcAttachment](#)」の「」を参照してください。

modify-transit-gateway

次の例は、modify-transit-gateway を使用する方法を説明しています。

AWS CLI

トランジットゲートウェイを変更するには

次のmodify-transit-gateway例では、VPNアタッチメントECMPのサポートを有効にして、指定されたトランジットゲートウェイを変更します。

```
aws ec2 modify-transit-gateway \
  --transit-gateway-id tgw-111111222222aaaaa \
  --options VpnEcmpSupport=enable
```

出力:

```
{
```

```
"TransitGateway": {
  "TransitGatewayId": "tgw-111111222222aaaaa",
  "TransitGatewayArn": "64512",
  "State": "modifying",
  "OwnerId": "123456789012",
  "CreationTime": "2020-04-30T08:41:37.000Z",
  "Options": {
    "AmazonSideAsn": 64512,
    "AutoAcceptSharedAttachments": "disable",
    "DefaultRouteTableAssociation": "enable",
    "AssociationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
    "DefaultRouteTablePropagation": "enable",
    "PropagationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
    "VpnEcmpSupport": "enable",
    "DnsSupport": "enable"
  }
}
```

詳細については、[「Transit Gateways ガイド」](#)の「Transit Gateways」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyTransitGateway](#)」の「」を参照してください。

modify-verified-access-endpoint-policy

次の例は、modify-verified-access-endpoint-policy を使用する方法を説明しています。

AWS CLI

エンドポイントの検証済みアクセスポリシーを設定するには

次のmodify-verified-access-endpoint-policy例では、指定された Verified Access エンドポイントに指定された Verified Access ポリシーを追加します。

```
aws ec2 modify-verified-access-endpoint-policy \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --policy-enabled \
  --policy-document file://policy.txt
```

policy.txt の内容:

```
permit(principal,action,resource)
```

```
when {
  context.identity.groups.contains("finance") &&
  context.identity.email.verified == true
};
```

出力:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access ポリシー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVerifiedAccessEndpointPolicy](#)」の「」を参照してください。

modify-verified-access-endpoint

次の例は、modify-verified-access-endpoint を使用する方法を説明しています。

AWS CLI

Verified Access エンドポイントの設定を変更するには

次のmodify-verified-access-endpoint例では、指定された Verified Access エンドポイントに指定された説明を追加します。

```
aws ec2 modify-verified-access-endpoint \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --description "Testing Verified Access"
```

出力:

```
{
  "VerifiedAccessEndpoint": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
    "VerifiedAccessEndpointId": "vae-066fac616d4d546f2",
```

```
    "ApplicationDomain": "example.com",
    "EndpointType": "network-interface",
    "AttachmentType": "vpc",
    "DomainCertificateArn": "arn:aws:acm:us-east-2:123456789012:certificate/
eb065ea0-26f9-4e75-a6ce-0a1a7EXAMPLE",
    "EndpointDomain": "my-ava-
app.edge-00c3372d53b1540bb.vai-0ce000c0b7643abea.prod.verified-access.us-
east-2.amazonaws.com",
    "SecurityGroupIds": [
      "sg-004915970c4c8f13a"
    ],
    "NetworkInterfaceOptions": {
      "NetworkInterfaceId": "eni-0aec70418c8d87a0f",
      "Protocol": "https",
      "Port": 443
    },
    "Status": {
      "Code": "updating"
    },
    "Description": "Testing Verified Access",
    "CreationTime": "2023-08-25T20:54:43",
    "LastUpdatedTime": "2023-08-25T22:46:32"
  }
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access エンドポイント」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVerifiedAccessEndpoint](#)」の「」を参照してください。

modify-verified-access-group-policy

次のコード例は、modify-verified-access-group-policy を使用する方法を示しています。

AWS CLI

グループの Verified Access ポリシーを設定するには

次のmodify-verified-access-group-policy例では、指定された Verified Access ポリシーを指定された Verified Access グループに追加します。

```
aws ec2 modify-verified-access-group-policy \
```

```
--verified-access-group-id vagr-0dbe967baf14b7235 \  
--policy-enabled \  
--policy-document file://policy.txt
```

policy.txt の内容:

```
permit(principal,action,resource)  
when {  
    context.identity.groups.contains("finance") &&  
    context.identity.email.verified == true  
};
```

出力:

```
{  
  "PolicyEnabled": true,  
  "PolicyDocument": "permit(principal,action,resource)\nwhen  
{\n  context.identity.groups.contains(\"finance\") &&\n  context.identity.email_verified == true\n};"  
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access グループ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVerifiedAccessGroupPolicy](#)」の「」を参照してください。

modify-verified-access-group

次の例は、modify-verified-access-group を使用する方法を説明しています。

AWS CLI

Verified Access グループの設定を変更するには

次のmodify-verified-access-group例では、指定された Verified Access グループに指定された説明を追加します。

```
aws ec2 modify-verified-access-group \  
--verified-access-group-id vagr-0dbe967baf14b7235 \  
--description "Testing Verified Access"
```

出力:

```
{
  "VerifiedAccessGroup": {
    "VerifiedAccessGroupId": "vagr-0dbe967baf14b7235",
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "Testing Verified Access",
    "Owner": "123456789012",
    "VerifiedAccessGroupArn": "arn:aws:ec2:us-east-2:123456789012:verified-
access-group/vagr-0dbe967baf14b7235",
    "CreationTime": "2023-08-25T19:55:19",
    "LastUpdatedTime": "2023-08-25T22:17:25"
  }
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access グループ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVerifiedAccessGroup](#)」の「」を参照してください。

modify-verified-access-instance-logging-configuration

次のコード例は、`modify-verified-access-instance-logging-configuration` を使用する方法を示しています。

AWS CLI

Verified Access インスタンスのログ記録を有効にするには

次の`modify-verified-access-instance-logging-configuration`例では、指定された Verified Access インスタンスのアクセスログ記録を有効にします。ログは、指定された CloudWatch Logs ロググループに配信されます。

```
aws ec2 modify-verified-access-instance-logging-configuration \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --access-logs CloudWatchLogs={Enabled=true,LogGroup=my-log-group}
```

出力:

```
{
  "LoggingConfiguration": {
```

```
"VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
"AccessLogs": {
  "S3": {
    "Enabled": false
  },
  "CloudWatchLogs": {
    "Enabled": true,
    "DeliveryStatus": {
      "Code": "success"
    },
    "LogGroup": "my-log-group"
  },
  "KinesisDataFirehose": {
    "Enabled": false
  },
  "LogVersion": "ocsf-1.0.0-rc.2",
  "IncludeTrustContext": false
}
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access ログ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVerifiedAccessInstanceLoggingConfiguration](#)」の「」を参照してください。

modify-verified-access-instance

次のコード例は、modify-verified-access-instance を使用する方法を示しています。

AWS CLI

Verified Access インスタンスの設定を変更するには

次のmodify-verified-access-instance例では、指定された Verified Access インスタンスに指定された説明を追加します。

```
aws ec2 modify-verified-access-instance \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --description "Testing Verified Access"
```

出力:

```
{
  "VerifiedAccessInstance": {
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
    "Description": "Testing Verified Access",
    "VerifiedAccessTrustProviders": [
      {
        "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
        "TrustProviderType": "user",
        "UserTrustProviderType": "iam-identity-center"
      }
    ],
    "CreationTime": "2023-08-25T18:27:56",
    "LastUpdatedTime": "2023-08-25T22:41:04"
  }
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access インスタンス」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVerifiedAccessInstance](#)」の「」を参照してください。

modify-verified-access-trust-provider

次のコード例は、`modify-verified-access-trust-provider` を使用する方法を示しています。

AWS CLI

Verified Access 信頼プロバイダーの設定を変更するには

次の`modify-verified-access-trust-provider`例では、指定された Verified Access 信頼プロバイダーに指定された説明を追加します。

```
aws ec2 modify-verified-access-trust-provider \
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7 \
  --description "Testing Verified Access"
```

出力:

```
{
```



```
"VerifiedAccessTrustProvider": {
  "VerifiedAccessTrustProviderId": "vatp-0bb32de759a3e19e7",
  "Description": "Testing Verified Access",
  "TrustProviderType": "user",
  "UserTrustProviderType": "iam-identity-center",
  "PolicyReferenceName": "idc",
  "CreationTime": "2023-08-25T19:00:38",
  "LastUpdatedTime": "2023-08-25T19:18:21"
}
```

詳細については、[「Verified Access ユーザーガイド」の「Verified Access の信頼プロバイダー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVerifiedAccessTrustProvider](#)」の「」を参照してください。

modify-volume-attribute

次の例は、modify-volume-attribute を使用する方法を説明しています。

AWS CLI

ボリューム属性を変更するには

この例では、ID を持つボリュームの autoEnableIo 属性を vol-1234567890abcdef0 に設定しますtrue。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVolumeAttribute](#)」の「」を参照してください。

modify-volume

次のコード例は、modify-volume を使用する方法を示しています。

AWS CLI

例 1: ボリュームのサイズを変更してボリュームを変更するには

次のmodify-volume例では、指定したボリュームのサイズを 150GBに変更します。

コマンド:

```
aws ec2 modify-volume --size 150 --volume-id vol-1234567890abcdef0
```

出力:

```
{
  "VolumeModification": {
    "TargetSize": 150,
    "TargetVolumeType": "io1",
    "ModificationState": "modifying",
    "VolumeId": " vol-1234567890abcdef0",
    "TargetIops": 100,
    "StartTime": "2019-05-17T11:27:19.000Z",
    "Progress": 0,
    "OriginalVolumeType": "io1",
    "OriginalIops": 100,
    "OriginalSize": 100
  }
}
```

例 2: タイプ、サイズ、IOPS値を変更してボリュームを変更するには

次のmodify-volume例では、ボリュームタイプをプロビジョンドに変更しIOPSSSD、ターゲットIOPSレートを 10,000 に設定し、ボリュームサイズを 350GBに設定します。

```
aws ec2 modify-volume \
  --volume-type io1 \
  --iops 10000 \
  --size 350 \
  --volume-id vol-1234567890abcdef0
```

出力:

```
{
  "VolumeModification": {
    "TargetSize": 350,
    "TargetVolumeType": "io1",
    "ModificationState": "modifying",
    "VolumeId": "vol-0721c1a9d08c93bf6",
  }
}
```

```
    "TargetIops": 10000,  
    "StartTime": "2019-05-17T11:38:57.000Z",  
    "Progress": 0,  
    "OriginalVolumeType": "gp2",  
    "OriginalIops": 150,  
    "OriginalSize": 50  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVolume](#)」の「」を参照してください。

modify-vpc-attribute

次の例は、`modify-vpc-attribute` を使用する方法を説明しています。

AWS CLI

`enableDnsSupport` 属性を変更するには

この例では、`enableDnsSupport` 属性を変更します。この属性は、でDNS解決が有効になっているかどうかを示しますVPC。この属性が の場合true、Amazon DNSサーバーはインスタンスのDNSホスト名を対応する IP アドレスに解決します。それ以外の場合は解決しません。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value  
\":false}"
```

`enableDnsHostnames` 属性を変更するには

この例では、`enableDnsHostnames` 属性を変更します。この属性は、VPC get DNS hostnames で起動されたインスタスかどうかを示します。この属性が の場合true、のインスタンスはDNSホスト名VPCを取得します。それ以外の場合は取得しません。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value  
\":false}"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVpcAttribute](#)」の「」を参照してください。

modify-vpc-endpoint-connection-notification

次のコード例は、`modify-vpc-endpoint-connection-notification` を使用する方法を示しています。

AWS CLI

エンドポイント接続通知を変更するには

この例では、指定されたエンドポイント接続通知の SNS トピックを変更します。

コマンド:

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-id vpce-nfn-008776de7e03f5abc --connection-events Accept Reject --connection-notification-arn arn:aws:sns:us-east-2:123456789012:mytopic
```

出力:

```
{
  "ReturnValue": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVpcEndpointConnectionNotification](#)」の「」を参照してください。

modify-vpc-endpoint-service-configuration

次の例は、`modify-vpc-endpoint-service-configuration` を使用する方法を説明しています。

AWS CLI

エンドポイントサービスの設定を変更するには

この例では、指定されたエンドポイントサービスの受け入れ要件を変更します。

コマンド:

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-svc-09222513e6e77dc86 --no-acceptance-required
```

出力:

```
{  
  "ReturnValue": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVpcEndpointServiceConfiguration](#)」の「」を参照してください。

modify-vpc-endpoint-service-payer-responsibility

次のコード例は、`modify-vpc-endpoint-service-payer-responsibility` を使用する方法を示しています。

AWS CLI

支払者の責任を変更するには

次の`modify-vpc-endpoint-service-payer-responsibility`例では、指定されたエンドポイントサービスの支払者の責任を変更します。

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \  
  --service-id vpce-svc-071afff70666e61e0 \  
  --payer-responsibility ServiceOwner
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVpcEndpointServicePayerResponsibility](#)」の「」を参照してください。

modify-vpc-endpoint-service-permissions

次の例は、`modify-vpc-endpoint-service-permissions` を使用する方法を説明しています。

AWS CLI

エンドポイントサービスのアクセス許可を変更するには

この例では、AWS アカウントが指定されたエンドポイントサービスに接続するためのアクセス許可を追加します。

コマンド:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

出力:

```
{
  "ReturnValue": true
}
```

この例では、特定のIAMユーザー (admin) が指定されたエンドポイントサービスに接続するためのアクセス許可を追加します。

コマンド:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:user/admin"]'
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVpcEndpointServicePermissions](#)」の「」を参照してください。

modify-vpc-endpoint

次のコード例は、modify-vpc-endpoint を使用する方法を示しています。

AWS CLI

ゲートウェイエンドポイントを変更するには

この例では、ルートテーブルをエンドポイント `rtb-aaa222bb` に関連付け、ポリシードキュメントをリセット `vpce-1a2b3c4d` することで、ゲートウェイエンドポイントを変更します。

コマンド:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-ids rtb-aaa222bb --reset-policy
```

出力:

```
{
  "Return": true
}
```

インターフェイスエンドポイントを変更するには

この例では、エンドポイントsubnet-d6fcaa8dにサブネットvpce-0fe5b17a0707d6fa5を追加してインターフェイスエンドポイントを変更します。

コマンド:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-id subnet-d6fcaa8d
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVpcEndpoint](#)」の「」を参照してください。

modify-vpc-peering-connection-options

次の例は、modify-vpc-peering-connection-options を使用する方法を説明しています。

AWS CLI

ローカル接続からVPCピアリング ClassicLink 接続を介した通信を有効にするには

この例では、ピアリング接続 の場合pcx-aaaabbbb、リクエスタの所有者はVPCピアリング接続オプションVPCを変更して、ローカル ClassicLink 接続がピア と通信できるようにしますVPC。

コマンド:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

出力:

```
{
  "RequesterPeeringConnectionOptions": {
    "AllowEgressFromLocalClassicLinkToRemoteVpc": true
  }
}
```

ローカルからリモート接続VPCへのVPCピアリング ClassicLink 接続を介した通信を有効にするには

この例では、アクセプターの所有者はVPCピアリング接続オプションVPCを変更して、ローカルがピア内の ClassicLink 接続とVPC通信できるようにしますVPC。

コマンド:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --accepter-peering-connection-options AllowEgressFromLocalVpcToRemoteClassicLink=true
```

出力:

```
{
  "AccepterPeeringConnectionOptions": {
    "AllowEgressFromLocalVpcToRemoteClassicLink": true
  }
}
```

VPCピアリング接続のDNS解決サポートを有効にするには

この例では、リクエストの所有者は のVPCピアリング接続オプションVPCを変更pcx-aaaabbbbして、ピアのインスタンスからクエリを実行するとVPC、ローカルがパブリックDNSホスト名をプライベートIPアドレスに解決できるようにしますVPC。

コマンド:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

出力:

```
{
```



```
"RequesterPeeringConnectionOptions": {  
  "AllowDnsResolutionFromRemoteVpc": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVpcPeeringConnectionOptions](#)」の「」を参照してください。

modify-vpc-tenancy

次の例は、modify-vpc-tenancy を使用する方法を説明しています。

AWS CLI

のテナンシーを変更するには VPC

この例では、のテナンシーVPCvpc-1a2b3c4dを に変更しますdefault。

コマンド:

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

出力:

```
{  
  "Return": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVpcTenancy](#)」の「」を参照してください。

modify-vpn-connection-options

次の例は、modify-vpn-connection-options を使用する方法を説明しています。

AWS CLI

VPN接続オプションを変更するには

次のmodify-vpn-connection-options例では、指定されたVPN接続のIPv4CIDRカスタマーゲートウェイ側のローカルを変更します。

```
aws ec2 modify-vpn-connection-options \  
--vpn-connection-id vpn-1122334455aabbccd \  
--local-ipv4-network-cidr 10.0.0.0/16
```

出力:

```
{  
  "VpnConnections": [  
    {  
      "CustomerGatewayConfiguration": "...configuration information...",  
      "CustomerGatewayId": "cgw-01234567abcde1234",  
      "Category": "VPN",  
      "State": "modifying",  
      "Type": "ipsec.1",  
      "VpnConnectionId": "vpn-1122334455aabbccd",  
      "TransitGatewayId": "tgw-00112233445566aab",  
      "Options": {  
        "EnableAcceleration": false,  
        "StaticRoutesOnly": true,  
        "LocalIpv4NetworkCidr": "10.0.0.0/16",  
        "RemoteIpv4NetworkCidr": "0.0.0.0/0",  
        "TunnelInsideIpVersion": "ipv4"  
      },  
      "Routes": [],  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "CanadaVPN"  
        }  
      ],  
      "VgwTelemetry": [  
        {  
          "AcceptedRouteCount": 0,  
          "LastStatusChange": "2020-07-29T10:35:11.000Z",  
          "OutsideIpAddress": "203.0.113.3",  
          "Status": "DOWN",  
          "StatusMessage": ""  
        },  
        {  
          "AcceptedRouteCount": 0,  
          "LastStatusChange": "2020-09-02T09:09:33.000Z",  
          "OutsideIpAddress": "203.0.113.5",  
          "Status": "UP",  
        }  
      ]  
    }  
  ]  
}
```

```
    "StatusMessage": ""
  }
]
}
}
```

詳細については、AWS Site-to-Site VPN ユーザーガイドの [Site-to-SiteVPN「接続オプションの変更」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [ModifyVpnConnectionOptions](#)」の「」を参照してください。

modify-vpn-connection

次のコード例は、modify-vpn-connection を使用する方法を示しています。

AWS CLI

VPN接続を変更するには

次のmodify-vpn-connection例では、仮想プライベートゲートウェイ VPN vpn-12345678901234567への接続のターゲットゲートウェイを変更しますvgw-11223344556677889。

```
aws ec2 modify-vpn-connection \  
  --vpn-connection-id vpn-12345678901234567 \  
  --vpn-gateway-id vgw-11223344556677889
```

出力:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "...configuration information...",  
    "CustomerGatewayId": "cgw-aabbccdde1122334",  
    "Category": "VPN",  
    "State": "modifying",  
    "Type": "ipsec.1",  
    "VpnConnectionId": "vpn-12345678901234567",  
    "VpnGatewayId": "vgw-11223344556677889",  
    "Options": {  
      "StaticRoutesOnly": false  
    }  
  }  
}
```

```

    },
    "VgwTelemetry": [
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2019-07-17T07:34:00.000Z",
        "OutsideIpAddress": "18.210.3.222",
        "Status": "DOWN",
        "StatusMessage": "IPSEC IS DOWN"
      },
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2019-07-20T21:20:16.000Z",
        "OutsideIpAddress": "34.193.129.33",
        "Status": "DOWN",
        "StatusMessage": "IPSEC IS DOWN"
      }
    ]
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyVpnConnection](#)」の「」を参照してください。

modify-vpn-tunnel-certificate

次の例は、modify-vpn-tunnel-certificate を使用する方法を説明しています。

AWS CLI

VPNトンネル証明書をローテーションするには

次のmodify-vpn-tunnel-certificate例では、VPN接続の指定されたトンネルの証明書をローテーションします。

```

aws ec2 modify-vpn-tunnel-certificate \
  --vpn-tunnel-outside-ip-address 203.0.113.17 \
  --vpn-connection-id vpn-12345678901234567

```

出力:

```

{
  "VpnConnection": {

```

```

"CustomerGatewayConfiguration": ...configuration information...,
"CustomerGatewayId": "cgw-aabbccdde1122334",
"Category": "VPN",
"State": "modifying",
"Type": "ipsec.1",
"VpnConnectionId": "vpn-12345678901234567",
"VpnGatewayId": "vgw-11223344556677889",
"Options": {
  "StaticRoutesOnly": false
},
"VgwTelemetry": [
  {
    "AcceptedRouteCount": 0,
    "LastStatusChange": "2019-09-11T17:27:14.000Z",
    "OutsideIpAddress": "203.0.113.17",
    "Status": "DOWN",
    "StatusMessage": "IPSEC IS DOWN",
    "CertificateArn": "arn:aws:acm:us-east-1:123456789101:certificate/
c544d8ce-20b8-4fff-98b0-example"
  },
  {
    "AcceptedRouteCount": 0,
    "LastStatusChange": "2019-09-11T17:26:47.000Z",
    "OutsideIpAddress": "203.0.114.18",
    "Status": "DOWN",
    "StatusMessage": "IPSEC IS DOWN",
    "CertificateArn": "arn:aws:acm:us-
east-1:123456789101:certificate/5ab64566-761b-4ad3-b259-example"
  }
]
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVpnTunnelCertificate](#)」の「」を参照してください。

modify-vpn-tunnel-options

次のコード例は、modify-vpn-tunnel-options を使用する方法を示しています。

AWS CLI

VPN接続のトンネルオプションを変更するには

次のmodify-vpn-tunnel-options例では、指定されたトンネルとVPN接続に許可されているDiffie-Hellman グループを更新します。

```
aws ec2 modify-vpn-tunnel-options \  
  --vpn-connection-id vpn-12345678901234567 \  
  --vpn-tunnel-outside-ip-address 203.0.113.17 \  
  --tunnel-options Phase1DHGroupNumbers=[{Value=14},{Value=15},{Value=16},  
{Value=17},{Value=18}],Phase2DHGroupNumbers=[{Value=14},{Value=15},{Value=16},  
{Value=17},{Value=18}]
```

出力:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "...configuration information...",  
    "CustomerGatewayId": "cgw-aabbccdde1122334",  
    "Category": "VPN",  
    "State": "available",  
    "Type": "ipsec.1",  
    "VpnConnectionId": "vpn-12345678901234567",  
    "VpnGatewayId": "vgw-11223344556677889",  
    "Options": {  
      "StaticRoutesOnly": false,  
      "TunnelOptions": [  
        {  
          "OutsideIpAddress": "203.0.113.17",  
          "Phase1DHGroupNumbers": [  
            {  
              "Value": 14  
            },  
            {  
              "Value": 15  
            },  
            {  
              "Value": 16  
            },  
            {  
              "Value": 17  
            },  
            {  
              "Value": 18  
            }  
          ]  
        }  
      ],  
    }  
  }  
}
```

```
        "Phase2DHGroupNumbers": [
            {
                "Value": 14
            },
            {
                "Value": 15
            },
            {
                "Value": 16
            },
            {
                "Value": 17
            },
            {
                "Value": 18
            }
        ],
        {
            "OutsideIpAddress": "203.0.114.19"
        }
    ],
},
"VgwTelemetry": [
    {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2019-09-10T21:56:54.000Z",
        "OutsideIpAddress": "203.0.113.17",
        "Status": "DOWN",
        "StatusMessage": "IPSEC IS DOWN"
    },
    {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2019-09-10T21:56:43.000Z",
        "OutsideIpAddress": "203.0.114.19",
        "Status": "DOWN",
        "StatusMessage": "IPSEC IS DOWN"
    }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyVpnTunnelOptions](#)」の「」を参照してください。

monitor-instances

次のコード例は、monitor-instances を使用する方法を示しています。

AWS CLI

インスタンスの詳細モニタリングを有効にするには

このコマンド例は、指定されたインスタンスの詳細モニタリングを有効にします。

コマンド:

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

出力:

```
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "Monitoring": {
        "State": "pending"
      }
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[MonitorInstances](#)」の「」を参照してください。

move-address-to-vpc

次の例は、move-address-to-vpc を使用する方法を説明しています。

AWS CLI

アドレスを に移動するには EC2VPC

この例では、Elastic IP アドレス 54.123.4.56 を EC2-VPC プラットフォームに移動します。

コマンド:

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

出力:

```
{
  "Status": "MoveInProgress"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[MoveAddressToVpc](#)」の「」を参照してください。

move-byoip-cidr-to-ipam

次の例は、move-byoip-cidr-to-ipam を使用する方法を説明しています。

AWS CLI

BYOIPCIDRを に転送するには IPAM

次のmove-byoip-cidr-to-ipam例では、BYOIPCIDRを に転送しますIPAM。

(Linux):

```
aws ec2 move-byoip-cidr-to-ipam \
  --region us-west-2 \
  --ipam-pool-id ipam-pool-0a03d430ca3f5c035 \
  --ipam-pool-owner 111111111111 \
  --cidr 130.137.249.0/24
```

(Windows):

```
aws ec2 move-byoip-cidr-to-ipam ^
  --region us-west-2 ^
  --ipam-pool-id ipam-pool-0a03d430ca3f5c035 ^
  --ipam-pool-owner 111111111111 ^
  --cidr 130.137.249.0/24
```

出力:

```
{
  "ByoipCidr": {
    "Cidr": "130.137.249.0/24",
    "State": "pending-transfer"
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「チュートリアル: 既存の BYOIPv4CIDR に転送IPAMする」](#)を参照してください。 VPC IPAM

- API 詳細については、AWS CLI 「コマンドリファレンス[MoveByoipCidrToIpam](#)」の「」を参照してください。

network-insights-access-scope

次のコード例は、network-insights-access-scope を使用する方法を示しています。

AWS CLI

Network Insights アクセススコープを作成するには

次のcreate-network-insights-access-scope例では、AWS アカウントにネットワークインサイトのアクセス範囲を作成します。

```
aws ec2 create-network-insights-access-scope \
  --cli-input-json file://access-scope-file.json
```

access-scope-file.json の内容:

```
{
  {
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "Resources": [
              "vpc-abcd12e3"
            ]
          }
        }
      }
    ]
  }
}
```

```

    }
  ],
  "ExcludePaths": [
    {
      "Source": {
        "ResourceStatement": {
          "ResourceTypes": [
            "AWS::EC2::InternetGateway"
          ]
        }
      }
    }
  ]
}

```

出力:

```

{
  "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111"
}{
  "NetworkInsightsAccessScope": {
    "NetworkInsightsAccessScopeId": "nis-123456789222",
    "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789222:network-insights-access-scope/nis-123456789222",
    "CreateDate": "2022-01-25T19:20:28.796000+00:00",
    "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
  },
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-04c0c0fbca737c404",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "Resources": [
              "vpc-abcd12e3"
            ]
          }
        }
      }
    ]
  },
  "ExcludePaths": [
    {

```

```

    "Source": {
      "ResourceStatement": {
        "ResourceTypes": [
          "AWS::EC2::InternetGateway"
        ]
      }
    }
  ]
}

```

詳細については、[「Network Access Analyzer ガイド」の「を使用した Network Access Analyzer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[NetworkInsightsAccessScope](#)」の「」を参照してください。

provision-byoip-cidr

次のコード例は、provision-byoip-cidr を使用する方法を示しています。

AWS CLI

アドレス範囲をプロビジョニングするには

次のprovision-byoip-cidr例では、で使用するパブリック IP アドレス範囲をプロビジョニングしています AWS。

```

aws ec2 provision-byoip-cidr \
  --cidr 203.0.113.25/24 \
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"

```

出力:

```

{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "State": "pending-provision"
  }
}

```

認証コンテキストのメッセージ文字列の作成の詳細については、「Amazon EC2ユーザーガイド」の「[独自の IP アドレスの持ち込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ProvisionByoipCidr](#)」の「」を参照してください。

provision-ipam-pool-cidr

次の例は、provision-ipam-pool-cidr を使用する方法を説明しています。

AWS CLI

IPAM プールCIDRに をプロビジョニングするには

次のprovision-ipam-pool-cidr例では、IPAMプールCIDRに をプロビジョニングします。

(Linux):

```
aws ec2 provision-ipam-pool-cidr \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --cidr 10.0.0.0/24
```

(Windows):

```
aws ec2 provision-ipam-pool-cidr ^  
  --ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --cidr 10.0.0.0/24
```

出力:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "10.0.0.0/24",  
    "State": "pending-provision"  
  }  
}
```

詳細については、「Amazon VPCIPAMユーザーガイド」の「[プールCIDRsへのプロビジョニング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ProvisionIpamPoolCidr](#)」の「」を参照してください。

purchase-host-reservation

次のコード例は、`purchase-host-reservation` を使用する方法を示しています。

AWS CLI

Dedicated Host Reservation を購入するには

この例では、アカウントで指定された Dedicated Host の指定された Dedicated Host 予約サービスを購入します。

コマンド:

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

出力:

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
    {
      "HourlyPrice": "1.499",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "HostIdSet": [
        "h-013abcd2a00cbd123"
      ],
      "HostReservationId": "hr-0d418a3a4ffc669ae",
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PurchaseHostReservation](#)」の「」を参照してください。

purchase-reserved-instances-offering

次の例は、`purchase-reserved-instances-offering` を使用する方法を説明しています。

AWS CLI

リザーブドインスタンスサービスを購入するには

このコマンド例は、リザーブドインスタンスサービスの購入を示し、サービス ID とインスタンス数を指定します。

コマンド:

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id ec06327e-dd07-46ee-9398-75b5fexample --instance-count 3
```

出力:

```
{
  "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
}
```

- API 詳細については、AWS CLI 「[コマンドリファレンスPurchaseReservedInstancesOffering](#)」の「」を参照してください。

purchase-scheduled-instances

次のコード例は、purchase-scheduled-instances を使用する方法を示しています。

AWS CLI

スケジュールされたインスタンスを購入するには

この例では、スケジュールされたインスタンスを購入します。

コマンド:

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-request.json
```

Purchase-request.json:

```
[
  {
```

```
    "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
    "InstanceCount": 1
  }
]
```

出力:

```
{
  "ScheduledInstanceSet": [
    {
      "AvailabilityZone": "us-west-2b",
      "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
      "HourlyPrice": "0.095",
      "CreateDate": "2016-01-25T21:43:38.612Z",
      "Recurrence": {
        "OccurrenceDaySet": [
          1
        ],
        "Interval": 1,
        "Frequency": "Weekly",
        "OccurrenceRelativeToEnd": false,
        "OccurrenceUnit": ""
      },
      "Platform": "Linux/UNIX",
      "TermEndDate": "2017-01-31T09:00:00Z",
      "InstanceCount": 1,
      "SlotDurationInHours": 32,
      "TermStartDate": "2016-01-31T09:00:00Z",
      "NetworkPlatform": "EC2-VPC",
      "TotalScheduledInstanceHours": 1696,
      "NextSlotStartTime": "2016-01-31T09:00:00Z",
      "InstanceType": "c4.large"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PurchaseScheduledInstances](#)」の「」を参照してください。

reboot-instances

次のコード例は、reboot-instances を使用する方法を示しています。

AWS CLI

Amazon EC2インスタンスを再起動するには

この例では、指定のインスタンスを再起動します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」でインスタンスの再起動方法を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootInstances](#)」の「」を参照してください。

register-image

次のコード例は、register-image を使用する方法を示しています。

AWS CLI

例 1: マニフェストファイルAMIを使用して を登録するには

次のregister-image例では、Amazon S3 で指定されたマニフェストファイルAMIを使用して を登録します。

```
aws ec2 register-image \  
  --name my-image \  
  --image-location my-s3-bucket/myimage/image.manifest.xml
```

出力:

```
{  
  "ImageId": "ami-1234567890EXAMPLE"  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon マシンイメージ \(AMI\)」](#)を参照してください。 EC2

例 2: ルートデバイスのスナップショットAMIを使用して を登録するには

次のregister-image例では、EBSルートボリュームの指定されたスナップショットAMIをデバイスとして使用して を登録します/dev/xvda。ブロックデバイスマッピングには、デバイスとして空の 100 GiB EBSボリュームも含まれます/dev/xvdf。

```
aws ec2 register-image \  
  --name my-image \  
  --root-device-name /dev/xvda \  
  --block-device-mappings DeviceName=/dev/  
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f} DeviceName=/dev/  
xvdf,Ebs={VolumeSize=100}
```

出力:

```
{  
  "ImageId": "ami-1a2b3c4d5eEXAMPLE"  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon マシンイメージ \(AMI\)」](#)を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterImage](#)」の「」を参照してください。

register-instance-event-notification-attributes

次の例は、register-instance-event-notification-attributes を使用する方法を説明しています。

AWS CLI

例 1: イベント通知にすべてのタグを含めるには

次のregister-instance-event-notification-attributes例には、イベント通知のすべてのタグが含まれています。

```
aws ec2 register-instance-event-notification-attributes \  
  --instance-tag-attribute IncludeAllTagsOfInstance=true
```

出力:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [],
    "IncludeAllTagsOfInstance": true
  }
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[インスタンスのスケジュールされたイベント](#)」を参照してください。

例 2: イベント通知に特定のタグを含めるには

次のregister-instance-event-notification-attributes例では、イベント通知に指定されたタグが含まれています。IncludeAllTagsOfInstance が の場合、タグを指定することはできませんtrue。

```
aws ec2 register-instance-event-notification-attributes \
  --instance-tag-attribute InstanceTagKeys="tag-key1","tag-key2"
```

出力:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [
      "tag-key1",
      "tag-key2"
    ],
    "IncludeAllTagsOfInstance": false
  }
}
```

詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「[インスタンスのスケジュールされたイベント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterInstanceEventNotificationAttributes](#)」の「」を参照してください。

register-transit-gateway-multicase-group-sources

次のコード例は、register-transit-gateway-multicase-group-sources を使用する方法を示しています。

AWS CLI

Transit Gateway マルチキャストグループにソースを登録するには。

次のregister-transit-gateway-multicast-group-sources例では、指定されたネットワークインターフェイスグループソースをマルチキャストグループに登録します。

```
aws ec2 register-transit-gateway-multicast-group-sources \  
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \  
  --group-ip-address 224.0.1.0 \  
  --network-interface-ids eni-07f290fc3c090cbae
```

出力:

```
{  
  "RegisteredMulticastGroupSources": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",  
    "RegisteredNetworkInterfaceIds": [  
      "eni-07f290fc3c090cbae"  
    ],  
    "GroupIpAddress": "224.0.1.0"  
  }  
}
```

詳細については、「[Transit Gateways ユーザーガイド](#)」の「[マルチキャストグループでソースを登録する](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスRegisterTransitGatewayMulticastGroupSources](#)」の「」を参照してください。

register-transit-gateway-multicast-group-members

次の例は、register-transit-gateway-multicast-group-members を使用する方法を説明しています。

AWS CLI

Transit Gateway マルチキャストドメインの関連付けに関する情報を表示するには

次のregister-transit-gateway-multicast-group-members例では、指定されたマルチキャストドメインの関連付けを返します。

```
aws ec2 register-transit-gateway-multicast-group-members \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \  
--group-ip-address 224.0.1.0 \  
--network-interface-ids eni-0e246d32695012e81
```

出力:

```
{  
  "RegisteredMulticastGroupMembers": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",  
    "RegisteredNetworkInterfaceIds": [  
      "eni-0e246d32695012e81"  
    ],  
    "GroupIpAddress": "224.0.1.0"  
  }  
}
```

詳細については、「Transit Gateways ユーザーガイド」の [「マルチキャストドメインの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RegisterTransitGatewayMulticastGroupMembers](#)」の「」を参照してください。

register-transit-gateway-multicast-group-sources

次の例は、register-transit-gateway-multicast-group-sources を使用する方法を説明しています。

AWS CLI

Transit Gateway マルチキャストグループにソースを登録するには。

次のregister-transit-gateway-multicast-group-sources例では、指定されたネットワークインターフェイスグループソースをマルチキャストグループに登録します。

```
aws ec2 register-transit-gateway-multicast-group-sources \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef79d6e597 \  
--group-ip-address 224.0.1.0 \  
--network-interface-ids eni-07f290fc3c090cbae
```

出力:

```
{
  "RegisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

詳細については、「Transit Gateways ガイド」の [「マルチキャストドメインの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RegisterTransitGatewayMulticastGroupSources](#)」の「」を参照してください。

reject-transit-gateway-peering-attachment

次の例は、`reject-transit-gateway-peering-attachment` を使用する方法を説明しています。

AWS CLI

Transit Gateway ピアリングアタッチメントを拒否するには

次の `reject-transit-gateway-peering-attachment` 例では、指定された Transit Gateway ピアリングアタッチメントリクエストを拒否します。 `--region` パラメータは、アクセプタトランジットゲートウェイがあるリージョンを指定します。

```
aws ec2 reject-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
  --region us-east-2
```

出力:

```
{
  "TransitGatewayPeeringAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
    "RequesterTgwInfo": {
      "TransitGatewayId": "tgw-123abc05e04123abc",
      "OwnerId": "123456789012",
      "Region": "us-west-2"
    }
  }
}
```

```
    },
    "AccepterTgwInfo": {
      "TransitGatewayId": "tgw-11223344aabbcc112",
      "OwnerId": "123456789012",
      "Region": "us-east-2"
    },
    "State": "rejecting",
    "CreationTime": "2019-12-09T11:50:31.000Z"
  }
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ピアリングアタッチメント」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RejectTransitGatewayPeeringAttachment](#)」の「」を参照してください。

reject-transit-gateway-vpc-attachment

次のコード例は、`reject-transit-gateway-vpc-attachment` を使用する方法を示しています。

AWS CLI

Transit Gateway アタッチメントを拒否するには VPC

次の`reject-transit-gateway-vpc-attachment`例では、指定された Transit Gateway VPC アタッチメントを拒否します。

```
aws ec2 reject-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

出力:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "VpcId": "vpc-07e8ffd50fEXAMPLE",
    "VpcOwnerId": "111122223333",
    "State": "pending",
    "SubnetIds": [
```

```

        "subnet-0752213d59EXAMPLE"
    ],
    "CreationTime": "2019-07-10T17:33:46.000Z",
    "Options": {
        "DnsSupport": "enable",
        "Ipv6Support": "disable"
    }
}
}
}

```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway の へのアタッチメント VPC」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RejectTransitGatewayVpcAttachment](#)」の「」を参照してください。

reject-transit-gateway-vpc-attachments

次の例は、`reject-transit-gateway-vpc-attachments` を使用する方法を説明しています。

AWS CLI

Transit Gateway アタッチメントを拒否するには VPC

次の`reject-transit-gateway-vpc-attachment`例では、指定された Transit Gateway VPC アタッチメントを拒否します。

```

aws ec2 reject-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE

```

出力:

```

{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0a34fe6b4fEXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "VpcId": "vpc-07e8ffd50fEXAMPLE",
    "VpcOwnerId": "111122223333",
    "State": "pending",
    "SubnetIds": [
      "subnet-0752213d59EXAMPLE"
    ]
  },
}

```



```
"CreationTime": "2019-07-10T17:33:46.000Z",
"Options": {
  "DnsSupport": "enable",
  "Ipv6Support": "disable"
}
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway の へのアタッチメント VPC」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RejectTransitGatewayVpcAttachments](#)」の「」を参照してください。

reject-vpc-endpoint-connections

次のコード例は、`reject-vpc-endpoint-connections` を使用する方法を示しています。

AWS CLI

インターフェイスエンドポイント接続リクエストを拒否するには

この例では、指定されたエンドポイントサービスの指定されたエンドポイント接続リクエストを拒否します。

コマンド:

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

出力:

```
{
  "Unsuccessful": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RejectVpcEndpointConnections](#)」の「」を参照してください。

reject-vpc-peering-connection

次の例は、`reject-vpc-peering-connection` を使用する方法を説明しています。

AWS CLI

VPCピアリング接続を拒否するには

この例では、指定されたVPCピアリング接続リクエストを拒否します。

コマンド:

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

出力:

```
{
  "Return": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RejectVpcPeeringConnection](#)」の「」を参照してください。

release-address

次の例は、release-address を使用する方法を説明しています。

AWS CLI

EC2-Classic の Elastic IP アドレスをリリースするには

この例では、EC2-Classic のインスタンスで使用する Elastic IP アドレスをリリースします。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 release-address --public-ip 198.51.100.0
```

EC2- の Elastic IP アドレスをリリースするにはVPC

この例では、 のインスタンスで使用する Elastic IP アドレスをリリースしますVPC。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReleaseAddress](#)」の「」を参照してください。

release-hosts

次のコード例は、`release-hosts` を使用する方法を示しています。

AWS CLI

アカウントから Dedicated ホストをリリースするには

アカウントから Dedicated ホストをリリースするには。ホスト上にあるインスタンスは、ホストを解放する前に停止または終了する必要があります。

コマンド:

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

出力:

```
{
  "Successful": [
    "h-0029d6e3cacf1b3da"
  ],
  "Unsuccessful": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReleaseHosts](#)」の「」を参照してください。

release-ipam-pool-allocation

次の例は、`release-ipam-pool-allocation` を使用する方法を説明しています。

AWS CLI

IPAM プール割り当てをリリースするには

この例では、プールを削除しようとしたが、IPAMプールに割り当てがある間はプールを削除できないというエラーを受け取ったIPAM委任された管理者です。このコマンドを使用してプール割り当てをリリースします。

次の点に注意してください。

このコマンドは、カスタム割り当てにのみ使用できます。リソースを削除せずにリソースの割り当てを削除するには、を使用してモニタリング状態を `false` に設定します [modify-ipam-resource-cidr](#)。このリクエストを完了するには、で取得できるIPAMプール ID が必要です [describe-ipam-pools](#)。また、で取得できる割り当て ID も必要です [get-ipam-pool-allocations](#)。割り当てを 1 つずつ削除しない場合は、IPAMプールを削除する `--cascade option` ときにを使用して、プール内の割り当てを削除する前に自動的にリリースできます。このコマンドを実行する前に、いくつかの前提条件があります。詳細については、「Amazon VPCIPAMユーザーガイド」の「[割り当てのリリース](#)」を参照してください。このコマンドを実行する `--region` は、割り当てがあるIPAMプールのロケールである必要があります。

次の `release-ipam-pool-allocation` 例では、IPAMプール割り当てをリリースします。

```
aws ec2 release-ipam-pool-allocation \
  --ipam-pool-id ipam-pool-07bdd12d7c94e4693 \
  --cidr 10.0.0.0/23 \
  --ipam-pool-allocation-id ipam-pool-alloc-0e66a1f730da54791b99465b79e7d1e89 \
  --region us-west-1
```

出力:

```
{
  "Success": true
}
```

割り当てをリリースしたら、[delete-ipam-pool](#) を実行することができます。

- API 詳細については、AWS CLI 「[コマンドリファレンスReleaseIpamPoolAllocation](#)」の「」を参照してください。

replace-iam-instance-profile-association

次の例は、`replace-iam-instance-profile-association` を使用する方法を説明しています。

AWS CLI

IAMインスタンスのインスタンスプロファイルを置き換えるには

この例では、という名前のIAMインスタンスプロファイル*iip-assoc-060bae234aac2e7fa*との関連付けによって表されるIAMインスタンスプロファイルを置き換えますAdminRole。

```
aws ec2 replace-iam-instance-profile-association \  
  --iam-instance-profile Name=AdminRole \  
  --association-id iip-assoc-060bae234aac2e7fa
```

出力:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-087711ddaf98f9489",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0b215292fab192820",  
    "IamInstanceProfile": {  
      "Id": "AIPAJLNLDX3AMYZNWYYAY",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ReplaceIamInstanceProfileAssociation](#)」の「」を参照してください。

replace-network-acl-association

次の例は、replace-network-acl-association を使用する方法を説明しています。

AWS CLI

サブネットACLに関連付けられているネットワークを置き換えるには

この例では、指定されたネットワークを、指定されたネットワークACL関連付けのサブネットACLに関連付けます。

コマンド:

```
aws ec2 replace-network-acl-association --association-id aiclassoc-e5b95c8c --  
network-acl-id acl-5fb85d36
```

出力:

```
{  
  "NewAssociationId": "aclassoc-3999875b"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReplaceNetworkAclAssociation](#)」の「」を参照してください。

replace-network-acl-entry

次のコード例は、replace-network-acl-entry を使用する方法を示しています。

AWS CLI

ネットワークACLエントリを置き換えるには

この例では、指定されたネットワークのエントリを置き換えますACL。新しいルール 100 では、UDPポート 53 (DNS) の 203.0.113.12/24 から関連するサブネットへの進入トラフィックが許可されます。

コマンド:

```
aws ec2 replace-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-  
number 100 --protocol udp --port-range From=53,To=53 --cidr-block 203.0.113.12/24 --  
rule-action allow
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReplaceNetworkAclEntry](#)」の「」を参照してください。

replace-route-table-association

次の例は、replace-route-table-association を使用する方法を説明しています。

AWS CLI

サブネットに関連付けられたルートテーブルを置き換えるには

この例では、指定されたルートテーブルを、指定されたルートテーブルの関連付けのサブネットに関連付けます。

コマンド:

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```

出力:

```
{  
  "NewAssociationId": "rtbassoc-3a1f0f58"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReplaceRouteTableAssociation](#)」の「」を参照してください。

replace-route

次のコード例は、replace-route を使用する方法を示しています。

AWS CLI

ルートを置き換えるには

この例では、指定されたルートテーブル内の指定されたルートを置き換えます。新しいルートは、指定されたに一致しCIDR、指定された仮想プライベートゲートウェイにトラフィックを送信します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReplaceRoute](#)」の「」を参照してください。

replace-transit-gateway-route

次のコード例は、replace-transit-gateway-route を使用する方法を示しています。

AWS CLI

指定された Transit Gateway ルートテーブルで指定されたルートを置き換えるには

次の`replace-transit-gateway-route`例では、指定された Transit Gateway ルートテーブルのルートを置き換えます。

```
aws ec2 replace-transit-gateway-route \  
  --destination-cidr-block 10.0.2.0/24 \  
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE \  
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

出力:

```
{  
  "Route": {  
    "DestinationCidrBlock": "10.0.2.0/24",  
    "TransitGatewayAttachments": [  
      {  
        "ResourceId": "vpc-4EXAMPLE",  
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",  
        "ResourceType": "vpc"  
      }  
    ],  
    "Type": "static",  
    "State": "active"  
  }  
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ReplaceTransitGatewayRoute](#)」の「」を参照してください。

report-instance-status

次の例は、`report-instance-status` を使用する方法を説明しています。

AWS CLI

インスタンスのステータスフィードバックを報告するには

この例では、指定されたインスタンスのステータスフィードバックを報告します。

コマンド:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --reason-codes unresponsive
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReportInstanceStatus](#)」の「」を参照してください。

request-spot-fleet

次の例は、request-spot-fleet を使用する方法を説明しています。

AWS CLI

最低料金でサブネット内のスポットフリートをリクエストするには

この例では、サブネットによってのみ異なる 2 つの起動仕様を持つスポットフリートリクエストを作成します。スポットフリートは、指定されたサブネット内のインスタンスを最低価格で起動します。インスタンスがデフォルトの で起動されると VPC、デフォルトでパブリック IP アドレスを受け取ります。インスタンスがデフォルト以外の で起動された場合 VPC、デフォルトではパブリック IP アドレスは受信されません。

スポットフリートリクエストでは、同じアベイラビリティーゾーンから異なるサブネットを指定することはできません。

コマンド:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",

```

```
    "SecurityGroups": [
      {
        "GroupId": "sg-1a2b3c4d"
      }
    ],
    "InstanceType": "m3.medium",
    "SubnetId": "subnet-1a2b3c4d, subnet-3c4d5e6f",
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
  }
]
```

出力:

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

アベイラビリティゾーンで最低料金でスポットフリートをリクエストするには

この例では、アベイラビリティゾーンによってのみ異なる 2 つの起動仕様を持つスポットフリートリクエストを作成します。スポットフリートは、指定されたアベイラビリティゾーン内のインスタンスを最低料金で起動します。アカウントが EC2-VPC のみをサポートしている場合、Amazon はアベイラビリティゾーンのデフォルトサブネットでスポットインスタンス EC2 を起動します。アカウントが EC2-Classic をサポートしている場合、Amazon はアベイラビリティゾーンの EC2-Classic でインスタンス EC2 を起動します。

コマンド:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
```

```
"ImageId": "ami-1a2b3c4d",
"KeyName": "my-key-pair",
"SecurityGroups": [
  {
    "GroupId": "sg-1a2b3c4d"
  }
],
"InstanceType": "m3.medium",
"Placement": {
  "AvailabilityZone": "us-west-2a, us-west-2b"
},
"IamInstanceProfile": {
  "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
}
}
]
```

サブネットでスポットインスタンスを起動し、パブリック IP アドレスを割り当てるには

この例では、デフォルト以外の で起動されたインスタンスにパブリックアドレスを割り当てます VPC。ネットワークインターフェイスを指定するときは、ネットワークインターフェイスを使用してサブネット ID とセキュリティグループ ID を含める必要があることに注意してください。

コマンド:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "InstanceType": "m3.medium",
      "NetworkInterfaces": [
        {
          "DeviceIndex": 0,
          "SubnetId": "subnet-1a2b3c4d",
```

```
        "Groups": [ "sg-1a2b3c4d" ],
        "AssociatePublicIpAddress": true
      }
    ],
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
    }
  }
]
}
```

分散配分戦略を使用してスポットフリートをリクエストするには

この例では、分散配分戦略を使用して 30 個のインスタンスを起動するスポットフリートリクエストを作成します。起動仕様はインスタンスタイプによって異なります。スポットフリートは、各タイプのインスタンスが 10 個になるように、インスタンスを起動仕様全体に分散します。

コマンド:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

Config.json:

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "c4.2xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "m3.2xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    },
    {
      "ImageId": "ami-1a2b3c4d",
      "InstanceType": "r3.2xlarge",
      "SubnetId": "subnet-1a2b3c4d"
    }
  ]
}
```

```
    }  
  ]  
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「スポットフリートリクエスト」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RequestSpotFleet](#)」の「」を参照してください。

request-spot-instances

次の例は、request-spot-instances を使用する方法を説明しています。

AWS CLI

スポットインスタンスをリクエストするには

この例では、指定されたアベイラビリティーゾーン内の 5 つのインスタンスに対して 1 回限りのスポットインスタンスリクエストを作成します。アカウントが EC2-VPC のみをサポートしている場合、Amazon は指定されたアベイラビリティーゾーンのデフォルトサブネットでインスタンス EC2 を起動します。アカウントが EC2-Classic をサポートしている場合、Amazon は指定されたアベイラビリティーゾーンの EC2-Classic でインスタンス EC2 を起動します。

コマンド:

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{  
  "ImageId": "ami-1a2b3c4d",  
  "KeyName": "my-key-pair",  
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
  "InstanceType": "m3.medium",  
  "Placement": {  
    "AvailabilityZone": "us-west-2a"  
  },  
  "IamInstanceProfile": {  
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
  }  
}
```

```
}
```

出力:

```
{
  "SpotInstanceRequests": [
    {
      "Status": {
        "UpdateTime": "2014-03-25T20:54:21.000Z",
        "Code": "pending-evaluation",
        "Message": "Your Spot request has been submitted for review, and is
pending evaluation."
      },
      "ProductDescription": "Linux/UNIX",
      "SpotInstanceRequestId": "sir-df6f405d",
      "State": "open",
      "LaunchSpecification": {
        "Placement": {
          "AvailabilityZone": "us-west-2a"
        },
        "ImageId": "ami-1a2b3c4d",
        "KeyName": "my-key-pair",
        "SecurityGroups": [
          {
            "GroupName": "my-security-group",
            "GroupId": "sg-1a2b3c4d"
          }
        ],
        "Monitoring": {
          "Enabled": false
        },
        "IamInstanceProfile": {
          "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        },
        "InstanceType": "m3.medium"
      },
      "Type": "one-time",
      "CreateTime": "2014-03-25T20:54:20.000Z",
      "SpotPrice": "0.050000"
    },
    ...
  ]
}
```

このコマンド例では、指定されたサブネット内の 5 つのインスタンスに対して 1 回限りのスポットインスタンスリクエストを作成します。Amazon は、指定されたサブネットでインスタンス EC2 を起動します。VPC がデフォルト以外の場合 VPC、インスタンスはデフォルトでパブリック IP アドレスを受信しません。

コマンド:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

出力:

```
{
  "SpotInstanceRequests": [
    {
      "Status": {
        "UpdateTime": "2014-03-25T22:21:58.000Z",
        "Code": "pending-evaluation",
        "Message": "Your Spot request has been submitted for review, and is pending evaluation."
      },
      "ProductDescription": "Linux/UNIX",
      "SpotInstanceRequestId": "sir-df6f405d",
      "State": "open",
      "LaunchSpecification": {
        "Placement": {
          "AvailabilityZone": "us-west-2a"
        }
      },
      "ImageId": "ami-1a2b3c4d",
      "SecurityGroups": [
```

```
        {
            "GroupName": "my-security-group",
            "GroupID": "sg-1a2b3c4d"
        }
    ]
    "SubnetId": "subnet-1a2b3c4d",
    "Monitoring": {
        "Enabled": false
    },
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    },
    "InstanceType": "m3.medium",
},
"Type": "one-time",
"CreateTime": "2014-03-25T22:21:58.000Z",
"SpotPrice": "0.050000"
},
...
]
```

この例では、デフォルト以外のもので起動するスポットインスタンスにパブリック IP アドレスを割り当てます VPC。ネットワークインターフェイスを指定するときは、ネットワークインターフェイスを使用してサブネット ID とセキュリティグループ ID を含める必要があることに注意してください。

コマンド:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "InstanceType": "m3.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d",
```



```
    "Groups": [ "sg-1a2b3c4d" ],
    "AssociatePublicIpAddress": true
  }
],
"IamInstanceProfile": {
  "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RequestSpotInstances](#)」の「」を参照してください。

reset-address-attribute

次の例は、reset-address-attribute を使用する方法を説明しています。

AWS CLI

Elastic IP アドレスに関連付けられたドメイン名属性をリセットするには

次のreset-address-attribute例では、Elastic IP アドレスのドメイン名属性をリセットします。

Linux :

```
aws ec2 reset-address-attribute \  
  --allocation-id eipalloc-abcdef01234567890 \  
  --attribute domain-name
```

Windows :

```
aws ec2 reset-address-attribute ^  
  --allocation-id eipalloc-abcdef01234567890 ^  
  --attribute domain-name
```

出力:

```
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",
```

```
    "AllocationId": "eipalloc-abcdef01234567890",
    "PtrRecord": "example.com."
    "PtrRecordUpdate": {
        "Value": "example.net.",
        "Status": "PENDING"
    }
  ]
}
```

保留中の変更をモニタリングするには、コマンドリファレンス[describe-addresses-attribute](#)の「」を参照してください。AWS CLI

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetAddressAttribute](#)」の「」を参照してください。

reset-ebs-default-kms-key-id

次の例は、reset-ebs-default-kms-key-id を使用する方法を説明しています。

AWS CLI

EBS暗号化CMKのデフォルトをリセットするには

次のreset-ebs-default-kms-key-id例では、現在のリージョンの AWS アカウントのEBS暗号化CMKのデフォルトをリセットします。

```
aws ec2 reset-ebs-default-kms-key-id
```

出力:

```
{
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513eEXAMPLE"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetEbsDefaultKmsKeyId](#)」の「」を参照してください。

reset-fpga-image-attribute

次のコード例は、reset-fpga-image-attribute を使用する方法を示しています。

AWS CLI

Amazon FPGAイメージの属性をリセットするには

この例では、指定された のロードアクセス許可をリセットしますAFI。

コマンド:

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --  
attribute loadPermission
```

出力:

```
{  
  "Return": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetFpgaImageAttribute](#)」の「」を参照してください。

reset-image-attribute

次のコード例は、reset-image-attribute を使用する方法を示しています。

AWS CLI

launchPermission 属性をリセットするには

この例では、指定された の launchPermission 属性をデフォルト値AMIにリセットします。デフォルトでは、AMIsはプライベートです。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute launchPermission
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetImageAttribute](#)」の「」を参照してください。

reset-instance-attribute

次の例は、reset-instance-attribute を使用する方法を説明しています。

AWS CLI

sourceDestCheck 属性をリセットするには

この例では、指定されたインスタンスの sourceDestCheck 属性をリセットします。インスタンスは `running` 状態にある必要がありますVPC。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute sourceDestCheck
```

カーネル属性をリセットするには

この例では、指定されたインスタンスの kernel 属性をリセットします。インスタンスは `stopped` の状態である必要があります。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute kernel
```

ramdisk 属性をリセットするには

この例では、指定されたインスタンスの ramdisk 属性をリセットします。インスタンスは `stopped` の状態である必要があります。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute ramdisk
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetInstanceAttribute](#)」の「」を参照してください。

reset-network-interface-attribute

次のコード例は、reset-network-interface-attribute を使用する方法を示しています。

AWS CLI

ネットワークインターフェイス属性をリセットするには

次のreset-network-interface-attribute例では、送信元/送信先チェック属性の値をにリセットしますtrue。

```
aws ec2 reset-network-interface-attribute \  
  --network-interface-id eni-686ea200 \  
  --source-dest-check
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetNetworkInterfaceAttribute](#)」の「」を参照してください。

reset-snapshot-attribute

次の例は、reset-snapshot-attribute を使用する方法を説明しています。

AWS CLI

スナップショット属性をリセットするには

この例では、スナップショットのボリューム作成アクセス許可をリセットしますsnap-1234567890abcdef0。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --  
attribute createVolumePermission
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetSnapshotAttribute](#)」の「」を参照してください。

restore-address-to-classic

次の例は、restore-address-to-classic を使用する方法を説明しています。

AWS CLI

アドレスを EC2-Classic に復元するには

この例では、Elastic IP アドレス 198.51.100.0 を EC2-Classic プラットフォームに復元します。

コマンド:

```
aws ec2 restore-address-to-classic --public-ip 198.51.100.0
```

出力:

```
{
  "Status": "MoveInProgress",
  "PublicIp": "198.51.100.0"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreAddressToClassic](#)」の「」を参照してください。

restore-image-from-recycle-bin

次のコード例は、restore-image-from-recycle-bin を使用する方法を示しています。

AWS CLI

ごみ箱からイメージを復元するには

次のrestore-image-from-recycle-bin例では、ごみ箱から AMI ami-0111222333444abcd を復元します。

```
aws ec2 restore-image-from-recycle-bin \
  --image-id ami-0111222333444abcd
```

出力:

```
{
  "Return": true
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[ごみ箱AMIsからの復旧](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreImageFromRecycleBin](#)」の「」を参照してください。

restore-managed-prefix-list-version

次の例は、restore-managed-prefix-list-version を使用する方法を説明しています。

AWS CLI

us-west-2**プレフィックスリストバージョンを復元するには**

以下では、指定されたプレフィックスリストのバージョン 1 からのエントリをrestore-managed-prefix-list-version復元します。

```
aws ec2 restore-managed-prefix-list-version \  
  --prefix-list-id pl-0123456abcabcabc1 \  
  --current-version 2 \  
  --previous-version 1
```

出力:

```
{  
  "PrefixList": {  
    "PrefixListId": "pl-0123456abcabcabc1",  
    "AddressFamily": "IPv4",  
    "State": "restore-in-progress",  
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-0123456abcabcabc1",  
    "PrefixListName": "vpc-cidrs",  
    "MaxEntries": 10,  
    "Version": 2,  
    "OwnerId": "123456789012"  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の [「マネージドプレフィックスリスト」](#) を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス [RestoreManagedPrefixListVersion](#)」の「」を参照してください。

restore-snapshot-from-recycle-bin

次のコード例は、restore-snapshot-from-recycle-bin を使用する方法を示しています。

AWS CLI

ごみ箱からスナップショットを復元するには

次の`restore-snapshot-from-recycle-bin`例では、ごみ箱からスナップショットを復元します。ごみ箱からスナップショットを復元すると、そのスナップショットはすぐに使用でき、ごみ箱から削除されます。復元されたスナップショットは、アカウント内の他のスナップショットと同じ方法で使用できます。

```
aws ec2 restore-snapshot-from-recycle-bin \  
  --snapshot-id snap-01234567890abcdef
```

このコマンドでは何も出力されません。

Amazon のごみ箱の詳細についてはEBS、「[Amazon EC2ユーザーガイド](#)」の「[ごみ箱からスナップショットを復元する](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスRestoreSnapshotFromRecycleBin](#)」の「」を参照してください。

`restore-snapshot-tier`

次のコード例は、`restore-snapshot-tier` を使用する方法を示しています。

AWS CLI

例 1: アーカイブされたスナップショットを永続的に復元するには

次の`restore-snapshot-tier`例では、指定されたスナップショットを完全に復元します。を指定`--snapshot-id`し、`permanent-restore`オプションを含めます。

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --permanent-restore
```

出力:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```


スナップショットのアーカイブの詳細については、「Amazon EC2ユーザーガイド」の「Amazon EBSスナップショットのアーカイブ <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>>」を参照してください。

例 2: アーカイブされたスナップショットを一時的に復元するには

次のrestore-snapshot-tier例では、指定されたスナップショットを一時的に復元します。--permanent-restore オプションを省略します。--snapshot-id と temporary-restore-daysを指定し、にスナップショットを復元する日数を指定します。は日数で指定temporary-restore-daysする必要があります。許可される範囲は 1 から です180。値を指定しないと、デフォルトで 1 に設定されます。

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --temporary-restore-days 5
```

出力:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "RestoreDuration": 5,  
  "IsPermanentRestore": false  
}
```

スナップショットのアーカイブの詳細については、「Amazon EC2ユーザーガイド」の「Amazon EBSスナップショットのアーカイブ <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>>」を参照してください。

例 3: 復元期間を変更するには

次のrestore-snapshot-tier例では、指定されたスナップショットの復元期間を 10 日に変更します。

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --temporary-restore-days 10
```

出力:

```
{  
  "SnapshotId": "snap-01234567890abcdef",
```

```
"RestoreDuration": 10,  
"IsPermanentRestore": false  
}
```

スナップショットのアーカイブの詳細については、「Amazon EC2ユーザーガイド」の「Amazon EBSスナップショットのアーカイブ <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>>」を参照してください。

例 4: 復元タイプを変更するには

次のrestore-snapshot-tier例では、指定されたスナップショットの復元タイプを一時スナップショットから永続的スナップショットに変更します。

```
aws ec2 restore-snapshot-tier \  
--snapshot-id snap-01234567890abcdef  
--permanent-restore
```

出力:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```

スナップショットのアーカイブの詳細については、「Amazon EC2ユーザーガイド」の「Amazon EBSスナップショットのアーカイブ <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreSnapshotTier](#)」の「」を参照してください。

revoke-client-vpn-ingress

次の例は、revoke-client-vpn-ingress を使用する方法を説明しています。

AWS CLI

クライアントVPNエンドポイントの承認ルールを取り消すには

次のrevoke-client-vpn-ingress例では、すべてのグループのインターネットアクセス(0.0.0.0/0)のルールを取り消します。

```
aws ec2 revoke-client-vpn-ingress \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --target-network-cidr 0.0.0.0/0 --revoke-all-groups
```

出力:

```
{  
  "Status": {  
    "Code": "revoking"  
  }  
}
```

詳細については、「クライアント管理者ガイド」の「[認証ルール](#)」を参照してください。AWS VPN

- API 詳細については、「コマンドリファレンス[RevokeClientVpnIngress](#)」の「」を参照してください。AWS CLI

revoke-security-group-egress

次の例は、`revoke-security-group-egress` を使用する方法を説明しています。

AWS CLI

例 1: 特定のアドレス範囲へのアウトバウンドトラフィックを許可するルールを削除するには

次のコマンド `revoke-security-group-egress` 例では、TCPポート 80 で指定されたアドレス範囲へのアクセスを許可するルールを削除します。

```
aws ec2 revoke-security-group-egress \  
  --group-id sg-026c12253ce15eff7 \  
  --ip-  
permissions [[{IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]]
```

このコマンドでは何も出力されません。

詳細については、「Amazon ユーザーガイド」の「[セキュリティグループ](#)」を参照してください。EC2

例 2: 特定のセキュリティグループへのアウトバウンドトラフィックを許可するルールを削除するには

次のコマンド revoke-security-group-egress 例では、TCPポート 80 で指定されたセキュリティグループへのアクセスを許可するルールを削除します。

```
aws ec2 revoke-security-group-egress \  
  --group-id sg-026c12253ce15eff7 \  
  --ip-permissions '[{"IpProtocol": "tcp", "FromPort": 443, "ToPort":  
  443, "UserIdGroupPairs": [{"GroupId": "sg-06df23a01ff2df86d"}]}]'
```

このコマンドでは何も出力されません。

詳細については、「Amazon ユーザーガイド」の「[セキュリティグループ](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [RevokeSecurityGroupEgress](#)」の「」を参照してください。

revoke-security-group-ingress

次のコード例は、revoke-security-group-ingress を使用する方法を示しています。

AWS CLI

例 1: セキュリティグループからルールを削除するには

次の revoke-security-group-ingress 例では、203.0.113.0/24 アドレス範囲の TCP ポート 22 アクセスを、デフォルトの の指定されたセキュリティグループから削除します VPC。

```
aws ec2 revoke-security-group-ingress \  
  --group-name mySecurityGroup \  
  --protocol tcp \  
  --port 22 \  
  --cidr 203.0.113.0/24
```

このコマンドは、成功すると出力を生成しません。

詳細については、「Amazon ユーザーガイド」の「[セキュリティグループ](#)」を参照してください。 EC2

例 2: IP アクセス許可セットを使用してルールを削除するには

次の revoke-security-group-ingress 例では、ip-permissions パラメータを使用して、ICMP メッセージを許可するインバウンドルール Destination Unreachable:

Fragmentation Needed and Don't Fragment was Set (タイプ 3、コード 4) を削除します。

```
aws ec2 revoke-security-group-ingress \  
  --group-id sg-026c12253ce15eff7 \  
  --ip-  
permissions IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges=[{CidrIp=0.0.0.0/0}]
```

このコマンドは、成功すると出力を生成しません。

詳細については、「Amazon ユーザーガイド」の「[セキュリティグループ](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[RevokeSecurityGroupIngress](#)」の「」を参照してください。

run-instances

次の例は、run-instances を使用する方法を説明しています。

AWS CLI

例 1: インスタンスをデフォルトサブネット内で起動するには

次のrun-instances例では、現在のリージョンのデフォルトサブネットt2.microに タイプの単一のインスタンスを起動し、リージョンのデフォルトサブネットに関連付けVPCます。(Linux) または SSH (RDPWindows) を使用してインスタンスに接続する予定がない場合は、キーペアはオプションです。

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --key-name MyKeyPair
```

出力:

```
{  
  "Instances": [  
    {  
      "AmiLaunchIndex": 0,  
      "ImageId": "ami-0abcdef1234567890",  
      "InstanceId": "i-1231231230abcdef0",
```

```
"InstanceType": "t2.micro",
"KeyName": "MyKeyPair",
"LaunchTime": "2018-05-10T08:05:20.000Z",
"Monitoring": {
  "State": "disabled"
},
"Placement": {
  "AvailabilityZone": "us-east-2a",
  "GroupName": "",
  "Tenancy": "default"
},
"PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
"PrivateIpAddress": "10.0.0.157",
"ProductCodes": [],
"PublicDnsName": "",
"State": {
  "Code": 0,
  "Name": "pending"
},
"StateTransitionReason": "",
"SubnetId": "subnet-04a636d18e83cfac",
"VpcId": "vpc-1234567890abcdef0",
"Architecture": "x86_64",
"BlockDeviceMappings": [],
"ClientToken": "",
"EbsOptimized": false,
"Hypervisor": "xen",
"NetworkInterfaces": [
  {
    "Attachment": {
      "AttachTime": "2018-05-10T08:05:20.000Z",
      "AttachmentId": "eni-attach-0e325c07e928a0405",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching"
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "MySecurityGroup",
        "GroupId": "sg-0598c7d356eba48d7"
      }
    ],
    "Ipv6Addresses": [],
```

```
    "MacAddress": "0a:ab:58:e0:67:e2",
    "NetworkInterfaceId": "eni-0c0a29997760baee7",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
    "PrivateIpAddress": "10.0.0.157",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
        "PrivateIpAddress": "10.0.0.157"
      }
    ],
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-04a636d18e83cfac",
    "VpcId": "vpc-1234567890abcdef0",
    "InterfaceType": "interface"
  }
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
  {
    "GroupName": "MySecurityGroup",
    "GroupId": "sg-0598c7d356eba48d7"
  }
],
"SourceDestCheck": true,
"StateReason": {
  "Code": "pending",
  "Message": "pending"
},
"Tags": [],
"VirtualizationType": "hvm",
"CpuOptions": {
  "CoreCount": 1,
  "ThreadsPerCore": 1
},
"CapacityReservationSpecification": {
  "CapacityReservationPreference": "open"
},
"MetadataOptions": {
  "State": "pending",
```

```
        "HttpTokens": "optional",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled"
    }
}
],
"OwnerId": "123456789012",
"ReservationId": "r-02a3f596d91211712"
}
```

例 2: デフォルトではないサブネットでインスタンスを起動し、パブリック IP アドレスを追加するには

次の `run-instances` の例では、デフォルト以外のサブネットで起動するインスタンスのパブリック IP アドレスをリクエストします。インスタンスは指定されたセキュリティグループに関連付けられます。

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --subnet-id subnet-08fc749671b2d077c \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --associate-public-ip-address \  
  --key-name MyKeyPair
```

`run-instances` の出力例については、例 1 を参照してください。

例 3: ボリュームを追加してインスタンスを起動するには

次の `run-instances` の例では、`mapping.json` で指定されたブロックデバイスマッピングを使用して、起動時に追加のボリュームをアタッチします。ブロックデバイスマッピングでは、EBS ボリューム、インスタンスストアボリューム、または EBS ボリュームとインスタンスストアボリュームの両方を指定できます。

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --subnet-id subnet-08fc749671b2d077c \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --key-name MyKeyPair \  
  --block-device-mappings file://mapping.json
```


mapping.json の内容。この例では /dev/sdh、サイズが 100 GiB の空の EBS ボリュームを追加します。

```
[
  {
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize": 100
    }
  }
]
```

mapping.json の内容。この例は、ephemeral1 をインスタンスストアボリュームとして追加しています。

```
[
  {
    "DeviceName": "/dev/sdc",
    "VirtualName": "ephemeral1"
  }
]
```

run-instances の出力例については、例 1 を参照してください。

ブロックデバイスマッピングの詳細については、「Amazon EC2 ユーザーガイド」の [「ブロックデバイスマッピング」](#) を参照してください。

例 4: インスタンスを起動し、作成時にタグを追加するには

次の run-instances の例では、キー webserver と値 production のタグをインスタンスに追加しています。コマンドは、 のキー cost-center と の値を持つタグ cc123 を、作成されたすべての EBS ボリューム (この場合はルートボリューム) にも適用します。

```
aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type t2.micro \
  --count 1 \
  --subnet-id subnet-08fc749671b2d077c \
  --key-name MyKeyPair \
  --security-group-ids sg-0b0384b66d7d692f9 \
```

```
--tag-specifications  
'ResourceType=instance,Tags=[{Key=webserver,Value=production}]'  
'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

run-instances の出力例については、例 1 を参照してください。

例 5: ユーザーデータを使用してインスタンスを起動するには

次の run-instances の例では、インスタンスの設定スクリプトを含む my_script.txt というファイルにユーザーデータを渡します。このスクリプトは起動時に実行されます。

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 1 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --user-data file://my_script.txt
```

run-instances の出力例については、例 1 を参照してください。

インスタンスユーザーデータの詳細については、「Amazon EC2ユーザーガイド」の [「インスタンスユーザーデータの操作」](#) を参照してください。

例 6: バーストパフォーマンスインスタンスを起動するには

次の run-instances の例では、unlimited クレジットオプションを使用して t2.micro インスタンスを起動しています。T2 インスタンスを起動する際に --credit-specification を指定しない場合、デフォルトは standard クレジットオプションです。T3 インスタンスを起動する際、デフォルトは unlimited クレジットオプションです。

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 1 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --credit-specification CpuCredits=unlimited
```

run-instances の出力例については、例 1 を参照してください。

バーストパフォーマンスインスタンスの詳細については、「Amazon EC2ユーザーガイド」の「[バーストパフォーマンスインスタンス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RunInstances](#)」の「」を参照してください。

run-scheduled-instances

次のコード例は、run-scheduled-instances を使用する方法を示しています。

AWS CLI

スケジュールされたインスタンスを起動するには

この例では、で指定されたスケジュールされたインスタンスを起動しますVPC。

コマンド:

```
aws ec2 run-scheduled-instances --scheduled-instance-  
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-  
specification file://launch-specification.json
```

Launch-specification.json:

```
{  
  "ImageId": "ami-12345678",  
  "KeyName": "my-key-pair",  
  "InstanceType": "c4.large",  
  "NetworkInterfaces": [  
    {  
      "DeviceIndex": 0,  
      "SubnetId": "subnet-12345678",  
      "AssociatePublicIpAddress": true,  
      "Groups": ["sg-12345678"]  
    }  
  ],  
  "IamInstanceProfile": {  
    "Name": "my-iam-role"  
  }  
}
```

出力:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

この例では、EC2-Classic で指定されたスケジュールされたインスタンスを起動します。

コマンド:

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

Launch-specification.json:

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": ["sg-12345678"],
  "InstanceType": "c4.large",
  "Placement": {
    "AvailabilityZone": "us-west-2b"
  }
  "IamInstanceProfile": {
    "Name": "my-iam-role"
  }
}
```

出力:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RunScheduledInstances](#)」の「」を参照してください。

search-local-gateway-routes

次のコード例は、search-local-gateway-routes を使用する方法を示しています。

AWS CLI

ローカルゲートウェイルートテーブルでルートを検索するには

次のsearch-local-gateway-routes例では、指定されたローカルゲートウェイルートテーブルで静的ルートを検索します。

```
aws ec2 search-local-gateway-routes \  
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \  
  --filters "Name=type,Values=static"
```

出力:

```
{  
  "Route": {  
    "DestinationCidrBlock": "0.0.0.0/0",  
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",  
    "Type": "static",  
    "State": "deleted",  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchLocalGatewayRoutes](#)」の「」を参照してください。

search-transit-gateway-multicast-groups

次のコード例は、search-transit-gateway-multicast-groups を使用する方法を示しています。

AWS CLI

1 つ以上の Transit Gateway マルチキャストグループを検索し、グループメンバーシップ情報を返すには

次のsearch-transit-gateway-multicast-groups例では、指定されたマルチキャストグループのグループメンバーシップを返します。

```
aws ec2 search-transit-gateway-multicast-groups \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-000fb24d04EXAMPLE
```

出力:

```
{  
  "MulticastGroups": [  
    {  
      "GroupIpAddress": "224.0.1.0",  
      "TransitGatewayAttachmentId": "tgw-attach-0372e72386EXAMPLE",  
      "SubnetId": "subnet-0187aff814EXAMPLE",  
      "ResourceId": "vpc-0065acced4EXAMPLE",  
      "ResourceType": "vpc",  
      "NetworkInterfaceId": "eni-03847706f6EXAMPLE",  
      "GroupMember": false,  
      "GroupSource": true,  
      "SourceType": "static"  
    }  
  ]  
}
```

詳細については、「Transit Gateways ガイド」の [「マルチキャストグループの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SearchTransitGatewayMulticastGroups](#)」の「」を参照してください。

search-transit-gateway-routes

次のコード例は、search-transit-gateway-routes を使用する方法を示しています。

AWS CLI

指定された Transit Gateway ルートテーブルでルートを検索するには

次のsearch-transit-gateway-routes例では、指定されたルートテーブルstaticでタイプ
のすべてのルートを返します。

```
aws ec2 search-transit-gateway-routes \  
--transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \  
--filters "Name=type,Values=static"
```

出力:

```
{
  "Routes": [
    {
      "DestinationCidrBlock": "10.0.2.0/24",
      "TransitGatewayAttachments": [
        {
          "ResourceId": "vpc-4EXAMPLE",
          "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
          "ResourceType": "vpc"
        }
      ],
      "Type": "static",
      "State": "active"
    },
    {
      "DestinationCidrBlock": "10.1.0.0/24",
      "TransitGatewayAttachments": [
        {
          "ResourceId": "vpc-4EXAMPLE",
          "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
          "ResourceType": "vpc"
        }
      ],
      "Type": "static",
      "State": "active"
    }
  ],
  "AdditionalRoutesAvailable": false
}
```

詳細については、[「Transit Gateways ガイド」の「Transit Gateway ルートテーブル」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchTransitGatewayRoutes](#)」の「」を参照してください。

send-diagnostic-interrupt

次の例は、send-diagnostic-interrupt を使用する方法を説明しています。

AWS CLI

診断割り込みを送信するには

次のsend-diagnostic-interrupt例では、指定されたインスタンスに診断割り込みを送信します。

```
aws ec2 send-diagnostic-interrupt \  
  --instance-id i-1234567890abcdef0
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendDiagnosticInterrupt](#)」の「」を参照してください。

start-instances

次のコード例は、start-instances を使用する方法を示しています。

AWS CLI

Amazon EC2インスタンスを起動するには

この例では、指定された Amazon EBS-backed インスタンスを開始します。

コマンド:

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

出力:

```
{  
  "StartingInstances": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "CurrentState": {  
        "Code": 0,  
        "Name": "pending"  
      },  
      "PreviousState": {  
        "Code": 80,  
        "Name": "stopped"  
      }  
    }  
  ]  
}
```



```
    }  
  ]  
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスの停止と起動」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartInstances](#)」の「」を参照してください。

start-network-insights-access-scope-analysis

次のコード例は、start-network-insights-access-scope-analysis を使用する方法を示しています。

AWS CLI

Network Insights アクセス範囲分析を開始するには

次のstart-network-insights-access-scope-analysis例では、AWS アカウントでスコープ分析を開始します。

```
aws ec2 start-network-insights-access-scope-analysis \  
  --region us-east-1 \  
  --network-insights-access-scope-id nis-123456789111
```

出力:

```
{  
  "NetworkInsightsAccessScopeAnalysis": {  
    "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",  
    "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-  
east-1:123456789012:network-insights-access-scope-analysis/nisa-123456789222",  
    "NetworkInsightsAccessScopeId": "nis-123456789111",  
    "Status": "running",  
    "StartDate": "2022-01-26T00:47:06.814000+00:00"  
  }  
}
```

詳細については、[「Network Access Analyzer ガイド」](#)の「[を使用した Network Access Analyzer の開始方法 AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartNetworkInsightsAccessScopeAnalysis](#)」の「」を参照してください。

start-network-insights-analysis

次のコード例は、start-network-insights-analysis を使用する方法を示しています。

AWS CLI

パスを分析するには

次のstart-network-insights-analysis例では、送信元と送信先間のパスを分析します。パス分析の結果を表示するには、describe-network-insights-analyses コマンドを使用します。

```
aws ec2 start-network-insights-analysis \  
  --network-insights-path-id nip-0b26f224f1d131fa8
```

出力:

```
{  
  "NetworkInsightsAnalysis": {  
    "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",  
    "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-  
insights-analysis/nia-02207aa13eb480c7a",  
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",  
    "StartDate": "2021-01-20T22:58:37.495Z",  
    "Status": "running"  
  }  
}
```

詳細については、「Reachability Analyzer ガイド」の「[の使用を開始する AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartNetworkInsightsAnalysis](#)」の「」を参照してください。

start-vpc-endpoint-service-private-dns-verification

次のコード例は、start-vpc-endpoint-service-private-dns-verification を使用する方法を示しています。

AWS CLI

DNS検証プロセスを開始するには

次のstart-vpc-endpoint-service-private-dns-verification例では、指定されたエンドポイントサービスDNSの検証プロセスを開始します。

```
aws ec2 start-vpc-endpoint-service-private-dns-verification \  
  --service-id vpce-svc-071afff70666e61e0
```

このコマンドでは何も出力されません。

詳細については、AWS PrivateLink 「ユーザーガイド」の[DNS「名前の管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartVpcEndpointServicePrivateDnsVerification](#)」の「」を参照してください。

stop-instances

次の例は、stop-instances を使用する方法を説明しています。

AWS CLI

例 1: Amazon EC2インスタンスを停止するには

次のstop-instances例では、指定された Amazon EBS-backed インスタンスを停止します。

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0
```

出力:

```
{  
  "StoppingInstances": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "CurrentState": {  
        "Code": 64,  
        "Name": "stopping"  
      },  
      "PreviousState": {  
        "Code": 16,  

```

```
        "Name": "running"
      }
    ]
  }
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「[インスタンスの停止と起動](#)」を参照してください。

例 2: Amazon EC2 インスタンスを休止するには

次の `stop-instances` 例では、インスタンスが休止用に有効になっており、休止の前提条件を満たしている場合、Amazon EBS-backed インスタンスを休止します。インスタンスが休止状態になると、インスタンスは停止されます。

```
aws ec2 stop-instances \
  --instance-ids i-1234567890abcdef0 \
  --hibernate
```

出力:

```
{
  "StoppingInstances": [
    {
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "InstanceId": "i-1234567890abcdef0",
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」で [オンデマンド Linux インスタンスの休止方法](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StopInstances](#)」の「」を参照してください。

terminate-client-vpn-connections

次の例は、`terminate-client-vpn-connections` を使用する方法を説明しています。

AWS CLI

クライアントVPNエンドポイントへの接続を終了するには

次の`terminate-client-vpn-connections`例では、クライアントVPNエンドポイントへの指定された接続を終了します。

```
aws ec2 terminate-client-vpn-connections \  
  --client-vpn-endpoint-id vpn-endpoint-123456789123abcde \  
  --connection-id cvpn-connection-04edd76f5201e0cb8
```

出力:

```
{  
  "ClientVpnEndpointId": "vpn-endpoint-123456789123abcde",  
  "ConnectionStatuses": [  
    {  
      "ConnectionId": "cvpn-connection-04edd76f5201e0cb8",  
      "PreviousStatus": {  
        "Code": "active"  
      },  
      "CurrentStatus": {  
        "Code": "terminating"  
      }  
    }  
  ]  
}
```

詳細については、[「クライアント管理者ガイド」の「クライアント接続AWS VPN」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TerminateClientVpnConnections](#)」の「」を参照してください。

terminate-instances

次のコード例は、`terminate-instances` を使用する方法を示しています。

AWS CLI

Amazon EC2インスタンスを終了するには

この例では、指定されたインスタンスを終了します。

コマンド:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

出力:

```
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

詳細については、AWS「コマンドラインインターフェイスユーザーガイド」の「Amazon EC2 インスタンスの使用」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TerminateInstances](#)」の「」を参照してください。

unassign-ipv6-addresses

次の例は、unassign-ipv6-addresses を使用する方法を説明しています。

AWS CLI

ネットワークインターフェイスからIPv6アドレスの割り当てを解除するには

この例では、指定されたネットワークインターフェイスから指定されたIPv6アドレスの割り当てを解除します。

コマンド:

```
aws ec2 unassign-ipv6-addresses --ipv6-  
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 --network-interface-id eni-23c49b68
```

出力:

```
{  
  "NetworkInterfaceId": "eni-23c49b68",  
  "UnassignedIpv6Addresses": [  
    "2001:db8:1234:1a00:3304:8879:34cf:4071"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス」の[UnassignIpv6Addresses](#)を参照してください。

unassign-private-ip-addresses

次の例は、unassign-private-ip-addresses を使用する方法を説明しています。

AWS CLI

セカンダリプライベート IP アドレスをネットワークインターフェイスから割り当て解除するには

この例では、指定されたプライベート IP アドレスを指定されたネットワークインターフェイスから割り当て解除します。コマンドが成功した場合、出力は返りません。

コマンド:

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-  
ip-addresses 10.0.0.82
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UnassignPrivateIpAddresses](#)」の「」を参照してください。

unassign-private-nat-gateway-address

次のコード例は、unassign-private-nat-gateway-address を使用する方法を示しています。

AWS CLI

プライベートNATゲートウェイからプライベート IP アドレスの割り当てを解除するには

次のunassign-private-nat-gateway-address例では、指定されたプライベートNATゲートウェイから指定された IP アドレスの割り当てを解除します。

```
aws ec2 unassign-private-nat-gateway-address \
  --nat-gateway-id nat-1234567890abcdef0 \
  --private-ip-addresses 10.0.20.197
```

出力:

```
{
  "NatGatewayId": "nat-0ee3edd182361f662",
  "NatGatewayAddresses": [
    {
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
      "PrivateIp": "10.0.20.197",
      "IsPrimary": false,
      "Status": "unassigning"
    }
  ]
}
```

詳細については、「Amazon VPCユーザーガイド [NAT](#)」の「[ゲートウェイ](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUnassignPrivateNatGatewayAddress](#)」の「」を参照してください。

unlock-snapshot

次のコード例は、unlock-snapshot を使用する方法を示しています。

AWS CLI

スナップショットをロック解除するには

次のunlock-snapshot例では、指定されたスナップショットをロック解除します。

```
aws ec2 unlock-snapshot \  
  --snapshot-id snap-0b5e733b4a8df6e0d
```

出力:

```
{  
  "SnapshotId": "snap-0b5e733b4a8df6e0d"  
}
```

詳細については、「Amazon ユーザーガイド」の「[スナップショットロック](#)」を参照してください。 EBS

- API 詳細については、AWS CLI 「コマンドリファレンス[UnlockSnapshot](#)」の「」を参照してください。

unmonitor-instances

次の例は、unmonitor-instances を使用する方法を説明しています。

AWS CLI

インスタンスの詳細モニタリングを無効にするには

このコマンド例は、指定されたインスタンスの詳細モニタリングを無効にします。

コマンド:

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

出力:

```
{  
  "InstanceMonitorings": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "Monitoring": {  
        "State": "disabling"  
      }  
    }  
  ]  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UnmonitorInstances](#)」の「」を参照してください。

update-security-group-rule-descriptions-egress

次の例は、update-security-group-rule-descriptions-egress を使用する方法を説明しています。

AWS CLI

アウトバウンドセキュリティグループルールの説明を更新するには

次のupdate-security-group-rule-descriptions-egress例では、指定されたポートとIPv4アドレス範囲のセキュリティグループルールの説明を更新します。説明 'Outbound HTTP access to server 2' は、ルールの既存の説明を置き換えます。

```
aws ec2 update-security-group-rule-descriptions-egress \  
  --group-id sg-02f0d35a850ba727f \  
  --ip-permissions  
  IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=203.0.113.0/24,Description="Outbound  
  HTTP access to server 2"}]
```

出力:

```
{  
  "Return": true  
}
```

詳細については、「Amazon ユーザーガイド」の「[セキュリティグループルール](#)」を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateSecurityGroupRuleDescriptionsEgress](#)」の「」を参照してください。

update-security-group-rule-descriptions-ingress

次の例は、update-security-group-rule-descriptions-ingress を使用する方法を説明しています。

AWS CLI

例 1: CIDRソースを使用してインバウンドセキュリティグループルールの説明を更新するには

次のupdate-security-group-rule-descriptions-ingress例では、指定されたポートとIPv4アドレス範囲のセキュリティグループルールの説明を更新します。説明 'SSH access from ABC office' は、ルールの既存の説明を置き換えます。

```
aws ec2 update-security-group-rule-descriptions-ingress \  
  --group-id sg-02f0d35a850ba727f \  
  --ip-permissions  
  IpProtocol=tcp,FromPort=22,ToPort=22,IpRanges='[{"CidrIp=203.0.113.0/16,Description="SSH  
access from corpnet"}]'
```

出力:

```
{  
  "Return": true  
}
```

詳細については、「Amazon ユーザーガイド」の [「セキュリティグループルール」](#) を参照してください。 EC2

例 2: プレフィックスリストソースを使用してインバウンドセキュリティグループルールの説明を更新するには

次のupdate-security-group-rule-descriptions-ingress例では、指定されたポートとプレフィックスリストのセキュリティグループルールの説明を更新します。説明 'SSH access from ABC office' は、ルールの既存の説明を置き換えます。

```
aws ec2 update-security-group-rule-descriptions-ingress \  
  --group-id sg-02f0d35a850ba727f \  
  --ip-permissions  
  IpProtocol=tcp,FromPort=22,ToPort=22,PrefixListIds='[{"PrefixListId=pl-12345678,Description=  
access from corpnet"}]'
```

出力:

```
{  
  "Return": true  
}
```

詳細については、「Amazon ユーザーガイド」の [「セキュリティグループルール」](#) を参照してください。 EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateSecurityGroupRuleDescriptionsIngress](#)」の「」を参照してください。

withdraw-byoip-cidr

次のコード例は、withdraw-byoip-cidr を使用する方法を示しています。

AWS CLI

アドレス範囲のアドバタイズを停止するには

次のwithdraw-byoip-cidr例では、指定されたアドレス範囲のアドバタイズを停止します。

```
aws ec2 withdraw-byoip-cidr
  --cidr 203.0.113.25/24
```

出力:

```
{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    "State": "advertised"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [WithdrawByoipCidr](#)」の「」を参照してください。

を使用した Amazon EC2 Instance Connect の例 AWS CLI

次のコード例は、Amazon EC2 Instance Connect AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

send-ssh-public-key

次の例は、send-ssh-public-key を使用する方法を説明しています。

AWS CLI

SSHパブリックキーをインスタンスに送信するには

次のsend-ssh-public-key例では、指定されたSSHパブリックキーを指定されたインスタンスに送信します。キーは、指定されたユーザーを認証するために使用されます。

```
aws ec2-instance-connect send-ssh-public-key \  
  --instance-id i-1234567890abcdef0 \  
  --instance-os-user ec2-user \  
  --availability-zone us-east-2b \  
  --ssh-public-key file://path/my-rsa-key.pub
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendSshPublicKey](#)」の「」を参照してください。

を使用した Amazon ECR の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますECR。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-check-layer-availability

次の例は、batch-check-layer-availability を使用する方法を説明しています。

AWS CLI

レイヤーの可用性を確認するには

次のbatch-check-layer-availability例では、cluster-autoscalerリポジトリsha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed内のダイジェストを持つレイヤーの可用性をチェックします。

```
aws ecr batch-check-layer-availability \  
  --repository-name cluster-autoscaler \  
  --layer-  
  digests sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

出力:

```
{  
  "layers": [  
    {  
      "layerDigest":  
      "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed",  
      "layerAvailability": "AVAILABLE",  
      "layerSize": 2777,  
      "mediaType": "application/vnd.docker.container.image.v1+json"  
    }  
  ],  
  "failures": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchCheckLayerAvailability](#)」の「」を参照してください。

batch-delete-image

次のコード例は、batch-delete-image を使用する方法を示しています。

AWS CLI

例 1: イメージを削除するには

次のbatch-delete-image例では、アカウントのデフォルトレジストリの指定されたリポジトリpreciseに タグが付いたイメージを削除します。

```
aws ecr batch-delete-image \  
  --repository-name ubuntu \  
  --image-ids imageTag=precise
```

出力:

```
{  
  "failures": [],  
  "imageIds": [  
    {  
      "imageTag": "precise",  
      "imageDigest":  
"sha256:19665f1e6d1e504117a1743c0a3d3753086354a38375961f2e665416ef4b1b2f"  
    }  
  ]  
}
```

例 2: 複数のイメージを削除するには

次のbatch-delete-image例では、指定されたリポジトリteam1内の prodおよび でタグ付けされたすべてのイメージを削除します。

```
aws ecr batch-delete-image \  
  --repository-name MyRepository \  
  --image-ids imageTag=prod imageTag=team1
```

出力:

```
{  
  "imageIds": [  
    {
```

```
    "imageDigest": "sha256:123456789012",
    "imageTag": "prod"
  },
  {
    "imageDigest": "sha256:567890121234",
    "imageTag": "team1"
  }
],
"failures": []
}
```

詳細については、「Amazon ECRユーザーガイド」の「[イメージの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDeleteImage](#)」の「」を参照してください。

batch-get-image

次の例は、batch-get-image を使用する方法を説明しています。

AWS CLI

例 1: イメージを取得するには

次のbatch-get-image例では、アカウントのデフォルトレジストリcluster-autoscalerでという名前のリポジトリv1.13.6に タグが付いたイメージを取得します。

```
aws ecr batch-get-image \
  --repository-name cluster-autoscaler \
  --image-ids imageTag=v1.13.6
```

出力:

```
{
  "images": [
    {
      "registryId": "012345678910",
      "repositoryName": "cluster-autoscaler",
      "imageId": {
        "imageDigest":
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
        "imageTag": "v1.13.6"
      }
    }
  ],
}
```



```

      "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType\": \"application/vnd.docker.distribution.manifest.v2+json\",\n  \"config\": {\n    \"mediaType\": \"application/vnd.docker.container.image.v1+json\",\n    \"size\": 2777,\n    \"digest\": \"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed\"\n  },\n  \"layers\": [\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 17743696,\n      \"digest\": \"sha256:39fafc05754f195f134ca11ecdb1c9a691ab0848c697fffefeb5a85f900caaf6e1\"\n    },\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 2565026,\n      \"digest\": \"sha256:8c8a779d3a537b767ae1091fe6e00c2590afd16767aa6096d1b318d75494819f\"\n    },\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 28005981,\n      \"digest\": \"sha256:c44ba47496991c9982ee493b47fd25c252caabf2b4ae7dd679c9a27b6a3c8fb7\"\n    },\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",\n      \"size\": 775,\n      \"digest\": \"sha256:e2c388b44226544363ca007be7b896bcce1baebea04da23cbd165eac30be650f\"\n    }\n  ]\n},\n  \"failures\": []\n}"

```

例 2: 複数のイメージを取得するには

次のbatch-get-image例では、指定されたリポジトリteam1で prodと でタグ付けされたすべてのイメージの詳細を表示します。

```

aws ecr batch-get-image \
  --repository-name MyRepository \
  --image-ids imageTag=prod imageTag=team1

```

出力:

```

{
  "images": [
    {
      "registryId": "123456789012",
      "repositoryName": "MyRepository",
      "imageId": {

```

```
        "imageDigest": "sha256:123456789012",
        "imageTag": "prod"
    },
    "imageManifest": "manifestExample1"
},
{
    "registryId": "567890121234",
    "repositoryName": "MyRepository",
    "imageId": {
        "imageDigest": "sha256:123456789012",
        "imageTag": "team1"
    },
    "imageManifest": "manifestExample2"
}
],
"failures": []
}
```

詳細については、「Amazon ユーザーガイド」の「[画像](#)」を参照してください。 ECR

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetImage](#)」の「」を参照してください。

complete-layer-upload

次のコード例は、complete-layer-upload を使用する方法を示しています。

AWS CLI

イメージレイヤーのアップロードを完了するには

次のcomplete-layer-upload例では、layer-testリポジトリへのイメージレイヤーのアップロードを完了します。

```
aws ecr complete-layer-upload \
  --repository-name layer-test \
  --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
  --layer-digests 6cb64b8a-9378-0e33-2ab1-
b780fab8a9e9:48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e
```

出力:

```
{
```

```
"uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
"layerDigest":
"sha256:9a77f85878aa1906f2020a0ecdf7a7e962d57e882250acd773383224b3fe9a02",
"repositoryName": "layer-test",
"registryId": "130757420319"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CompleteLayerUpload](#)」の「」を参照してください。

create-repository

次のコード例は、create-repository を使用する方法を示しています。

AWS CLI

例 1: リポジトリを作成するには

次のcreate-repository例では、アカウントのデフォルトレジストリで指定された名前空間内にリポジトリを作成します。

```
aws ecr create-repository \
  --repository-name project-a/sample-repo
```

出力:

```
{
  "repository": {
    "registryId": "123456789012",
    "repositoryName": "project-a/sample-repo",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/sample-repo"
  }
}
```

詳細については、「Amazon ECRユーザーガイド」の「[リポジトリの作成](#)」を参照してください。

例 2: イメージタグのイミュータブルで設定されたリポジトリを作成するには

次のcreate-repository例では、アカウントのデフォルトレジストリでタグのイミュータビリティが設定されているリポジトリを作成します。

```
aws ecr create-repository \  
  --repository-name project-a/sample-repo \  
  --image-tag-mutability IMMUTABLE
```

出力:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "project-a/sample-repo",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/  
sample-repo",  
    "imageTagMutability": "IMMUTABLE"  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の [「Image Tag Mutability」](#) を参照してください。
ECR

例 3: スキャン設定で設定されたリポジトリを作成するには

次のcreate-repository例では、アカウントのデフォルトレジストリでイメージプッシュに対して脆弱性スキャンを実行するように構成されたリポジトリを作成します。

```
aws ecr create-repository \  
  --repository-name project-a/sample-repo \  
  --image-scanning-configuration scanOnPush=true
```

出力:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "project-a/sample-repo",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/  
sample-repo",  
    "imageScanningConfiguration": {  
      "scanOnPush": true  
    }  
  }  
}
```

詳細については、「Amazon ECRユーザーガイド」の「[イメージスキャン](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRepository](#)」の「」を参照してください。

delete-lifecycle-policy

次のコード例は、delete-lifecycle-policy を使用する方法を示しています。

AWS CLI

リポジトリのライフサイクルポリシーを削除するには

次のdelete-lifecycle-policy例では、hello-world リポジトリのライフサイクルポリシーを削除します。

```
aws ecr delete-lifecycle-policy \  
  --repository-name hello-world
```

出力:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "hello-world",  
  "lifecyclePolicyText": "{\n    \"rules\": [\n      {\n        \"rulePriority\": 1,\n        \"description\": \"Remove untagged images.\",\n        \"selection\": {\n          \"tagStatus\": \"untagged\",  
          \"countType\": \"sinceImagePushed\",  
          \"countUnit\": \"days\",  
          \"countNumber\": 10\n        },  
          \"action\": {\n            \"type\": \"expire\"\n          }\n        }\n      }\n    ]\n  }",  
  "lastEvaluatedAt": 0.0  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLifecyclePolicy](#)」の「」を参照してください。

delete-repository-policy

次のコード例は、delete-repository-policy を使用する方法を示しています。

AWS CLI

リポジトリのリポジトリポリシーを削除するには

次のdelete-repository-policy例では、リポジトリのcluster-autoscalerリポジトリポリシーを削除します。

```
aws ecr delete-repository-policy \  
  --repository-name cluster-autoscaler
```

出力:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "cluster-autoscaler",  
  "policyText": "{  
    \"Version\" : \"2008-10-17\",  
    \"Statement\" : [ {  
      \"Sid\" : \"allow public pull\",  
      \"Effect\" : \"Allow\",  
      \"Principal\" :  
        \"*\",  
      \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\",  
        \"ecr:GetDownloadUrlForLayer\" ]  
    } ]  
  }"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRepositoryPolicy](#)」の「」を参照してください。

delete-repository

次のコード例は、delete-repository を使用する方法を示しています。

AWS CLI

リポジトリを削除するには

次のコマンドフォースdelete-repositoryの例では、アカウントのデフォルトレジストリで指定されたリポジトリを削除します。リポジトリにイメージが含まれている場合は、--forceフラグが必要です。

```
aws ecr delete-repository \  
  --repository-name ubuntu \  
  --force
```

出力:

```
{  
  "repository": {
```

```
    "registryId": "123456789012",
    "repositoryName": "ubuntu",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/ubuntu"
  }
}
```

詳細については、「Amazon ECRユーザーガイド」の「[リポジトリの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRepository](#)」の「」を参照してください。

describe-image-scan-findings

次のコード例は、describe-image-scan-findings を使用する方法を示しています。

AWS CLI

イメージのスキャン結果を説明するには

次のdescribe-image-scan-findings例では、アカウントのデフォルトレジストリの指定されたリポジトリでイメージダイジェストを使用して、イメージのイメージスキャン結果を返します。

```
aws ecr describe-image-scan-findings \
  --repository-name sample-repo \
  --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

出力:

```
{
  "imageScanFindings": {
    "findings": [
      {
        "name": "CVE-2019-5188",
        "description": "A code execution vulnerability exists in the directory rehashing functionality of E2fsprogs e2fsck 1.45.4. A specially crafted ext4 directory can cause an out-of-bounds write on the stack, resulting in code execution. An attacker can corrupt a partition to trigger this vulnerability.",
        "uri": "http://people.ubuntu.com/~ubuntu-security/cve/CVE-2019-5188",
```

```
    "severity": "MEDIUM",
    "attributes": [
      {
        "key": "package_version",
        "value": "1.44.1-1ubuntu1.1"
      },
      {
        "key": "package_name",
        "value": "e2fsprogs"
      },
      {
        "key": "CVSS2_VECTOR",
        "value": "AV:L/AC:L/Au:N/C:P/I:P/A:P"
      },
      {
        "key": "CVSS2_SCORE",
        "value": "4.6"
      }
    ]
  },
  "imageScanCompletedAt": 1579839105.0,
  "vulnerabilitySourceUpdatedAt": 1579811117.0,
  "findingSeverityCounts": {
    "MEDIUM": 1
  }
},
"registryId": "123456789012",
"repositoryName": "sample-repo",
"imageId": {
  "imageDigest":
"sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
},
"imageScanStatus": {
  "status": "COMPLETE",
  "description": "The scan was completed successfully."
}
}
```

詳細については、「Amazon ECRユーザーガイド」の「[イメージスキャン](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeImageScanFindings](#)」の「」を参照してください。

describe-images

次の例は、describe-images を使用する方法を説明しています。

AWS CLI

リポジトリ内のイメージを記述するには

次のdescribe-images例では、cluster-autoscalerリポジトリ内のイメージに関する詳細をタグとともに表示しますv1.13.6。

```
aws ecr describe-images \  
  --repository-name cluster-autoscaler \  
  --image-ids imageTag=v1.13.6
```

出力:

```
{  
  "imageDetails": [  
    {  
      "registryId": "012345678910",  
      "repositoryName": "cluster-autoscaler",  
      "imageDigest":  
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",  
      "imageTags": [  
        "v1.13.6"  
      ],  
      "imageSizeInBytes": 48318255,  
      "imagePushedAt": 1565128275.0  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeImages](#)」の「」を参照してください。

describe-repositories

次の例は、describe-repositories を使用する方法を説明しています。

AWS CLI

レジストリ内のリポジトリを記述するには

この例は、アカウントのデフォルトレジストリ内のリポジトリを記述します。

コマンド:

```
aws ecr describe-repositories
```

出力:

```
{
  "repositories": [
    {
      "registryId": "012345678910",
      "repositoryName": "ubuntu",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/ubuntu"
    },
    {
      "registryId": "012345678910",
      "repositoryName": "test",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/test"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRepositories](#)」の「」を参照してください。

get-authorization-token

次のコード例は、get-authorization-token を使用する方法を示しています。

AWS CLI

デフォルトのレジストリの承認トークンを取得するには

次のコマンドget-authorization-token例では、デフォルトのレジストリの承認トークンを取得します。

```
aws ecr get-authorization-token
```

出力:

```
{
  "authorizationData": [
    {
      "authorizationToken": "QVdT0kN...",
      "expiresAt": 1448875853.241,
      "proxyEndpoint": "https://123456789012.dkr.ecr.us-west-2.amazonaws.com"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAuthorizationToken](#)」の「」を参照してください。

get-download-url-for-layer

次の例は、`get-download-url-for-layer` を使用する方法を説明しています。

AWS CLI

URLレイヤーのダウンロードを取得するには

次の`get-download-url-for-layer`例は、`cluster-autoscaler`リポジトリ `sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed`内のダイジェストを含むURLレイヤーのダウンロードを示しています。

```
aws ecr get-download-url-for-layer \
  --repository-name cluster-autoscaler \
  --layer-
digest sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

出力:

```
{
  "downloadUrl": "https://prod-us-west-2-starport-layer-bucket.s3.us-west-2.amazonaws.com/e501-012345678910-9cb60dc0-7284-5643-3987-da6dac0465f0/04620aac-66a5-4167-8232-55ee7ef6d565?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190814T220617Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=AKIA32P3D2JDNMVAJLGF%2F20190814%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Signature=9161345894947a1672467a0da7a1550f2f7157318312fe4941b59976239c3337",
```

```
"layerDigest":  
"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDownloadUrlForLayer](#)」の「」を参照してください。

get-lifecycle-policy-preview

次のコード例は、get-lifecycle-policy-preview を使用する方法を示しています。

AWS CLI

ライフサイクルポリシープレビューの詳細を取得するには

次のget-lifecycle-policy-preview例では、アカウントのデフォルトレジストリで指定されたリポジトリのライフサイクルポリシープレビューの結果を取得します。

コマンド:

```
aws ecr get-lifecycle-policy-preview \  
--repository-name "project-a/amazon-ecs-sample"
```

出力:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "project-a/amazon-ecs-sample",  
  "lifecyclePolicyText": "{\n    \"rules\": [\n      {\n        \"rulePriority\": 1,\n        \"description\": \"Expire images older than 14 days\",\n        \"selection\": {\n          \"tagStatus\": \"untagged\",\n          \"countType\": \"sinceImagePushed\",\n          \"countUnit\": \"days\",\n          \"countNumber\": 14\n        },\n        \"action\": {\n          \"type\": \"expire\"\n        }\n      }\n    ]\n  }",  
  "status": "COMPLETE",  
  "previewResults": [],  
  "summary": {\n    "expiringImageTotalCount": 0  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の [「ライフサイクルポリシー」](#) を参照してください。 ECR

- API 詳細については、AWS CLI 「コマンドリファレンス [GetLifecyclePolicyPreview](#)」の「」を参照してください。

get-lifecycle-policy

次の例は、get-lifecycle-policy を使用する方法を説明しています。

AWS CLI

ライフサイクルポリシーを取得するには

次のget-lifecycle-policy例では、アカウントのデフォルトレジストリで、指定されたリポジトリのライフサイクルポリシーの詳細を表示します。

```
aws ecr get-lifecycle-policy \  
  --repository-name "project-a/amazon-ecs-sample"
```

出力:

```
{  
  "registryId": "123456789012",  
  "repositoryName": "project-a/amazon-ecs-sample",  
  "lifecyclePolicyText": "{\"rules\": [{\"rulePriority\": 1, \"description\":  
  \"Expire images older than 14 days\", \"selection\": {\"tagStatus\": \"untagged\",  
  \"countType\": \"sinceImagePushed\", \"countUnit\": \"days\", \"countNumber\": 14},  
  \"action\": {\"type\": \"expire\"}}]}",  
  "lastEvaluatedAt": 1504295007.0  
}
```

詳細については、「Amazon ユーザーガイド」の [「ライフサイクルポリシー」](#) を参照してください。 ECR

- API 詳細については、AWS CLI 「コマンドリファレンス [GetLifecyclePolicy](#)」の「」を参照してください。

get-login-password

次の例は、get-login-password を使用する方法を説明しています。

AWS CLI

レジストリに認証するパスワードを取得するには

以下は、選択したコンテナクライアントで使用して、IAMプリンシパルがアクセスできる Amazon ECRレジストリに認証できるパスワード `get-login-password` を示しています。

```
aws ecr get-login-password
```

出力:

```
<password>
```

Docker で を使用するには CLI、 `get-login-password` コマンドの出力を `docker login` コマンドにパイプします。パスワードを取得するときは、Amazon ECRレジストリが存在するのと同じリージョンを指定する必要があります。

```
aws ecr get-login-password \  
  --region <region> \  
 \  
 | docker login \  
  --username AWS \  
  --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

詳細については、「Amazon ユーザーガイド」の [「レジストリ認証」](#) を参照してください。

ECR

- API 詳細については、AWS CLI 「コマンドリファレンス [GetLoginPassword](#)」の「」を参照してください。

get-login

次のコード例は、`get-login` を使用する方法を示しています。

AWS CLI

デフォルトのレジストリに Docker ログインコマンドを取得するには

この例では、デフォルトの Amazon ECRレジストリにログインするために使用できるコマンドを出力します。

コマンド:

```
aws ecr get-login
```

出力:

```
docker login -u AWS -p <password> -e none https://  
<aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

別のアカウントのレジストリにログインするには

この例では、他のアカウントに関連付けられた Amazon ECR レジストリにログインするために使用できる 1 つ以上のコマンドを出力します。

コマンド:

```
aws ecr get-login --registry-ids 012345678910 023456789012
```

出力:

```
docker login -u <username> -p <token-1> -e none <endpoint-1>  
docker login -u <username> -p <token-2> -e none <endpoint-2>
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLogin](#)」の「」を参照してください。

get-repository-policy

次の例は、get-repository-policy を使用する方法を説明しています。

AWS CLI

リポジトリのリポジトリポリシーを取得するには

次のget-repository-policy例では、リポジトリのcluster-autoscalerリポジトリポリシーの詳細を表示します。

```
aws ecr get-repository-policy \  
  --repository-name cluster-autoscaler
```

出力:

```
{
  "registryId": "012345678910",
  "repositoryName": "cluster-autoscaler",
  "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n    \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :\n    \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\",\n    \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRepositoryPolicy](#)」の「」を参照してください。

initiate-layer-upload

次の例は、initiate-layer-upload を使用する方法を説明しています。

AWS CLI

イメージレイヤーのアップロードを開始するには

次のinitiate-layer-upload例では、layer-testリポジトリへのイメージレイヤーのアップロードを開始します。

```
aws ecr initiate-layer-upload \  
  --repository-name layer-test
```

出力:

```
{
  "partSize": 10485760,
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[InitiateLayerUpload](#)」の「」を参照してください。

list-images

次のコード例は、list-images を使用する方法を示しています。

AWS CLI

リポジトリ内のイメージを一覧表示するには

次の `list-images` の例は、`cluster-autoscaler` リポジトリ内のイメージのリストを表示します。

```
aws ecr list-images \  
  --repository-name cluster-autoscaler
```

出力:

```
{  
  "imageIds": [  
    {  
      "imageDigest":  
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",  
      "imageTag": "v1.13.8"  
    },  
    {  
      "imageDigest":  
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",  
      "imageTag": "v1.13.7"  
    },  
    {  
      "imageDigest":  
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",  
      "imageTag": "v1.13.6"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListImages](#)」の「」を参照してください。

list-tags-for-resource

次の例は、`list-tags-for-resource` を使用する方法を説明しています。

AWS CLI

リポジトリのタグを一覧表示するには

次のlist-tags-for-resource例では、hello-worldリポジトリに関連付けられているタグのリストを表示します。

```
aws ecr list-tags-for-resource \  
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world
```

出力:

```
{  
  "tags": [  
    {  
      "Key": "Stage",  
      "Value": "Integ"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-image-scanning-configuration

次のコード例は、put-image-scanning-configuration を使用する方法を示しています。

AWS CLI

リポジトリのイメージスキャン設定を更新するには

次のput-image-scanning-configuration例では、指定されたりリポジトリのイメージスキャン設定を更新します。

```
aws ecr put-image-scanning-configuration \  
  --repository-name sample-repo \  
  --image-scanning-configuration scanOnPush=true
```

出力:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "sample-repo",
```

```
"imageScanningConfiguration": {
  "scanOnPush": true
}
```

詳細については、「Amazon ECRユーザーガイド」の「[イメージスキャン](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutImageScanningConfiguration](#)」の「」を参照してください。

put-image-tag-mutability

次のコード例は、put-image-tag-mutability を使用する方法を示しています。

AWS CLI

リポジトリのイメージタグのミュータビリティ設定を更新するには

次のput-image-tag-mutability例では、タグのイミュータブル性のために指定されたリポジトリを設定します。これにより、リポジトリ内のすべてのイメージタグが上書きされるのを防ぎます。

```
aws ecr put-image-tag-mutability \
  --repository-name hello-repository \
  --image-tag-mutability IMMUTABLE
```

出力:

```
{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageTagMutability": "IMMUTABLE"
}
```

詳細については、「Amazon ユーザーガイド」の「[Image Tag Mutability](#)」を参照してください。

ECR

- API 詳細については、AWS CLI 「コマンドリファレンス[PutImageTagMutability](#)」の「」を参照してください。

put-image

次の例は、put-image を使用する方法を説明しています。

AWS CLI

マニフェストを使用してイメージにタグを付け直すには

次のput-image例では、既存のイメージマニフェストを使用してhello-worldリポジトリに新しいタグを作成します。

```
aws ecr put-image \  
  --repository-name hello-world \  
  --image-tag 2019.08 \  
  --image-manifest file://hello-world.manifest.json
```

hello-world.manifest.json の内容:

```
{  
  "schemaVersion": 2,  
  "mediaType": "application/vnd.docker.distribution.manifest.v2+json",  
  "config": {  
    "mediaType": "application/vnd.docker.container.image.v1+json",  
    "size": 5695,  
    "digest":  
    "sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980"  
  },  
  "layers": [  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",  
      "size": 39096921,  
      "digest":  
      "sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295"  
    },  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",  
      "size": 57938,  
      "digest":  
      "sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed"  
    },  
    {  
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",  
      "size": 423,  
      "digest":  
      "sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed"  
    }  
  ]  
}
```

```
    "digest":
"sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610"
  },
  {
    "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
    "size": 680,
    "digest":
"sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c"
  },
  {
    "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
    "size": 162,
    "digest":
"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b"
  },
  {
    "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
    "size": 28268840,
    "digest":
"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f"
  },
  {
    "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
    "size": 35369152,
    "digest":
"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276"
  },
  {
    "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
    "size": 155,
    "digest":
"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85"
  },
  {
    "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
    "size": 28737,
    "digest":
"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4"
  },
  {
    "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
    "size": 190,
    "digest":
"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7"
```

```

    },
    {
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
      "size": 28748,
      "digest":
"sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee"
    }
  ]
}

```

出力:

```

{
  "image": {
    "registryId": "130757420319",
    "repositoryName": "hello-world",
    "imageId": {
      "imageDigest":
"sha256:8ece96b74f87652876199d83bd107d0435a196133af383ac54cb82b6cc5283ae",
      "imageTag": "2019.08"
    },
    "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType
\": \"application/vnd.docker.distribution.manifest.v2+json
\",,\n  \"config\": {\n    \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",,\n    \"size\": 5695,\n    \"digest\":
\"sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980\"\n
  },\n  \"layers\": [\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 39096921,\n      \"digest
\": \"sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 57938,\n      \"digest
\": \"sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed
\"\n    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 423,\n      \"digest\":
\"sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610\"\n
    },\n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",,\n
    \"size\": 680,\n    \"digest\":
\"sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c
\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 162,\n      \"digest
\": \"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b
\"\n
    },\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 28268840,\n      \"digest

```

```

\"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f
\"\\n  },\\n  {\\n    \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\\n    \"size\": 35369152,\\n    \"digest
\": \"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276\"\\n
  },\\n  {\\n    \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\\n    \"size\": 155,\\n    \"digest\":
\"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85\"\\n  },
\\n  {\\n    \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\\n    \"size\": 28737,\\n    \"digest\":
\"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4\"\\n  },
\\n  {\\n    \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\\n    \"size\": 190,\\n    \"digest\":
\"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7\"\\n  },
\\n  {\\n    \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\\n    \"size\": 28748,\\n    \"digest\":
\"sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee\"\\n
  }\\n ]\\n}\\n"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutImage](#)」の「」を参照してください。

put-lifecycle-policy

次の例は、put-lifecycle-policy を使用する方法を説明しています。

AWS CLI

ライフサイクルポリシーを作成するには

次のput-lifecycle-policy例では、アカウントのデフォルトレジストリに指定されたりポジトリのライフサイクルポリシーを作成します。

```

aws ecr put-lifecycle-policy \
  --repository-name "project-a/amazon-ecs-sample" \
  --lifecycle-policy-text "file://policy.json"

```

policy.json の内容:

```

{
  "rules": [

```

```

    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "untagged",
        "countType": "sinceImagePushed",
        "countUnit": "days",
        "countNumber": 14
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}

```

出力:

```

{
  "registryId": "<aws_account_id>",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\"rules\": [{\"rulePriority\": 1, \"description\": \"Expire images older than 14 days\", \"selection\": {\"tagStatus\": \"untagged\", \"countType\": \"sinceImagePushed\", \"countUnit\": \"days\", \"countNumber\": 14}, \"action\": {\"type\": \"expire\"}}]}"
}

```

詳細については、「Amazon ユーザーガイド」の「[ライフサイクルポリシー](#)」を参照してください。 ECR

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLifecyclePolicy](#)」の「」を参照してください。

set-repository-policy

次の例は、set-repository-policy を使用する方法を説明しています。

AWS CLI

リポジトリのリポジトリポリシーを設定するには

次のset-repository-policy例では、ファイルに含まれるリポジトリポリシーをcluster-autoscalerリポジトリにアタッチします。


```
aws ecr set-repository-policy \  
  --repository-name cluster-autoscaler \  
  --policy-text file://my-policy.json
```

my-policy.json の内容:

```
{  
  "Version" : "2008-10-17",  
  "Statement" : [  
    {  
      "Sid" : "allow public pull",  
      "Effect" : "Allow",  
      "Principal" : "*",  
      "Action" : [  
        "ecr:BatchCheckLayerAvailability",  
        "ecr:BatchGetImage",  
        "ecr:GetDownloadUrlForLayer"  
      ]  
    }  
  ]  
}
```

出力:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "cluster-autoscaler",  
  "policyText": "{  
    \"Version\" : \"2008-10-17\",  
    \"Statement\" : [ {  
      \"Sid\" : \"allow public pull\",  
      \"Effect\" : \"Allow\",  
      \"Principal\" :  
        \"*\",  
      \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\",  
        \"ecr:GetDownloadUrlForLayer\" ]  
    } ]  
  }"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [SetRepositoryPolicy](#)」の「」を参照してください。

start-image-scan

次の例は、start-image-scan を使用する方法を説明しています。

AWS CLI

イメージ脆弱性スキャンを開始するには

次のstart-image-scan例では、のイメージスキャンを開始し、指定されたリポジトリのイメージダイジェストで指定します。

```
aws ecr start-image-scan \  
  --repository-name sample-repo \  
  --image-  
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

出力:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "sample-repo",  
  "imageId": {  
    "imageDigest":  
"sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"  
  },  
  "imageScanStatus": {  
    "status": "IN_PROGRESS"  
  }  
}
```

詳細については、「Amazon ECRユーザーガイド」の「[イメージスキャン](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartImageScan](#)」の「」を参照してください。

start-lifecycle-policy-preview

次の例は、start-lifecycle-policy-preview を使用する方法を説明しています。

AWS CLI

ライフサイクルポリシープレビューを作成するには

次のstart-lifecycle-policy-preview例では、指定されたリポジトリのJSONファイルで定義されるライフサイクルポリシープレビューを作成します。

```
aws ecr start-lifecycle-policy-preview \  
  --repository-name "project-a/amazon-ecs-sample" \  
  --lifecycle-policy-text "file://policy.json"
```

policy.json の内容:

```
{  
  "rules": [  
    {  
      "rulePriority": 1,  
      "description": "Expire images older than 14 days",  
      "selection": {  
        "tagStatus": "untagged",  
        "countType": "sinceImagePushed",  
        "countUnit": "days",  
        "countNumber": 14  
      },  
      "action": {  
        "type": "expire"  
      }  
    }  
  ]  
}
```

出力:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "project-a/amazon-ecs-sample",  
  "lifecyclePolicyText": "{  
    \"rules\": [  
      {  
        \"rulePriority\": 1,  
        \"description\": \"Expire images older than 14  
days\",  
        \"selection\": {  
          \"tagStatus\": \"untagged\",  
          \"countType\": \"sinceImagePushed\",  
          \"countUnit  
\": \"days\",  
          \"countNumber\": 14  
        },  
        \"action\": {  
          \"type\": \"expire\"  
        }  
      }  
    ]  
  }",  
  "status": "IN_PROGRESS"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [StartLifecyclePolicyPreview](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リポジトリにタグを付けるには

次のtag-resource例では、hello-worldリポジトリIntegにキーStageと値を持つタグを設定します。

```
aws ecr tag-resource \  
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \  
  --tags Key=Stage,Value=Integ
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リポジトリのタグを解除するには

次のuntag-resource例では、キーを持つタグをhello-worldリポジトリStageから削除します。

```
aws ecr untag-resource \  
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \  
  --tag-keys Stage
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

upload-layer-part

次のコード例は、upload-layer-part を使用する方法を示しています。

AWS CLI

レイヤーパートをアップロードするには

以下では、イメージレイヤーパートをlayer-testリポジトリupload-layer-partにアップロードします。

```
aws ecr upload-layer-part \  
  --repository-name layer-test \  
  --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \  
  --part-first-byte 0 \  
  --part-last-byte 8323314 \  
  --layer-part-blob file:///var/lib/docker/image/overlay2/layerdb/sha256/  
ff986b10a018b48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e/layer.b64
```

出力:

```
{  
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",  
  "registryId": "012345678910",  
  "lastByteReceived": 8323314,  
  "repositoryName": "layer-test"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadLayerPart](#)」の「」を参照してください。

を使用した Amazon ECR Public の例 AWS CLI

次のコード例は、Amazon ECR Public AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-repository

次のコード例は、create-repository を使用する方法を示しています。

AWS CLI

例 1: パブリックレジストリにリポジトリを作成するには

次のcreate-repository例では、パブリックレジストリproject-a/nginx-web-appに という名前のリポジトリを作成します。

```
aws ecr-public create-repository \  
  --repository-name project-a/nginx-web-app
```

出力:

```
{  
  "repository": {  
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/  
nginx-web-app",  
    "registryId": "123456789012",  
    "repositoryName": "project-a/nginx-web-app",  
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/  
nginx-web-app",  
    "createdAt": "2024-07-01T21:08:55.131000+00:00"  
  },  
  "catalogData": {}  
}
```

詳細については、[「Amazon パブリックユーザーガイド」の「パブリックリポジトリの作成」](#)を参照してください。 ECR

例 2: リポジトリ内のイメージと互換性があるリポジトリ、システム、およびオペレーションアーキテクチャの内容を簡潔に説明して、パブリックレジストリにリポジトリを作成するには

次のcreate-repository例では、パブリックレジストリproject-a/nginx-web-appに という名前のリポジトリを作成し、リポジトリ内のイメージと互換性のあるリポジトリ、システム、およびオペレーションアーキテクチャの内容を簡単に説明します。

```
aws ecr-public create-repository \  
  --repository-name project-a/nginx-web-app \  
  --catalog-data 'description=My project-a ECR Public  
Repository,architectures=ARM,ARM 64,x86,x86-64,operatingSystems=Linux'
```

出力:

```
{  
  "repository": {  
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/  
nginx-web-app",  
    "registryId": "123456789012",  
    "repositoryName": "project-a/nginx-web-app",  
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/  
nginx-web-app",  
    "createdAt": "2024-07-01T21:23:20.455000+00:00"  
  },  
  "catalogData": {  
    "description": "My project-a ECR Public Repository",  
    "architectures": [  
      "ARM",  
      "ARM 64",  
      "x86",  
      "x86-64"  
    ],  
    "operatingSystems": [  
      "Linux"  
    ]  
  }  
}
```

詳細については、[「Amazon パブリックユーザーガイド」の「パブリックリポジトリの作成」](#)を参照してください。 ECR

例 3: パブリックレジストリにリポジトリを作成するには、`logImageBlob`、`aboutText`、`usageText` およびタグ情報とともに

次の`create-repository`例では、パブリックレジストリ`nginx-web-app`に`project-a/`という名前のリポジトリを、`logImageBlob`、`aboutText`、`usageText` タグ情報とともに作成します。

```
aws ecr-public create-repository \  
  --cli-input-json file://myfile.json
```

`myfile.json` の内容:

```
{  
  "repositoryName": "project-a/nginx-web-app",  
  "catalogData": {  
    "description": "My project-a ECR Public Repository",  
    "architectures": [  
      "ARM",  
      "ARM 64",  
      "x86",  
      "x86-64"  
    ],  
    "operatingSystems": [  
      "Linux"  
    ],  
    "logoImageBlob": "iVBORw0KGgoA<<truncated-for-better-reading>>ErkJggg==",  
    "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]  
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the  
  Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]  
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?  
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n* [ `2.0.20200722.0` , `2` , `latest` ](https://github.com/amazonlinux/container-images/  
blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n\n* [ `2.0.20200722.0-  
with-sources` , `2-with-sources` , `with-sources` ](https://github.com/  
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/  
Dockerfile)\n\n* [ `2018.03.0.20200602.1` , `2018.03` , `1` ](https://github.com/  
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/  
Dockerfile)\n\n* [ `2018.03.0.20200602.1-with-sources` , `2018.03-with-sources` ,  
  `1-with-sources` ](https://github.com/amazonlinux/container-images/  
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon  
Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed  
to provide a stable, secure, and high-performance execution environment for  
applications running on Amazon EC2. The full distribution includes packages that  
enable easy integration with AWS, including launch configuration tools and many
```



```
popular AWS libraries and tools. AWS provides ongoing security and maintenance
updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
contains a minimal set of packages. To install additional packages, [use `yum`]
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
\n\nFor information on security updates for Amazon Linux, please refer to [Amazon
Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon
Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker
Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,
which does not reflect the state of backported patches for vulnerabilities.\n
\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux
container images in any Docker based environment. Examples include, your laptop,
in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is
available under the [GNU General Public License, version 2.0](https://github.com/
aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages
are available under their own licenses; run `rpm -qi [package name]` or check
`/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for
details.\n\nAs with all Docker images, these likely also contain other software
which may be under other licenses (such as Bash, etc from the base distribution,
along with any direct or indirect dependencies of the primary software being
contained).\n\nSome additional license information which was able to be auto-
detected might be found in [the `repo-info` repository's `amazonlinux/` directory]
(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##
Security\n\nFor information on security updates for Amazon Linux, please refer
to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note
that Docker Hub's vulnerability scanning for Amazon Linux is currently based
on RPM versions, which does not reflect the state of backported patches for
vulnerabilities.",
    "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where
can I run Amazon Linux container images?\n\nYou can run Amazon Linux container
images in any Docker based environment. Examples include, your laptop, in Amazon
EC2 instances, and ECS clusters.\n\n## How do I install a software package from
Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the
`amazon-linux-extras` command. Packages can be installed with the `amazon-linux-
extras install <package>` command. Example: `amazon-linux-extras install rust1`\n
\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon
Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will
get ongoing updates from Amazon in the form of security updates, bug fix updates,
and other enhancements. Security bulletins for Amazon Linux are available at
https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon
Linux going forward?\n\nYes; in order to avoid any disruption to your existing
applications and to facilitate migration to Amazon Linux 2, AWS will provide
```

```
regular security updates for Amazon Linux 2018.03 AMI and container image for 2
years after the final LTS build is announced. You can also use all your existing
support channels such as AWS Support and Amazon Linux Discussion Forum to continue
to submit support requests."
```

```
  },
  "tags": [
    {
      "Key": "Name",
      "Value": "project-a/nginx-web-app"
    },
    {
      "Key": "Environment",
      "Value": "Prod"
    }
  ]
}
```

出力:

```
{
  "repository": {
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
    "createdAt": "2024-07-01T21:53:05.749000+00:00"
  },
  "catalogData": {
    "description": "My project-a ECR Public Repository",
    "architectures": [
      "ARM",
      "ARM 64",
      "x86",
      "x86-64"
    ],
    "operatingSystems": [
      "Linux"
    ],
    "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/
logo/23861450-4b9b-403c-9a4c-7aa0ef140bb8/2f9bf5a7-a32f-45b4-b5cd-c5770a35e6d7.png",
```

```
"aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]\n(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the\n Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]\n(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?\ntab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n* [ `2.0.20200722.0` , `2` , `latest` ](https://github.com/amazonlinux/container-images/\nblob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n\n* [ `2.0.20200722.0-\nwith-sources` , `2-with-sources` , `with-sources` ](https://github.com/\namazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/\nDockerfile)\n\n* [ `2018.03.0.20200602.1` , `2018.03` , `1` ](https://github.com/\namazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/\nDockerfile)\n\n* [ `2018.03.0.20200602.1-with-sources` , `2018.03-with-sources` ,\n `1-with-sources` ](https://github.com/amazonlinux/container-images/\nblob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon\n Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed\nto provide a stable, secure, and high-performance execution environment for\napplications running on Amazon EC2. The full distribution includes packages that\nenable easy integration with AWS, including launch configuration tools and many\npopular AWS libraries and tools. AWS provides ongoing security and maintenance\nupdates to all instances running Amazon Linux.\n\nThe Amazon Linux container image\ncontains a minimal set of packages. To install additional packages, [use `yum`]\n(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS\nprovides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/\namazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).\n\nFor information on security updates for Amazon Linux, please refer to [Amazon\n Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon\n Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker\n Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,\nwhich does not reflect the state of backported patches for vulnerabilities.\n\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux\ncontainer images in any Docker based environment. Examples include, your laptop,\nin Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is\navailable under the [GNU General Public License, version 2.0](https://github.com/\naws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages\nare available under their own licenses; run `rpm -qi [package name]` or check\n`/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for\ndetails.\n\nAs with all Docker images, these likely also contain other software\nwhich may be under other licenses (such as Bash, etc from the base distribution,\nalong with any direct or indirect dependencies of the primary software being\ncontained).\n\nSome additional license information which was able to be auto-\ndetected might be found in [the `repo-info` repository's `amazonlinux/` directory]\n(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##\n Security\n\nFor information on security updates for Amazon Linux, please refer\nto [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
```

and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions, which does not reflect the state of backported patches for vulnerabilities.",

```
"usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux container images in any Docker based environment. Examples include, your laptop, in Amazon EC2 instances, and ECS clusters.\n\n## How do I install a software package from Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the `amazon-linux-extras` command. Packages can be installed with the `amazon-linux-extras install <package>` command. Example: `amazon-linux-extras install rust1`\n\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will get ongoing updates from Amazon in the form of security updates, bug fix updates, and other enhancements. Security bulletins for Amazon Linux are available at https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon Linux going forward?\n\nYes; in order to avoid any disruption to your existing applications and to facilitate migration to Amazon Linux 2, AWS will provide regular security updates for Amazon Linux 2018.03 AMI and container image for 2 years after the final LTS build is announced. You can also use all your existing support channels such as AWS Support and Amazon Linux Discussion Forum to continue to submit support requests."
}
```

詳細については、「Amazon ECR パブリックユーザーガイド」の「[パブリックリポジトリの作成](#)」および「Amazon ECR パブリックユーザーガイド」の「[リポジトリカタログデータ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRepository](#)」の「」を参照してください。

delete-repository

次のコード例は、delete-repository を使用する方法を示しています。

AWS CLI

パブリックレジストリのリポジトリを削除するには

次のdelete-repository例では、パブリックレジストリproject-a/nginx-web-appから という名前のリポジトリを削除します。

```
aws ecr-public delete-repository \  
  --repository-name project-a/nginx-web-app
```

出力:

```
{  
  "repository": {  
    "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/  
nginx-web-app",  
    "registryId": "123456789012",  
    "repositoryName": "project-a/nginx-web-app",  
    "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/  
nginx-web-app",  
    "createdAt": "2024-07-01T22:14:50.103000+00:00"  
  }  
}
```

詳細については、[「Amazon パブリックユーザーガイド」の「パブリックリポジトリの削除」](#)を参照してください。 ECR

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRepository](#)」の「」を参照してください。

を使用した Amazon ECS の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますECS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-capacity-provider

次のコード例は、create-capacity-provider を使用する方法を示しています。

AWS CLI

キャパシティープロバイダーを作成するには

次の create-capacity-provider 例では、My という名前の Auto Scaling グループを使用し ASG、マネージドスケーリングとマネージド終了保護が有効になっているキャパシティープロバイダーを作成します。この設定は、Amazon ECS クラスターの自動スケーリングに使用されます。

```
aws ecs create-capacity-provider \  
  --name "MyCapacityProvider" \  
  --auto-scaling-group-provider "autoScalingGroupArn=arn:aws:autoscaling:us-  
east-1:123456789012:autoScalingGroup:57ffcb94-11f0-4d6d-  
bf60-3bac5EXAMPLE:autoScalingGroupName/  
MyASG,managedScaling={status=ENABLED,targetCapacity=100},managedTerminationProtection=ENABLED"
```

出力:

```
{  
  "capacityProvider": {  
    "capacityProviderArn": "arn:aws:ecs:us-east-1:123456789012:capacity-provider/  
MyCapacityProvider",  
    "name": "MyCapacityProvider",  
    "status": "ACTIVE",  
    "autoScalingGroupProvider": {  
      "autoScalingGroupArn": "arn:aws:autoscaling:us-  
east-1:123456789012:autoScalingGroup:57ffcb94-11f0-4d6d-  
bf60-3bac5EXAMPLE:autoScalingGroupName/MyASG",  
      "managedScaling": {  
        "status": "ENABLED",  
        "targetCapacity": 100,  
        "minimumScalingStepSize": 1,  
        "maximumScalingStepSize": 10000,  
        "instanceWarmupPeriod": 300  
      },  
      "managedTerminationProtection": "ENABLED"  
    },  
    "tags": []  
  }  
}
```

```
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS クラスターの自動スケーリング」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateCapacityProvider](#)」の「」を参照してください。

create-cluster

次のコード例は、create-cluster を使用する方法を示しています。

AWS CLI

例 1: 新しいクラスターを作成するには

以下の create-cluster の例は、クラスターを作成します。

```
aws ecs create-cluster \
  --cluster-name MyCluster
```

出力:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": []
  }
}
```

詳細については、「Amazon デベロッパーガイド」の[「クラスターの作成」](#)を参照してください。 ECS

例 2: キャパシティプロバイダーを使用して新しいクラスターを作成するには

次の `create-cluster` の例は、クラスターを作成し、2 つの既存のキャパシティープロバイダーをそのクラスターに関連付けます。キャパシティープロバイダーを作成するには、`create-capacity-provider` コマンドを使用します。デフォルトのキャパシティープロバイダー戦略を指定することは任意ですが、指定することをお勧めします。この例では、MyCluster という名前のクラスターを作成し、MyCapacityProvider1 と MyCapacityProvider2 をキャパシティープロバイダーに関連付けます。デフォルトのキャパシティープロバイダー戦略は、タスクを両方のキャパシティープロバイダーに均等に分散するように指定されています。

```
AWS ECS create-cluster --cluster-name MyCluster --capacity-providers
MyCapacityProvider1 MyCapacityProvider2 --default-capacity-
provider-strategy capacityProvider=MyCapacityProvider1,weight=1
capacityProvider=MyCapacityProvider2,weight=1
```

出力:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "PROVISIONING",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ],
    "capacityProviders": [
      "MyCapacityProvider1",
      "MyCapacityProvider2"
    ],
    "defaultCapacityProviderStrategy": [
      {
        "capacityProvider": "MyCapacityProvider1",
        "weight": 1,
        "base": 0
      },
      {
```



```

        "capacityProvider": "MyCapacityProvider2",
        "weight": 1,
        "base": 0
    }
],
"attachments": [
    {
        "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
        "type": "asp",
        "status": "PRECREATED",
        "details": [
            {
                "name": "capacityProviderName",
                "value": "MyCapacityProvider1"
            },
            {
                "name": "scalingPlanName",
                "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE111111"
            }
        ]
    },
    {
        "id": "ae592060-2382-4663-9476-b015c685593c",
        "type": "asp",
        "status": "PRECREATED",
        "details": [
            {
                "name": "capacityProviderName",
                "value": "MyCapacityProvider2"
            },
            {
                "name": "scalingPlanName",
                "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE222222"
            }
        ]
    }
],
"attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}

```

詳細については、「Amazon デベロッパーガイド」の [「クラスターキャパシティプロバイダー」](#) を参照してください。 ECS

例 3: 複数のタグを含む新しいクラスターを作成するには

次の `create-cluster` の例は、複数のタグを含むクラスターを作成します。短縮構文を使用したタグの追加の詳細については、AWS CLI ユーザーガイドの [AWS 「コマンドラインインターフェイスでの短縮構文の使用」](#) を参照してください。

```
aws ecs create-cluster \  
  --cluster-name MyCluster \  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

出力:

```
{  
  "cluster": {  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "clusterName": "MyCluster",  
    "status": "ACTIVE",  
    "registeredContainerInstancesCount": 0,  
    "pendingTasksCount": 0,  
    "runningTasksCount": 0,  
    "activeServicesCount": 0,  
    "statistics": [],  
    "tags": [  
      {  
        "key": "key1",  
        "value": "value1"  
      },  
      {  
        "key": "key2",  
        "value": "value2"  
      },  
      {  
        "key": "key3",  
        "value": "value3"  
      }  
    ]  
  }  
}
```

詳細については、「Amazon デベロッパーガイド」の「[クラスターの作成](#)」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCluster](#)」の「」を参照してください。

create-service

次のコード例は、create-service を使用する方法を示しています。

AWS CLI

例 1: Fargate タスクを使用してサービスを作成するには

次の create-service の例は、Fargate タスクを使用してサービスを作成する方法を示しています。

```
aws ecs create-service \  
  --cluster MyCluster \  
  --service-name MyService \  
  --task-definition sample-fargate:1 \  
  --desired-count 2 \  
  --launch-type FARGATE \  
  --platform-version LATEST \  
  --network-  
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],a  
\  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

出力:

```
{  
  "service": {  
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/  
MyService",  
    "serviceName": "MyService",  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "loadBalancers": [],  
    "serviceRegistries": [],  
    "status": "ACTIVE",  
    "desiredCount": 2,  
    "runningCount": 0,  
    "pendingCount": 0,
```

```
    "launchType": "FARGATE",
    "platformVersion": "LATEST",
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sample-fargate:1",
    "deploymentConfiguration": {
      "maximumPercent": 200,
      "minimumHealthyPercent": 100
    },
    "deployments": [
      {
        "id": "ecs-svc/1234567890123456789",
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sample-fargate:1",
        "desiredCount": 2,
        "pendingCount": 0,
        "runningCount": 0,
        "createdAt": 1557119253.821,
        "updatedAt": 1557119253.821,
        "launchType": "FARGATE",
        "platformVersion": "1.3.0",
        "networkConfiguration": {
          "awsvpcConfiguration": {
            "subnets": [
              "subnet-12344321"
            ],
            "securityGroups": [
              "sg-12344321"
            ],
            "assignPublicIp": "ENABLED"
          }
        }
      }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
    "events": [],
    "createdAt": 1557119253.821,
    "placementConstraints": [],
    "placementStrategy": [],
    "networkConfiguration": {
      "awsvpcConfiguration": {
        "subnets": [
          "subnet-12344321"
        ]
      }
    }
  }
}
```

```
    ],
    "securityGroups": [
      "sg-12344321"
    ],
    "assignPublicIp": "ENABLED"
  }
},
"schedulingStrategy": "REPLICA",
"tags": [
  {
    "key": "key1",
    "value": "value1"
  },
  {
    "key": "key2",
    "value": "value2"
  },
  {
    "key": "key3",
    "value": "value3"
  }
],
"enableECSTags": false,
"propagateTags": "NONE"
}
}
```

例 2: EC2 起動タイプを使用してサービスを作成するには

次の `create-service` 例は、EC2 起動タイプを使用するタスク `ecs-simple-service` でというサービスを作成する方法を示しています。このサービスは `sleep360` タスク定義を使用し、タスクのインスタンスを 1 つ保持します。

```
aws ecs create-service \
  --cluster MyCluster \
  --service-name ecs-simple-service \
  --task-definition sleep360:2 \
  --desired-count 1
```

出力:

```
{
```

```
"service": {
  "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/ecs-
simple-service",
  "serviceName": "ecs-simple-service",
  "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
  "loadBalancers": [],
  "serviceRegistries": [],
  "status": "ACTIVE",
  "desiredCount": 1,
  "runningCount": 0,
  "pendingCount": 0,
  "launchType": "EC2",
  "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sleep360:2",
  "deploymentConfiguration": {
    "maximumPercent": 200,
    "minimumHealthyPercent": 100
  },
  "deployments": [
    {
      "id": "ecs-svc/1234567890123456789",
      "status": "PRIMARY",
      "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sleep360:2",
      "desiredCount": 1,
      "pendingCount": 0,
      "runningCount": 0,
      "createdAt": 1557206498.798,
      "updatedAt": 1557206498.798,
      "launchType": "EC2"
    }
  ],
  "events": [],
  "createdAt": 1557206498.798,
  "placementConstraints": [],
  "placementStrategy": [],
  "schedulingStrategy": "REPLICA",
  "enableECSTags": false,
  "propagateTags": "NONE"
}
}
```

例 3: 外部のデプロイコントローラーを使用するサービスを作成するには

次の `create-service` の例は、外部のデプロイコントローラーを使用するサービスを作成します。

```
aws ecs create-service \  
  --cluster MyCluster \  
  --service-name MyService \  
  --deployment-controller type=EXTERNAL \  
  --desired-count 1
```

出力:

```
{  
  "service": {  
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/  
MyService",  
    "serviceName": "MyService",  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "loadBalancers": [],  
    "serviceRegistries": [],  
    "status": "ACTIVE",  
    "desiredCount": 1,  
    "runningCount": 0,  
    "pendingCount": 0,  
    "launchType": "EC2",  
    "deploymentConfiguration": {  
      "maximumPercent": 200,  
      "minimumHealthyPercent": 100  
    },  
    "taskSets": [],  
    "deployments": [],  
    "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/  
ecs.amazonaws.com/AWSServiceRoleForECS",  
    "events": [],  
    "createdAt": 1557128207.101,  
    "placementConstraints": [],  
    "placementStrategy": [],  
    "schedulingStrategy": "REPLICA",  
    "deploymentController": {  
      "type": "EXTERNAL"  
    },  
    "enableECSTags": false,  
    "propagateTags": "NONE"  
  }  
}
```

```
}
```

例 4: ロードバランサーの背後に新しいサービスを作成するには

次の `create-service` の例は、ロードバランサーの背後にサービスを作成する方法を示しています。コンテナインスタンスと同じリージョンに、ロードバランサーを設定する必要があります。この例では、`--cli-input-json` オプションと、次の内容 `ecs-simple-service-elb.json` を含む という JSON 入力ファイルを使用します。

```
{
  "serviceName": "ecs-simple-service-elb",
  "taskDefinition": "ecs-demo",
  "loadBalancers": [
    {
      "loadBalancerName": "EC2Contai-EcsElast-123456789012",
      "containerName": "simple-demo",
      "containerPort": 80
    }
  ],
  "desiredCount": 10,
  "role": "ecsServiceRole"
}
```

コマンド:

```
aws ecs create-service \
  --cluster MyCluster \
  --service-name ecs-simple-service-elb \
  --cli-input-json file://ecs-simple-service-elb.json
```

出力:

```
{
  "service": {
    "status": "ACTIVE",
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/ecs-demo:1",
    "pendingCount": 0,
    "loadBalancers": [
      {
        "containerName": "ecs-demo",
        "containerPort": 80,

```



```
        "loadBalancerName": "EC2Contai-EcsElast-123456789012"
      }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/ecsServiceRole",
    "desiredCount": 10,
    "serviceName": "ecs-simple-service-elb",
    "clusterArn": "arn:aws:ecs:<us-west-2:123456789012:cluster/MyCluster",
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/ecs-simple-
service-elb",
    "deployments": [
      {
        "status": "PRIMARY",
        "pendingCount": 0,
        "createdAt": 1428100239.123,
        "desiredCount": 10,
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/ecs-demo:1",
        "updatedAt": 1428100239.123,
        "id": "ecs-svc/1234567890123456789",
        "runningCount": 0
      }
    ],
    "events": [],
    "runningCount": 0
  }
}
```

詳細については、「Amazon ECSデベロッパーガイド」の「[サービスの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateService](#)」の「」を参照してください。

create-task-set

次の例は、create-task-set を使用する方法を説明しています。

AWS CLI

タスクセットを作成するには

次のcreate-task-set例では、外部デプロイコントローラーを使用するサービスにタスクセットを作成します。

```
aws ecs create-task-set \  
  --cluster MyCluster \  
  --service MyService \  
  --task-definition MyTaskDefinition:2 \  
  --network-  
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321]}"
```

出力:

```
{  
  "taskSet": {  
    "id": "ecs-svc/1234567890123456789",  
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
    "status": "ACTIVE",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
MyTaskDefinition:2",  
    "computedDesiredCount": 0,  
    "pendingCount": 0,  
    "runningCount": 0,  
    "createdAt": 1557128360.711,  
    "updatedAt": 1557128360.711,  
    "launchType": "EC2",  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "subnets": [  
          "subnet-12344321"  
        ],  
        "securityGroups": [  
          "sg-12344321"  
        ],  
        "assignPublicIp": "DISABLED"  
      }  
    },  
    "loadBalancers": [],  
    "serviceRegistries": [],  
    "scale": {  
      "value": 0.0,  
      "unit": "PERCENT"  
    },  
    "stabilityStatus": "STABILIZING",  
    "stabilityStatusAt": 1557128360.711  
  }  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTaskSet](#)」の「」を参照してください。

delete-account-setting

次の例は、delete-account-setting を使用する方法を説明しています。

AWS CLI

特定のIAMユーザーまたはIAMロールのアカウント設定を削除するには

次の例では、特定のIAMユーザーまたはIAMロールのアカウント設定delete-account-settingを削除します。

```
aws ecs delete-account-setting \  
  --name serviceLongArnFormat \  
  --principal-arn arn:aws:iam::123456789012:user/MyUser
```

出力:

```
{  
  "setting": {  
    "name": "serviceLongArnFormat",  
    "value": "enabled",  
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"  
  }  
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon リソースネーム \(ARNs \)」およびIDs](#)「」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccountSetting](#)」の「」を参照してください。

delete-attributes

次の例は、delete-attributes を使用する方法を説明しています。

AWS CLI

Amazon ECSリソースから 1 つ以上のカスタム属性を削除するには

以下では、コンテナインスタンスstackから という名前の属性delete-attributesを削除します。

```
aws ecs delete-attributes \  
  --attributes name=stack,targetId=arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

出力:

```
{  
  "attributes": [  
    {  
      "name": "stack",  
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",  
      "value": "production"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAttributes](#)」の「」を参照してください。

delete-capacity-provider

次の例は、delete-capacity-provider を使用する方法を説明しています。

AWS CLI

例 1: Amazon リソースネーム (ARN) を使用してキャパシティプロバイダーを削除するには

次のdelete-capacity-provider例では、キャパシティープロバイダーの Amazon リソースネーム (ARN) を指定して、キャパシティープロバイダーを削除します。ARN およびキャパシティープロバイダーの削除のステータスは、describe-capacity-providers コマンドを使用して取得できます。

```
aws ecs delete-capacity-provider \  

```

```
--capacity-provider arn:aws:ecs:us-west-2:123456789012:capacity-provider/  
ExampleCapacityProvider
```

出力:

```
{  
  "capacityProvider": {  
    "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-  
provider/ExampleCapacityProvider",  
    "name": "ExampleCapacityProvider",  
    "status": "ACTIVE",  
    "autoScalingGroupProvider": {  
      "autoScalingGroupArn": "arn:aws:autoscaling:us-  
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",  
      "managedScaling": {  
        "status": "ENABLED",  
        "targetCapacity": 100,  
        "minimumScalingStepSize": 1,  
        "maximumScalingStepSize": 10000  
      },  
      "managedTerminationProtection": "DISABLED"  
    },  
    "updateStatus": "DELETE_IN_PROGRESS",  
    "tags": []  
  }  
}
```

詳細については、「Amazon デベロッパーガイド」の「[クラスターキャパシティープロバイダー](#)」を参照してください。 ECS

例 2: 名前を使用してキャパシティープロバイダーを削除するには

次のdelete-capacity-provider例では、キャパシティープロバイダーの短縮名を指定して、キャパシティープロバイダーを削除します。短縮名とキャパシティープロバイダーの削除のステータスは、describe-capacity-providers コマンドを使用して取得できます。

```
aws ecs delete-capacity-provider \  
--capacity-provider ExampleCapacityProvider
```

出力:

```
{
  "capacityProvider": {
    "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/ExampleCapacityProvider",
    "name": "ExampleCapacityProvider",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
      "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
      "managedScaling": {
        "status": "ENABLED",
        "targetCapacity": 100,
        "minimumScalingStepSize": 1,
        "maximumScalingStepSize": 10000
      },
      "managedTerminationProtection": "DISABLED"
    },
    "updateStatus": "DELETE_IN_PROGRESS",
    "tags": []
  }
}
```

詳細については、「Amazon デベロッパーガイド」の「[クラスターキャパシティプロバイダー](#)」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCapacityProvider](#)」の「」を参照してください。

delete-cluster

次の例は、delete-cluster を使用する方法を説明しています。

AWS CLI

空のクラスターを削除するには

次の delete-cluster の例は、指定された空のクラスターを削除します。

```
aws ecs delete-cluster --cluster MyCluster
```

出力:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "status": "INACTIVE",
    "clusterName": "MyCluster",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0
    "statistics": [],
    "tags": []
  }
}
```

詳細については、「Amazon ECSデベロッパーガイド」の[「クラスターの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCluster](#)」の「」を参照してください。

delete-service

次の例は、delete-service を使用する方法を説明しています。

AWS CLI

サービスを削除するには

次の `ecs delete-service` の例は、指定されたサービスをクラスターから削除します。--force パラメータを含めると、タスクがゼロになっていない場合でも、サービスを削除することができます。

```
aws ecs delete-service --cluster MyCluster --service MyService1 --force
```

詳細については、「Amazon ECSデベロッパーガイド」の[「サービスの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteService](#)」の「」を参照してください。

delete-task-definitions

次の例は、delete-task-definitions を使用する方法を説明しています。

AWS CLI

タスク定義を削除するには

次のdelete-task-definitions例では、INACTIVEタスク定義を削除します。

```
aws ecs delete-task-definitions \  
  --task-definition curltest:1
```

出力:

```
{  
  "taskDefinitions": [  
    {  
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/  
curltest:1",  
      "containerDefinitions": [  
        {  
          "name": "ctest",  
          "image": "mreferre/eksutils",  
          "cpu": 0,  
          "portMappings": [],  
          "essential": true,  
          "entryPoint": [  
            "sh",  
            "-c"  
          ],  
          "command": [  
            "curl ${ECS_CONTAINER_METADATA_URI_V4}/task"  
          ],  
          "environment": [],  
          "mountPoints": [],  
          "volumesFrom": [],  
          "logConfiguration": {  
            "logDriver": "awslogs",  
            "options": {  
              "awslogs-create-group": "true",  
              "awslogs-group": "/ecs/curltest",  
              "awslogs-region": "us-east-1",
```



```
        "awslogs-stream-prefix": "ecs"
      }
    }
  ],
  "family": "curltest",
  "taskRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
  "networkMode": "awsvpc",
  "revision": 1,
  "volumes": [],
  "status": "DELETE_IN_PROGRESS",
  "compatibilities": [
    "EC2",
    "FARGATE"
  ],
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "cpu": "256",
  "memory": "512",
  "registeredAt": "2021-09-10T12:56:24.704000+00:00",
  "deregisteredAt": "2023-03-14T15:20:59.419000+00:00",
  "registeredBy": "arn:aws:sts::123456789012:assumed-role/Admin/jdoe"
}
],
"failures": []
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS タスク定義」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTaskDefinitions](#)」の「」を参照してください。

delete-task-set

次のコード例は、delete-task-set を使用する方法を示しています。

AWS CLI

タスクセットを削除するには

次のdelete-task-set例は、タスクセットを削除する方法を示しています。--force パラメータを含めると、タスクセットがゼロにスケールされていない場合でも、タスクセットを削除できます。

```
aws ecs delete-task-set \  
  --cluster MyCluster \  
  --service MyService \  
  --task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-  
svc/1234567890123456789 \  
  --force
```

出力:

```
{  
  "taskSet": {  
    "id": "ecs-svc/1234567890123456789",  
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
    "status": "DRAINING",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sample-fargate:2",  
    "computedDesiredCount": 0,  
    "pendingCount": 0,  
    "runningCount": 0,  
    "createdAt": 1557130260.276,  
    "updatedAt": 1557130290.707,  
    "launchType": "EC2",  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "subnets": [  
          "subnet-12345678"  
        ],  
        "securityGroups": [  
          "sg-12345678"  
        ],  
        "assignPublicIp": "DISABLED"  
      }  
    },  
    "loadBalancers": [],  
    "serviceRegistries": [],  
    "scale": {  
      "value": 0.0,  
      "unit": "PERCENT"  
    }  
  }  
}
```

```
    },
    "stabilityStatus": "STABILIZING",
    "stabilityStatusAt": 1557130290.707
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteTaskSet](#)」の「」を参照してください。

deregister-container-instance

次の例は、`deregister-container-instance` を使用する方法を説明しています。

AWS CLI

クラスターからコンテナインスタンスの登録を解除するには

次の `deregister-container-instance` 例では、指定されたクラスターからコンテナインスタンスを登録解除します。コンテナインスタンスでまだ実行中のタスクがある場合は、登録を解除する前にそれらのタスクを停止するか、`--force` オプションを使用する必要があります。

```
aws ecs deregister-container-instance \
  --cluster arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
  --container-instance arn:aws:ecs:us-west-2:123456789012:container-instance/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --force
```

出力:

```
{
  "containerInstance": {
    "remainingResources": [
      {
        "integerValue": 1024,
        "doubleValue": 0.0,
        "type": "INTEGER",
        "longValue": 0,
        "name": "CPU"
      },
      {
        "integerValue": 985,
        "doubleValue": 0.0,
```

```
        "type": "INTEGER",
        "longValue": 0,
        "name": "MEMORY"
    },
    {
        "type": "STRINGSET",
        "integerValue": 0,
        "name": "PORTS",
        "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678",
            "51679"
        ],
        "longValue": 0,
        "doubleValue": 0.0
    },
    {
        "type": "STRINGSET",
        "integerValue": 0,
        "name": "PORTS_UDP",
        "stringSetValue": [],
        "longValue": 0,
        "doubleValue": 0.0
    }
],
"agentConnected": true,
"attributes": [
    {
        "name": "ecs.capability.secrets.asm.environment-variables"
    },
    {
        "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
    },
    {
        "value": "ami-01a82c3fce2c3ba58",
        "name": "ecs.ami-id"
    },
    {
        "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
    },
    {
        "name": "com.amazonaws.ecs.capability.logging-driver.none"
    }
]
```

```
    },  
    {  
      "name": "ecs.capability.ecr-endpoint"  
    },  
    {  
      "name": "com.amazonaws.ecs.capability.logging-driver.json-file"  
    },  
    {  
      "value": "vpc-1234567890123467",  
      "name": "ecs.vpc-id"  
    },  
    {  
      "name": "ecs.capability.execution-role-awslogs"  
    },  
    {  
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"  
    },  
    {  
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"  
    },  
    {  
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"  
    },  
    {  
      "name": "ecs.capability.docker-plugin.local"  
    },  
    {  
      "name": "ecs.capability.task-eni"  
    },  
    {  
      "name": "ecs.capability.task-cpu-mem-limit"  
    },  
    {  
      "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"  
    },  
    {  
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"  
    },  
    {  
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"  
    },  
    {  
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"  
    },  
  ],  
}
```

```
{
  "name": "ecs.capability.execution-role-ecr-pull"
},
{
  "name": "ecs.capability.container-health-check"
},
{
  "value": "subnet-1234567890123467",
  "name": "ecs.subnet-id"
},
{
  "value": "us-west-2a",
  "name": "ecs.availability-zone"
},
{
  "value": "t2.micro",
  "name": "ecs.instance-type"
},
{
  "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
},
{
  "name": "ecs.capability.aws-appmesh"
},
{
  "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
},
{
  "name": "com.amazonaws.ecs.capability.privileged-container"
},
{
  "name": "ecs.capability.container-ordering"
```

```
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
  },
  {
    "value": "x86_64",
    "name": "ecs.cpu-architecture"
  },
  {
    "value": "93f43776-2018.10.0",
    "name": "ecs.capability.cni-plugin-version"
  },
  {
    "name": "ecs.capability.secrets.ssm.environment-variables"
  },
  {
    "name": "ecs.capability.pid-ipc-namespace-sharing"
  },
  {
    "name": "com.amazonaws.ecs.capability.ecr-auth"
  },
  {
    "value": "linux",
    "name": "ecs.os-type"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
  },
  {
    "name": "ecs.capability.task-eia"
  },
  {
    "name": "ecs.capability.private-registry-
authentication.secretsmanager"
  },
  },
```

```
    {
      "name": "com.amazonaws.ecs.capability.task-iam-role"
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
    }
  ],
  "pendingTasksCount": 0,
  "tags": [],
  "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "registeredResources": [
    {
      "integerValue": 1024,
      "doubleValue": 0.0,
      "type": "INTEGER",
      "longValue": 0,
      "name": "CPU"
    },
    {
      "integerValue": 985,
      "doubleValue": 0.0,
      "type": "INTEGER",
      "longValue": 0,
      "name": "MEMORY"
    },
    {
      "type": "STRINGSET",
      "integerValue": 0,
      "name": "PORTS",
      "stringSetValue": [
        "22",
        "2376",
        "2375",
        "51678",
        "51679"
      ],
      "longValue": 0,
      "doubleValue": 0.0
    },
    {
      "type": "STRINGSET",
      "integerValue": 0,
      "name": "PORTS_UDP",
```



```
        "stringSetValue": [],
        "longValue": 0,
        "doubleValue": 0.0
    }
],
"status": "INACTIVE",
"registeredAt": 1557768075.681,
"version": 4,
"versionInfo": {
    "agentVersion": "1.27.0",
    "agentHash": "aabe65ee",
    "dockerVersion": "DockerVersion: 18.06.1-ce"
},
"attachments": [],
"runningTasksCount": 0,
"ec2InstanceId": "i-12345678901234678"
}
}
```

詳細については、「[デベロッパーガイド](#)」の「[コンテナインスタンスの登録解除](#)」を参照してください。ECS

- API 詳細については、AWS CLI 「[コマンドリファレンスDeregisterContainerInstance](#)」の「」を参照してください。

deregister-task-definition

次のコード例は、deregister-task-definition を使用する方法を示しています。

AWS CLI

タスク定義の登録を解除するには

次のderegister-task-definition例では、デフォルトのリージョンでcurlerタスク定義の最初のリビジョンを登録解除します。

```
aws ecs deregister-task-definition --task-definition curler:1
```

結果の出力では、タスク定義ステータスに `INACTIVE` が表示されることに注意してください。

```
{
  "taskDefinition": {
```

```
    "status": "INACTIVE",
    "family": "curler",
    "volumes": [],
    "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-definition/
curler:1",
    "containerDefinitions": [
      {
        "environment": [],
        "name": "curler",
        "mountPoints": [],
        "image": "curl:latest",
        "cpu": 100,
        "portMappings": [],
        "entryPoint": [],
        "memory": 256,
        "command": [
          "curl -v http://example.com/"
        ],
        "essential": true,
        "volumesFrom": []
      }
    ],
    "revision": 1
  }
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS タスク定義」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterTaskDefinition](#)」の「」を参照してください。

describe-capacity-providers

次の例は、describe-capacity-providers を使用する方法を説明しています。

AWS CLI

例 1: すべてのキャパシティプロバイダーを記述するには

次のdescribe-capacity-providers例では、すべてのキャパシティープロバイダーの詳細を取得します。

aws ecs describe-capacity-providers

出力:

```
{
  "capacityProviders": [
    {
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/MyCapacityProvider",
      "name": "MyCapacityProvider",
      "status": "ACTIVE",
      "autoScalingGroupProvider": {
        "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
        "managedScaling": {
          "status": "ENABLED",
          "targetCapacity": 100,
          "minimumScalingStepSize": 1,
          "maximumScalingStepSize": 1000
        },
        "managedTerminationProtection": "ENABLED"
      },
      "tags": []
    },
    {
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE",
      "name": "FARGATE",
      "status": "ACTIVE",
      "tags": []
    },
    {
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-
provider/FARGATE_SPOT",
      "name": "FARGATE_SPOT",
      "status": "ACTIVE",
      "tags": []
    }
  ]
}
```

詳細については、「Amazon デベロッパーガイド」の [「クラスターキャパシティプロバイダー」](#) を参照してください。 ECS

例 2: 特定のキャパシティプロバイダーを記述するには

次のdescribe-capacity-providers例では、特定のキャパシティープロバイダーの詳細を取得します。--include TAGS パラメータを使用すると、キャパシティープロバイダーに関連付けられたタグが出力に追加されます。

```
aws ecs describe-capacity-providers \  
  --capacity-providers MyCapacityProvider \  
  --include TAGS
```

出力:

```
{  
  "capacityProviders": [  
    {  
      "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-provider/MyCapacityProvider",  
      "name": "MyCapacityProvider",  
      "status": "ACTIVE",  
      "autoScalingGroupProvider": {  
        "autoScalingGroupArn": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",  
        "managedScaling": {  
          "status": "ENABLED",  
          "targetCapacity": 100,  
          "minimumScalingStepSize": 1,  
          "maximumScalingStepSize": 1000  
        },  
        "managedTerminationProtection": "ENABLED"  
      },  
      "tags": [  
        {  
          "key": "environment",  
          "value": "production"  
        }  
      ]  
    }  
  ]  
}
```

```
}
```

詳細については、「Amazon デベロッパーガイド」の「[クラスターキャパシティプロバイダー](#)」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCapacityProviders](#)」の「」を参照してください。

describe-clusters

次の例は、describe-clusters を使用する方法を説明しています。

AWS CLI

例 1: クラスターを記述するには

次の describe-clusters の例は、指定されたクラスターの詳細情報を取得します。

```
aws ecs describe-clusters \  
  --cluster default
```

出力:

```
{  
  "clusters": [  
    {  
      "status": "ACTIVE",  
      "clusterName": "default",  
      "registeredContainerInstancesCount": 0,  
      "pendingTasksCount": 0,  
      "runningTasksCount": 0,  
      "activeServicesCount": 1,  
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default"  
    }  
  ],  
  "failures": []  
}
```

詳細については、「Amazon デベロッパーガイド」の「[Amazon ECS クラスター](#)」を参照してください。 ECS

例 2: 添付ファイルオプションを使用してクラスターを記述するには

次のdescribe-clusters例では、ATTACHMENTSオプションを指定します。指定されたクラスターの詳細情報と、クラスターにアタッチされているリソースのリストを添付ファイルの形式で取得します。クラスターでキャパシティープロバイダーを使用する場合、AutoScaling リソースはプランまたはスケーリングポリシーのいずれかで、asp または as_policy として表されます ATTACHMENTS。

```
aws ecs describe-clusters \  
  --include ATTACHMENTS \  
  --clusters sampleCluster
```

出力:

```
{  
  "clusters": [  
    {  
      "clusterArn": "arn:aws:ecs:af-south-1:123456789222:cluster/  
sampleCluster",  
      "clusterName": "sampleCluster",  
      "status": "ACTIVE",  
      "registeredContainerInstancesCount": 0,  
      "runningTasksCount": 0,  
      "pendingTasksCount": 0,  
      "activeServicesCount": 0,  
      "statistics": [],  
      "tags": [],  
      "settings": [],  
      "capacityProviders": [  
        "sampleCapacityProvider"  
      ],  
      "defaultCapacityProviderStrategy": [],  
      "attachments": [  
        {  
          "id": "a1b2c3d4-5678-901b-cdef-EXAMPLE22222",  
          "type": "as_policy",  
          "status": "CREATED",  
          "details": [  
            {  
              "name": "capacityProviderName",  
              "value": "sampleCapacityProvider"  
            },  
            {  
              "name": "scalingPolicyName",
```

```

        "value": "ECSManagedAutoScalingPolicy-3048e262-
fe39-4eaf-826d-6f975d303188"
      }
    ]
  },
  "attachmentsStatus": "UPDATE_COMPLETE"
}
],
"failures": []
}

```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS クラスター」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClusters](#)」の「」を参照してください。

describe-container-instances

次のコード例は、describe-container-instances を使用する方法を示しています。

AWS CLI

コンテナインスタンスを記述するには

次のdescribe-container-instances例では、コンテナインスタンスを識別子UUIDとして使用して、updateクラスター内のコンテナインスタンスの詳細を取得します。

```

aws ecs describe-container-instances \
  --cluster update \
  --container-instances a1b2c3d4-5678-90ab-cdef-11111EXAMPLE

```

出力:

```

{
  "failures": [],
  "containerInstances": [
    {
      "status": "ACTIVE",
      "registeredResources": [
        {
          "integerValue": 2048,

```

```
        "longValue": 0,
        "type": "INTEGER",
        "name": "CPU",
        "doubleValue": 0.0
    },
    {
        "integerValue": 3955,
        "longValue": 0,
        "type": "INTEGER",
        "name": "MEMORY",
        "doubleValue": 0.0
    },
    {
        "name": "PORTS",
        "longValue": 0,
        "doubleValue": 0.0,
        "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678"
        ],
        "type": "STRINGSET",
        "integerValue": 0
    }
],
"ec2InstanceId": "i-A1B2C3D4",
"agentConnected": true,
"containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
"pendingTasksCount": 0,
"remainingResources": [
    {
        "integerValue": 2048,
        "longValue": 0,
        "type": "INTEGER",
        "name": "CPU",
        "doubleValue": 0.0
    },
    {
        "integerValue": 3955,
        "longValue": 0,
        "type": "INTEGER",
        "name": "MEMORY",
```



```
        "doubleValue": 0.0
      },
      {
        "name": "PORTS",
        "longValue": 0,
        "doubleValue": 0.0,
        "stringSetValue": [
          "22",
          "2376",
          "2375",
          "51678"
        ],
        "type": "STRINGSET",
        "integerValue": 0
      }
    ],
    "runningTasksCount": 0,
    "versionInfo": {
      "agentVersion": "1.0.0",
      "agentHash": "4023248",
      "dockerVersion": "DockerVersion: 1.5.0"
    }
  }
]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECSコンテナインスタンス」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeContainerInstances](#)」の「」を参照してください。

describe-services

次の例は、describe-services を使用する方法を説明しています。

AWS CLI

サービスを記述するには

次のdescribe-services例では、デフォルトクラスター内のmy-http-serviceサービスの詳細を取得します。

```
aws ecs describe-services --services my-http-service
```

出力:

```
{
  "services": [
    {
      "status": "ACTIVE",
      "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
amazon-ecs-sample:1",
      "pendingCount": 0,
      "loadBalancers": [],
      "desiredCount": 10,
      "createdAt": 1466801808.595,
      "serviceName": "my-http-service",
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
      "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/my-http-
service",
      "deployments": [
        {
          "status": "PRIMARY",
          "pendingCount": 0,
          "createdAt": 1466801808.595,
          "desiredCount": 10,
          "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:1",
          "updatedAt": 1428326312.703,
          "id": "ecs-svc/1234567890123456789",
          "runningCount": 10
        }
      ],
      "events": [
        {
          "message": "(service my-http-service) has reached a steady
state.",
          "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
          "createdAt": 1466801812.435
        }
      ],
      "runningCount": 10
    }
  ],
  "failures": []
}
```

```
}
```

詳細については、「Amazon デベロッパーガイド」の「[サービス](#)」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeServices](#)」の「」を参照してください。

describe-task-definition

次のコード例は、describe-task-definition を使用する方法を示しています。

AWS CLI

タスク定義を記述するには

次のdescribe-task-definition例では、タスク定義の詳細を取得します。

```
aws ecs describe-task-definition \  
  --task-definition hello_world:8
```

出力:

```
{  
  "taskDefinition": {  
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/  
hello_world:8",  
    "containerDefinitions": [  
      {  
        "cpu": 10,  
        "environment": [],  
        "essential": true,  
        "image": "wordpress",  
        "links": [  
          "mysql"  
        ] ,  
        "memory": 500,  
        "mountPoints": [],  
        "name": "wordpress",  
        "portMappings": [  
          {  
            "containerPort": 80,  
            "hostPort": 80
```

```
    }
  ],
  "volumesFrom": []
},
{
  "cpu": 10,
  "environment": [
    {
      "name": "MYSQL_ROOT_PASSWORD",
      "value": "password"
    }
  ],
  "essential": true,
  "image": "mysql",
  "memory": 500,
  "mountPoints": [],
  "name": "mysql",
  "portMappings": [],
  "volumesFrom": []
}
],
"family": "hello_world",
"revision": 8,
"volumes": [],
"status": "ACTIVE",
"placementConstraints": [],
"compatibilities": [
  "EXTERNAL",
  "EC2"
],
"registeredAt": "2024-06-21T11:15:12.669000-05:00",
"registeredBy": "arn:aws:sts::012345678910:assumed-role/demo-role/jane-doe"
},
"tags": []
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS タスク定義」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTaskDefinition](#)」の「」を参照してください。

describe-task-sets

次のコード例は、describe-task-sets を使用する方法を示しています。

AWS CLI

タスクセットを記述するには

次のdescribe-task-sets例では、外部デプロイを使用するサービス内のタスクセットについて説明します。

```
aws ecs describe-task-sets \  
  --cluster MyCluster \  
  --service MyService \  
  --task-sets arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-  
svc/1234567890123456789
```

出力:

```
{  
  "taskSets": [  
    {  
      "id": "ecs-svc/1234567890123456789",  
      "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
      "status": "ACTIVE",  
      "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sample-fargate:2",  
      "computedDesiredCount": 0,  
      "pendingCount": 0,  
      "runningCount": 0,  
      "createdAt": 1557207715.195,  
      "updatedAt": 1557207740.014,  
      "launchType": "EC2",  
      "networkConfiguration": {  
        "awsvpcConfiguration": {  
          "subnets": [  
            "subnet-12344321"  
          ],  
          "securityGroups": [  
            "sg-1234431"  
          ],  
          "assignPublicIp": "DISABLED"  
        }  
      }  
    }  
  ]  
}
```

```
    }
  },
  "loadBalancers": [],
  "serviceRegistries": [],
  "scale": {
    "value": 0.0,
    "unit": "PERCENT"
  },
  "stabilityStatus": "STEADY_STATE",
  "stabilityStatusAt": 1557207740.014
}
],
"failures": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTaskSets](#)」の「」を参照してください。

describe-tasks

次の例は、describe-tasks を使用する方法を説明しています。

AWS CLI

例 1: 単一のタスクを記述するには

次の describe-tasks の例は、クラスター内の単一のタスクの詳細情報を取得します。タスクは、ID またはタスクARN全体を使用して指定できます。この例では、タスクARNの完全な を使用します。

```
aws ecs describe-tasks \
  --cluster MyCluster \
  --tasks arn:aws:ecs:us-east-1:123456789012:task/MyCluster/4d590253bb114126b7afa7b58EXAMPLE
```

出力:

```
{
  "tasks": [
    {
      "attachments": [],
      "attributes": [
```

```
        {
            "name": "ecs.cpu-architecture",
            "value": "x86_64"
        }
    ],
    "availabilityZone": "us-east-1b",
    "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
    "connectivity": "CONNECTED",
    "connectivityAt": "2021-08-11T12:21:26.681000-04:00",
    "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-
instance/test/025c7e2c5e054a6790a29fc1fEXAMPLE",
    "containers": [
        {
            "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/4d590253bb114126b7afa7b58eea9221/a992d1cc-ea46-474a-b6e8-24688EXAMPLE",
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58EXAMPLE",
            "name": "simple-app",
            "image": "httpd:2.4",
            "runtimeId":
"91251eed27db90006ad67b1a08187290869f216557717dd5c39b37c94EXAMPLE",
            "lastStatus": "RUNNING",
            "networkBindings": [
                {
                    "bindIP": "0.0.0.0",
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "networkInterfaces": [],
            "healthStatus": "UNKNOWN",
            "cpu": "10",
            "memory": "300"
        }
    ],
    "cpu": "10",
    "createdAt": "2021-08-11T12:21:26.681000-04:00",
    "desiredStatus": "RUNNING",
    "enableExecuteCommand": false,
    "group": "service:testupdate",
    "healthStatus": "UNKNOWN",
    "lastStatus": "RUNNING",
    "launchType": "EC2",
```

```

    "memory": "300",
    "overrides": {
      "containerOverrides": [
        {
          "name": "simple-app"
        }
      ],
      "inferenceAcceleratorOverrides": []
    },
    "pullStartedAt": "2021-08-11T12:21:28.234000-04:00",
    "pullStoppedAt": "2021-08-11T12:21:33.793000-04:00",
    "startedAt": "2021-08-11T12:21:34.945000-04:00",
    "startedBy": "ecs-svc/968695068243EXAMPLE",
    "tags": [],
    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/4d590253bb114126b7afa7b58eea9221",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/console-sample-app-static2:1",
    "version": 2
  }
],
"failures": []
}

```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS タスク定義」](#)を参照してください。 ECS

例 2: 複数のタスクを記述するには

次の describe-tasks の例は、クラスター内の複数のタスクの詳細情報を取得します。タスクは、ID またはタスクARN全体を使用して指定できます。この例では、タスクIDsをすべて使用します。

```

aws ecs describe-tasks \
  --cluster MyCluster \
  --tasks "74de0355a10a4f979ac495c14EXAMPLE" "d789e94343414c25b9f6bd59eEXAMPLE"

```

出力:

```

{
  "tasks": [
    {
      "attachments": [

```



```
{
  "id": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
  "type": "ElasticNetworkInterface",
  "status": "ATTACHED",
  "details": [
    {
      "name": "subnetId",
      "value": "subnet-0d0eab1bb3EXAMPLE"
    },
    {
      "name": "networkInterfaceId",
      "value": "eni-0fa40520aeEXAMPLE"
    },
    {
      "name": "macAddress",
      "value": "0e:89:76:28:07:b3"
    },
    {
      "name": "privateDnsName",
      "value": "ip-10-0-1-184.ec2.internal"
    },
    {
      "name": "privateIPv4Address",
      "value": "10.0.1.184"
    }
  ]
},
"attributes": [
  {
    "name": "ecs.cpu-architecture",
    "value": "x86_64"
  }
],
"availabilityZone": "us-east-1b",
"clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
"connectivity": "CONNECTED",
"connectivityAt": "2021-12-20T12:13:37.875000-05:00",
"containers": [
  {
    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/MyCluster/74de0355a10a4f979ac495c14EXAMPLE/aad3ba00-83b3-4dac-84d4-11f8cEXAMPLE",
    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
```

```
        "name": "web",
        "image": "nginx",
        "runtimeId": "74de0355a10a4f979ac495c14EXAMPLE-265927825",
        "lastStatus": "RUNNING",
        "networkBindings": [],
        "networkInterfaces": [
            {
                "attachmentId": "d9e7735a-16aa-4128-bc7a-b2d51EXAMPLE",
                "privateIpv4Address": "10.0.1.184"
            }
        ],
        "healthStatus": "UNKNOWN",
        "cpu": "99",
        "memory": "100"
    }
],
"cpu": "256",
"createdAt": "2021-12-20T12:13:20.226000-05:00",
"desiredStatus": "RUNNING",
"enableExecuteCommand": false,
"group": "service:tdsevicetag",
"healthStatus": "UNKNOWN",
"lastStatus": "RUNNING",
"launchType": "FARGATE",
"memory": "512",
"overrides": {
    "containerOverrides": [
        {
            "name": "web"
        }
    ],
    "inferenceAcceleratorOverrides": []
},
"platformVersion": "1.4.0",
"platformFamily": "Linux",
"pullStartedAt": "2021-12-20T12:13:42.665000-05:00",
"pullStoppedAt": "2021-12-20T12:13:46.543000-05:00",
"startedAt": "2021-12-20T12:13:48.086000-05:00",
"startedBy": "ecs-svc/988401040018EXAMPLE",
"tags": [],
"taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/74de0355a10a4f979ac495c14EXAMPLE",
"taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/webserver:2",
```

```
    "version": 3,
    "ephemeralStorage": {
      "sizeInGiB": 20
    }
  },
  {
    "attachments": [
      {
        "id": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
        "type": "ElasticNetworkInterface",
        "status": "ATTACHED",
        "details": [
          {
            "name": "subnetId",
            "value": "subnet-0d0eab1bb3EXAMPLE"
          },
          {
            "name": "networkInterfaceId",
            "value": "eni-064c7766daEXAMPLE"
          },
          {
            "name": "macAddress",
            "value": "0e:76:83:01:17:a9"
          },
          {
            "name": "privateDnsName",
            "value": "ip-10-0-1-41.ec2.internal"
          },
          {
            "name": "privateIPv4Address",
            "value": "10.0.1.41"
          }
        ]
      }
    ],
    "attributes": [
      {
        "name": "ecs.cpu-architecture",
        "value": "x86_64"
      }
    ],
    "availabilityZone": "us-east-1b",
    "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
    "connectivity": "CONNECTED",
```

```
"connectivityAt": "2021-12-20T12:13:35.243000-05:00",
"containers": [
  {
    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
MyCluster/d789e94343414c25b9f6bd59eEXAMPLE/9afef792-609b-43a5-bb6a-3efdbEXAMPLE",
    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
    "name": "web",
    "image": "nginx",
    "runtimeId": "d789e94343414c25b9f6bd59eEXAMPLE-265927825",
    "lastStatus": "RUNNING",
    "networkBindings": [],
    "networkInterfaces": [
      {
        "attachmentId": "214eb5a9-45cd-4bf8-87bc-57fefEXAMPLE",
        "privateIpv4Address": "10.0.1.41"
      }
    ],
    "healthStatus": "UNKNOWN",
    "cpu": "99",
    "memory": "100"
  }
],
"cpu": "256",
"createdAt": "2021-12-20T12:13:20.226000-05:00",
"desiredStatus": "RUNNING",
"enableExecuteCommand": false,
"group": "service:tdsevicetag",
"healthStatus": "UNKNOWN",
"lastStatus": "RUNNING",
"launchType": "FARGATE",
"memory": "512",
"overrides": {
  "containerOverrides": [
    {
      "name": "web"
    }
  ],
  "inferenceAcceleratorOverrides": []
},
"platformVersion": "1.4.0",
"platformFamily": "Linux",
"pullStartedAt": "2021-12-20T12:13:44.611000-05:00",
"pullStoppedAt": "2021-12-20T12:13:48.251000-05:00",
```

```
    "startedAt": "2021-12-20T12:13:49.326000-05:00",
    "startedBy": "ecs-svc/988401040018EXAMPLE",
    "tags": [],
    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/webserver:2",
    "version": 3,
    "ephemeralStorage": {
      "sizeInGiB": 20
    }
  },
  "failures": []
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS タスク定義」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTasks](#)」の「」を参照してください。

execute-command

次の例は、execute-command を使用する方法を説明しています。

AWS CLI

interactive /bin/sh コマンドを実行するには

次のexecute-command例では、ID が のタスク MyContainer に という名前のコンテナに対して interactive /bin/sh コマンドを実行しますarn:aws:ecs:us-east-1:123456789012:task/MyCluster/d789e94343414c25b9f6bd59eEXAMPLE。

```
aws ecs execute-command \
  --cluster MyCluster \
  --task arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
d789e94343414c25b9f6bd59eEXAMPLE \
  --container MyContainer \
  --interactive \
  --command "/bin/sh"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon デベロッパーガイド」の「デバッグに Amazon ECS Exec を使用する」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ExecuteCommand](#)」の「」を参照してください。

list-account-settings

次の例は、list-account-settings を使用する方法を説明しています。

AWS CLI

例 1: アカウントのアカウント設定を表示するには

次のlist-account-settings例は、アカウントの有効なアカウント設定を示しています。

```
aws ecs list-account-settings --effective-settings
```

出力:

```
{
  "settings": [
    {
      "name": "containerInstanceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    },
    {
      "name": "serviceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    },
    {
      "name": "taskLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    }
  ]
}
```

例 2: 特定のIAMユーザーまたはIAMロールのアカウント設定を表示するには

次のlist-account-settings例では、指定されたIAMユーザーまたはIAMロールのアカウント設定を表示します。

```
aws ecs list-account-settings --principal-arn arn:aws:iam::123456789012:user/MyUser
```

出力:

```
{
  "settings": [
    {
      "name": "serviceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
  ]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon リソースネーム \(ARNs \)」およびIDs](#)「」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccountSettings](#)」の「」を参照してください。

list-attributes

次の例は、list-attributes を使用する方法を説明しています。

AWS CLI

特定の属性を含むコンテナインスタンスを一覧表示するには

次の例では、デフォルトクラスターに 属性を持つコンテナインスタンスのstack=production属性を一覧表示します。

```
aws ecs list-attributes \
  --target-type container-instance \
  --attribute-name stack \
  --attribute-value production \
```

```
--cluster default
```

出力:

```
{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS コンテナエージェント設定」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAttributes](#)」の「」を参照してください。

list-clusters

次のコード例は、list-clusters を使用する方法を示しています。

AWS CLI

使用可能なクラスターを一覧表示するには

次の list-clusters の例は、使用可能なすべてのクラスターを一覧表示します。

```
aws ecs list-clusters
```

出力:

```
{
  "clusterArns": [
    "arn:aws:ecs:us-west-2:123456789012:cluster/MyECSCluster1",
    "arn:aws:ecs:us-west-2:123456789012:cluster/AnotherECSCluster"
  ]
}
```



```
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS クラスター」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListClusters](#)」の「」を参照してください。

list-container-instances

次の例は、list-container-instances を使用する方法を説明しています。

AWS CLI

クラスター内のコンテナインスタンスを一覧表示するには

次のlist-container-instances例では、クラスター内で使用可能なすべてのコンテナインスタンスを一覧表示します。

```
aws ecs list-container-instances --cluster MyCluster
```

出力:

```
{
  "containerInstanceArns": [
    "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "arn:aws:ecs:us-west-2:123456789012:container-instance/MyCluster/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
  ]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECSコンテナインスタンス」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListContainerInstances](#)」の「」を参照してください。

list-services-by-namespace

次の例は、list-services-by-namespace を使用する方法を説明しています。

AWS CLI

名前空間内のサービスを一覧表示するには

次の `list-services-by-namespace` 例では、デフォルトのリージョンで指定された名前空間用に設定されたすべてのサービスを一覧表示します。

```
aws ecs list-services-by-namespace \  
  --namespace service-connect
```

出力:

```
{  
  "serviceArns": [  
    "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",  
    "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-  
service"  
  ]  
}
```

詳細については、「Amazon デベロッパーガイド」の「[Service Connect](#)」を参照してください。
ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServicesByNamespace](#)」の「」を参照してください。

list-services

次のコード例は、`list-services` を使用する方法を示しています。

AWS CLI

クラスター内のサービスを一覧表示するには

次の `list-services` の例は、クラスターで実行中のサービスを一覧表示する方法を示しています。

```
aws ecs list-services --cluster MyCluster
```

出力:

```
{
  "serviceArns": [
    "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService"
  ]
}
```

詳細については、「Amazon デベロッパーガイド」の「[サービス](#)」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServices](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、特定のクラスターのタグを一覧表示します。

```
aws ecs list-tags-for-resource \
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
```

出力:

```
{
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value2"
    },
    {
      "key": "key3",
      "value": "value3"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-task-definition-families

次の例は、list-task-definition-families を使用する方法を説明しています。

AWS CLI

例 1: 登録されたタスク定義ファミリーを一覧表示するには

次のlist-task-definition-families例では、登録されたすべてのタスク定義ファミリーを一覧表示します。

```
aws ecs list-task-definition-families
```

出力:

```
{
  "families": [
    "node-js-app",
    "web-timer",
    "hpcc",
    "hpcc-c4-8xlarge"
  ]
}
```

例 2: 登録されたタスク定義ファミリーをフィルタリングするには

次のlist-task-definition-families例では、「hpcc」で始まるタスク定義のリビジョンを一覧表示します。

```
aws ecs list-task-definition-families --family-prefix hpcc
```

出力:

```
{
  "families": [
    "hpcc",
    "hpcc-c4-8xlarge"
  ]
}
```

```
}
```

詳細については、「Amazon デベロッパーガイド」の「[タスク定義パラメータ](#)」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTaskDefinitionFamilies](#)」の「」を参照してください。

list-task-definitions

次のコード例は、list-task-definitions を使用する方法を示しています。

AWS CLI

例 1: 登録されたタスク定義を一覧表示するには

次のlist-task-definitions例では、登録されたすべてのタスク定義を一覧表示します。

```
aws ecs list-task-definitions
```

出力:

```
{
  "taskDefinitionArns": [
    "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep300:2",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/sleep360:1",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
  ]
}
```

例 2: ファミリーに登録されたタスク定義を一覧表示するには

次の list-task-definitions例では、指定されたファミリーのタスク定義リビジョンを一覧表示します。

```
aws ecs list-task-definitions --family-prefix wordpress
```

出力:

```
{
  "taskDefinitionArns": [
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:3",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:4",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:5",
    "arn:aws:ecs:us-west-2:123456789012:task-definition/wordpress:6"
  ]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS タスク定義」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTaskDefinitions](#)」の「」を参照してください。

list-tasks

次の例は、list-tasks を使用する方法を説明しています。

AWS CLI

例 1: クラスター内のタスクを一覧表示するには

次の list-tasks の例は、クラスター内のすべてのタスクを一覧表示します。

```
aws ecs list-tasks --cluster default
```

出力:

```
{
  "taskArns": [
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
  ]
}
```

例 2: 特定のコンテナインスタンスのタスクを一覧表示するには

次のlist-tasks例では、コンテナインスタンスをフィルターUUIDとして使用して、コンテナインスタンスのタスクを一覧表示します。

```
aws ecs list-tasks --cluster default --container-instance a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

出力:

```
{
  "taskArns": [
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
  ]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS タスク定義」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTasks](#)」の「」を参照してください。

put-account-setting-default

次のコード例は、put-account-setting-default を使用する方法を示しています。

AWS CLI

デフォルトのアカウント設定を変更するには

次のput-account-setting-default例では、アカウントのすべてのIAMユーザーまたはロールのデフォルトのアカウント設定を変更します。これらの変更は、IAMユーザーまたはロールがこれらの設定を明示的に上書きしない限り、AWS アカウント全体に適用されます。

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled
```

出力:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:root"
  }
}
```

```
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon リソースネーム \(ARNs \) およびIDs」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「 コマンドリファレンス[PutAccountSettingDefault](#)」の「」を参照してください。

put-account-setting

次の例は、put-account-setting を使用する方法を説明しています。

AWS CLI

IAMユーザーアカウントのアカウント設定を変更するには

次のput-account-setting例では、IAMユーザーアカウントのアカウントserviceLongArnFormat設定を有効にします。

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled
```

出力:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::130757420319:user/your_username"
  }
}
```

詳細については、「Amazon ECSデベロッパーガイド」の[「アカウント設定の変更」](#)を参照してください。

- API 詳細については、AWS CLI 「 コマンドリファレンス[PutAccountSetting](#)」の「」を参照してください。

put-account-settings

次の例は、put-account-settings を使用する方法を説明しています。

AWS CLI

IAM ユーザーまたはIAMロールのアカウント設定を変更するには

次のput-account-setting例では、指定されたIAMユーザーまたはIAMロールのアカウント設定を変更します。

```
aws ecs put-account-setting \  
  --name serviceLongArnFormat \  
  --value enabled \  
  --principal-arn arn:aws:iam::123456789012:user/MyUser
```

出力:

```
{  
  "setting": {  
    "name": "serviceLongArnFormat",  
    "value": "enabled",  
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutAccountSettings](#)」の「」を参照してください。

put-attributes

次のコード例は、put-attributes を使用する方法を示しています。

AWS CLI

属性を作成して Amazon ECS リソースに関連付けるには

以下では、名前スタックと値の本番稼働を含む属性をコンテナインスタンスput-attributesに適用します。

```
aws ecs put-attributes \  
  --attributes name=stack,value=production,targetId=arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

出力:

```
{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutAttributes](#)」の「」を参照してください。

put-cluster-capacity-providers

次の例は、put-cluster-capacity-providers を使用する方法を説明しています。

AWS CLI

例 1: 既存のキャパシティープロバイダーをクラスターに追加するには

次のput-cluster-capacity-providers例では、既存のキャパシティープロバイダーをクラスターに追加します。キャパシティープロバイダーを作成するには、create-capacity-provider コマンドを使用します。describe-clusters コマンドは、クラスターに関連付けられた現在のキャパシティープロバイダーとデフォルトのキャパシティープロバイダー戦略を記述するために使用されます。新しいキャパシティープロバイダーをクラスターに追加するときは、クラスターに関連付ける新しいキャパシティープロバイダーに加えて、既存のすべてのキャパシティープロバイダーを指定する必要があります。また、クラスターに関連付けるデフォルトのキャパシティープロバイダー戦略を指定する必要があります。この例では、MyClusterクラスターにMyCapacityProvider1キャパシティープロバイダーが関連付けられているため、両方のキャパシティープロバイダーにタスクが均等に分散されるように、MyCapacityProvider2キャパシティープロバイダーを追加してデフォルトのキャパシティープロバイダー戦略に含めます。

```
aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
  --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

出力:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ],
    "capacityProviders": [
      "MyCapacityProvider1",
      "MyCapacityProvider2"
    ],
    "defaultCapacityProviderStrategy": [
      {
        "capacityProvider": "MyCapacityProvider1",
        "weight": 1,
        "base": 0
      },
      {
        "capacityProvider": "MyCapacityProvider2",
        "weight": 1,
        "base": 0
      }
    ],
    "attachments": [
      {
        "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
        "type": "as_policy",
        "status": "ACTIVE",
        "details": [
          {
            "name": "capacityProviderName",
            "value": "MyCapacityProvider1"
          }
        ]
      }
    ]
  }
}
```

```

        },
        {
            "name": "scalingPolicyName",
            "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE111111"
        }
    ]
},
{
    "id": "ae592060-2382-4663-9476-b015c685593c",
    "type": "as_policy",
    "status": "ACTIVE",
    "details": [
        {
            "name": "capacityProviderName",
            "value": "MyCapacityProvider2"
        },
        {
            "name": "scalingPolicyName",
            "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE222222"
        }
    ]
}
],
"attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}

```

詳細については、「Amazon デベロッパーガイド」の [「クラスターキャパシティープロバイダー」](#) を参照してください。 ECS

例 2: クラスターからキャパシティープロバイダーを削除するには

次のput-cluster-capacity-providers例では、クラスターからキャパシティープロバイダーを削除します。describe-clusters コマンドは、クラスターに関連付けられている現在のキャパシティープロバイダーを記述するために使用されます。クラスターからキャパシティープロバイダーを削除するときは、クラスターに関連付けるキャパシティープロバイダーと、クラスターに関連付けるデフォルトのキャパシティープロバイダー戦略を指定する必要があります。この例では、クラスターに MyCapacityProvider1と MyCapacityProvider2 キャパシティープロバイダーが関連付けられているため、MyCapacityProvider2キャパシティープロバイダーを削除

するため、コマンド `MyCapacityProvider1` のみを指定し、デフォルトのキャパシティープロバイダー戦略を更新します。

```
aws ecs put-cluster-capacity-providers \  
  --cluster MyCluster \  
  --capacity-providers MyCapacityProvider1 \  
  --default-capacity-provider-  
strategy capacityProvider=MyCapacityProvider1,weight=1,base=0
```

出力:

```
{  
  "cluster": {  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "clusterName": "MyCluster",  
    "status": "ACTIVE",  
    "registeredContainerInstancesCount": 0,  
    "runningTasksCount": 0,  
    "pendingTasksCount": 0,  
    "activeServicesCount": 0,  
    "statistics": [],  
    "tags": [],  
    "settings": [  
      {  
        "name": "containerInsights",  
        "value": "enabled"  
      }  
    ],  
    "capacityProviders": [  
      "MyCapacityProvider1"  
    ],  
    "defaultCapacityProviderStrategy": [  
      "capacityProvider": "MyCapacityProvider1",  
      "weight": 1,  
      "base": 0  
    ],  
    "attachments": [  
      {  
        "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",  
        "type": "as_policy",  
        "status": "ACTIVE",  
        "details": [  
          {
```

```

        "name": "capacityProviderName",
        "value": "MyCapacityProvider1"
      },
      {
        "name": "scalingPolicyName",
        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE111111"
      }
    ]
  },
  {
    "id": "ae592060-2382-4663-9476-b015c685593c",
    "type": "as_policy",
    "status": "DELETING",
    "details": [
      {
        "name": "capacityProviderName",
        "value": "MyCapacityProvider2"
      },
      {
        "name": "scalingPolicyName",
        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE222222"
      }
    ]
  }
],
"attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}

```

詳細については、「Amazon デベロッパーガイド」の [「クラスターキャパシティプロバイダー」](#) を参照してください。 ECS

例 3: クラスターからすべてのキャパシティプロバイダーを削除するには

次のput-cluster-capacity-providers例では、クラスターから既存のキャパシティプロバイダーをすべて削除します。

```

aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers [] \
  --default-capacity-provider-strategy []

```

出力:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ],
    "capacityProviders": [],
    "defaultCapacityProviderStrategy": [],
    "attachments": [
      {
        "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
        "type": "as_policy",
        "status": "DELETING",
        "details": [
          {
            "name": "capacityProviderName",
            "value": "MyCapacityProvider1"
          },
          {
            "name": "scalingPolicyName",
            "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
          }
        ]
      },
      {
        "id": "ae592060-2382-4663-9476-b015c685593c",
        "type": "as_policy",
        "status": "DELETING",
        "details": [
          {
```

```

        "name": "capacityProviderName",
        "value": "MyCapacityProvider2"
      },
      {
        "name": "scalingPolicyName",
        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
      }
    ]
  },
  "attachmentsStatus": "UPDATE_IN_PROGRESS"
}

```

詳細については、「Amazon デベロッパーガイド」の「[クラスターキャパシティプロバイダー](#)」を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[PutClusterCapacityProviders](#)」の「」を参照してください。

register-task-definition

次の例は、register-task-definition を使用する方法を説明しています。

AWS CLI

例 1: タスク定義を JSON ファイルに登録するには

次のregister-task-definition例では、指定されたファミリーにタスク定義を登録します。コンテナ定義は、指定されたファイル場所にJSON形式で保存されます。

```

aws ecs register-task-definition \
  --cli-input-json file://<path_to_json_file>/sleep360.json

```

sleep360.json の内容:

```

{
  "containerDefinitions": [
    {
      "name": "sleep",
      "image": "busybox",

```



```
        "cpu": 10,
        "command": [
            "sleep",
            "360"
        ],
        "memory": 10,
        "essential": true
    }
],
"family": "sleep360"
}
```

出力:

```
{
  "taskDefinition": {
    "status": "ACTIVE",
    "family": "sleep360",
    "placementConstraints": [],
    "compatibilities": [
      "EXTERNAL",
      "EC2"
    ],
    "volumes": [],
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/sleep360:1",
    "containerDefinitions": [
      {
        "environment": [],
        "name": "sleep",
        "mountPoints": [],
        "image": "busybox",
        "cpu": 10,
        "portMappings": [],
        "command": [
          "sleep",
          "360"
        ],
        "memory": 10,
        "essential": true,
        "volumesFrom": []
      }
    ],
  },
}
```

```
    "revision": 1
  }
}
```

詳細については、「Amazon デベロッパーガイド」の [「タスク定義の例」](#) を参照してください。
ECS

例 2: タスク定義をJSON文字列パラメータに登録するには

次のregister-task-definition例では、エスケープされた二重引用符を持つJSON文字列パラメータとして提供されるコンテナ定義を使用してタスク定義を登録します。

```
aws ecs register-task-definition \  
  --family sleep360 \  
  --container-definitions "[{\\"name\\":\\"sleep\\",\\"image\\":\\"busybox\\",\\"cpu\\":10,\  
  \\"command\\":[\\"sleep\\",\\"360\\"],\\"memory\\":10,\\"essential\\":true}]"
```

出力は前の例と同じです。

詳細については、「Amazon デベロッパーガイド」の [「タスク定義の作成」](#) を参照してください。
ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterTaskDefinition](#)」の「」を参照してください。

run-task

次の例は、run-task を使用する方法を説明しています。

AWS CLI

デフォルトクラスターでタスクを実行するには

次のrun-task例では、デフォルトのクラスターでタスクを実行し、クライアントトークンを使用します。

```
aws ecs run-task \  
  --cluster default \  
  --task-definition sleep360:1 \  
  --client-token 550e8400-e29b-41d4-a716-446655440000
```

出力:

```
{
  "tasks": [
    {
      "attachments": [],
      "attributes": [
        {
          "name": "ecs.cpu-architecture",
          "value": "x86_64"
        }
      ],
      "availabilityZone": "us-east-1b",
      "capacityProviderName": "example-capacity-provider",
      "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/default",
      "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-
instance/default/bc4d2ec611d04bb7bb97e83ceEXAMPLE",
      "containers": [
        {
          "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/
default/d6f51cc5bbc94a47969c92035e9f66f8/75853d2d-711e-458a-8362-0f0aEXAMPLE",
          "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
          "name": "sleep",
          "image": "busybox",
          "lastStatus": "PENDING",
          "networkInterfaces": [],
          "cpu": "10",
          "memory": "10"
        }
      ],
      "cpu": "10",
      "createdAt": "2023-11-21T16:59:34.403000-05:00",
      "desiredStatus": "RUNNING",
      "enableExecuteCommand": false,
      "group": "family:sleep360",
      "lastStatus": "PENDING",
      "launchType": "EC2",
      "memory": "10",
      "overrides": {
        "containerOverrides": [
          {
            "name": "sleep"
          }
        ]
      }
    }
  ]
}
```

```
    ],
    "inferenceAcceleratorOverrides": [],
  },
  "tags": [],
  "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/
d6f51cc5bbc94a47969c9203EXAMPLE",
  "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-
definition/sleep360:1",
  "version": 1
}
],
"failures": []
}
```

詳細については、「Amazon ECSデベロッパーガイド」の「[タスクの実行](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RunTask](#)」の「」を参照してください。

start-task

次の例は、start-task を使用する方法を説明しています。

AWS CLI

新しいタスクを開始するには

以下では、デフォルトクラスター内の指定されたコンテナインスタンスで、タスク定義の最新リビジョンを使用してsleep360タスクstart-taskを開始します。

```
aws ecs start-task \
  --task-definition sleep360 \
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

出力:

```
{
  "tasks": [
    {
      "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
```

```
    "clusterArn": "arn:aws:ecs:us-west-2:130757420319:cluster/default",
    "taskDefinitionArn": "arn:aws:ecs:us-west-2:130757420319:task-
definition/sleep360:3",
    "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
    "overrides": {
      "containerOverrides": [
        {
          "name": "sleep"
        }
      ]
    },
    "lastStatus": "PENDING",
    "desiredStatus": "RUNNING",
    "cpu": "128",
    "memory": "128",
    "containers": [
      {
        "containerArn": "arn:aws:ecs:us-
west-2:130757420319:container/75f11ed4-8a3d-4f26-a33b-ad1db9e02d41",
        "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
        "name": "sleep",
        "lastStatus": "PENDING",
        "networkInterfaces": [],
        "cpu": "10",
        "memory": "10"
      }
    ],
    "version": 1,
    "createdAt": 1563421494.186,
    "group": "family:sleep360",
    "launchType": "EC2",
    "attachments": [],
    "tags": []
  }
],
"failures": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartTask](#)」の「」を参照してください。

stop-task

次のコード例は、stop-task を使用する方法を示しています。

AWS CLI

タスクを停止するには

以下stop-taskでは、指定されたタスクがデフォルトのクラスターで実行されないようにします。

```
aws ecs stop-task \  
--task 666fdccc2e2d4b6894dd422f4eeee8f8
```

出力:

```
{  
  "task": {  
    "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/  
default/666fdccc2e2d4b6894dd422f4eeee8f8",  
    "clusterArn": "arn:aws:ecs:us-west-2:130757420319:cluster/default",  
    "taskDefinitionArn": "arn:aws:ecs:us-west-2:130757420319:task-definition/  
sleep360:3",  
    "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/default/765936fadbdd46b5991a4bd70c2a43d4",  
    "overrides": {  
      "containerOverrides": []  
    },  
    "lastStatus": "STOPPED",  
    "desiredStatus": "STOPPED",  
    "cpu": "128",  
    "memory": "128",  
    "containers": [],  
    "version": 2,  
    "stoppedReason": "Taskfailedtostart",  
    "stopCode": "TaskFailedToStart",  
    "connectivity": "CONNECTED",  
    "connectivityAt": 1563421494.186,  
    "pullStartedAt": 1563421494.252,  
    "pullStoppedAt": 1563421496.252,  
    "executionStoppedAt": 1563421497,  
    "createdAt": 1563421494.186,  
    "stoppingAt": 1563421497.252,  
  }  
}
```

```
    "stoppedAt": 1563421497.252,  
    "group": "family:sleep360",  
    "launchType": "EC2",  
    "attachments": [],  
    "tags": []  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StopTask](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、指定されたリソースに 1 つのタグを追加します。

```
aws ecs tag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tags key=key1,value=value1
```

このコマンドでは何も出力されません。

リソースに複数のタグを追加するには

次のtag-resource例では、指定されたリソースに複数のタグを追加します。

```
aws ecs tag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、`untag-resource` を使用する方法を示しています。

AWS CLI

リソースからタグを削除するには

次の`untag-resource`例では、指定されたリソースからリストされているタグを削除します。

```
aws ecs untag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tag-keys key1,key2
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-cluster-settings

次の例は、`update-cluster-settings` を使用する方法を説明しています。

AWS CLI

クラスターの設定を変更するには

次の`update-cluster-settings`例では、defaultクラスターの CloudWatch Container Insights を有効にします。

```
aws ecs update-cluster-settings \  
  --cluster default \  
  --settings name=containerInsights,value=enabled
```

出力:

```
{  
  "cluster": {  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "clusterName": "default",  
    "status": "ACTIVE",  
    "registeredContainerInstancesCount": 0,  
    "runningTasksCount": 0,  
  }  
}
```



```
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ]
  }
}
```

詳細については、「Amazon ECSデベロッパーガイド」の「[アカウント設定の変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateClusterSettings](#)」の「」を参照してください。

update-container-agent

次のコード例は、update-container-agent を使用する方法を示しています。

AWS CLI

Amazon コンテナインスタンスでECSコンテナエージェントを更新するには

次のupdate-container-agent例では、デフォルトのクラスターで指定されたコンテナインスタンスのコンテナエージェントを更新します。

```
aws ecs update-container-agent --cluster default --container-  
instance a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
```

出力:

```
{
  "containerInstance": {
    "status": "ACTIVE",
    ...
    "agentUpdateStatus": "PENDING",
    "versionInfo": {
      "agentVersion": "1.0.0",
      "agentHash": "4023248",
    }
  }
}
```

```
        "dockerVersion": "DockerVersion: 1.5.0"
      }
    }
  }
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon ECS コンテナエージェントの更新」](#)を参照してください。 ECS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateContainerAgent](#)」の「」を参照してください。

update-container-instances-state

次のコード例は、update-container-instances-state を使用する方法を示しています。

AWS CLI

コンテナインスタンスの状態を更新するには

以下に、 が登録DRAINING先のクラスターから削除する、指定されたコンテナインスタンスの状態update-container-instances-stateを更新します。

```
aws ecs update-container-instances-state \
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4 \
  --status DRAINING
```

出力:

```
{
  "containerInstances": [
    {
      "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
      "ec2InstanceId": "i-013d87ffbb4d513bf",
      "version": 4390,
      "versionInfo": {
        "agentVersion": "1.29.0",
        "agentHash": "a190a73f",
        "dockerVersion": "DockerVersion:18.06.1-ce"
      },
      "remainingResources": [
        {
          "name": "CPU",
```

```
        "type": "INTEGER",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 1536
    },
    {
        "name": "MEMORY",
        "type": "INTEGER",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 2681
    },
    {
        "name": "PORTS",
        "type": "STRINGSET",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 0,
        "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678",
            "51679"
        ]
    },
    {
        "name": "PORTS_UDP",
        "type": "STRINGSET",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 0,
        "stringSetValue": []
    }
],
"registeredResources": [
    {
        "name": "CPU",
        "type": "INTEGER",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 2048
    },
    {
```

```
        "name": "MEMORY",
        "type": "INTEGER",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 3705
    },
    {
        "name": "PORTS",
        "type": "STRINGSET",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 0,
        "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678",
            "51679"
        ]
    },
    {
        "name": "PORTS_UDP",
        "type": "STRINGSET",
        "doubleValue": 0,
        "longValue": 0,
        "integerValue": 0,
        "stringSetValue": []
    }
],
"status": "DRAINING",
"agentConnected": true,
"runningTasksCount": 2,
"pendingTasksCount": 0,
"attributes": [
    {
        "name": "ecs.capability.secrets.asm.environment-variables"
    },
    {
        "name": "ecs.capability.branch-cni-plugin-version",
        "value": "e0703516-"
    },
    {
        "name": "ecs.ami-id",
        "value": "ami-00e0090ac21971297"
    }
]
```

```
    },
    {
      "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
    },
    {
      "name": "com.amazonaws.ecs.capability.logging-driver.none"
    },
    {
      "name": "ecs.capability.ecr-endpoint"
    },
    {
      "name": "ecs.capability.docker-plugin.local"
    },
    {
      "name": "ecs.capability.task-cpu-mem-limit"
    },
    {
      "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
    },
    {
      "name": "ecs.availability-zone",
      "value": "us-west-2c"
    },
    {
      "name": "ecs.capability.aws-appmesh"
    },
    {
      "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
    },
    {
      "name": "ecs.capability.task-eni-trunking"
    },
  ],
```

```
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
},
{
  "name": "com.amazonaws.ecs.capability.privileged-container"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
},
{
  "name": "ecs.cpu-architecture",
  "value": "x86_64"
},
{
  "name": "com.amazonaws.ecs.capability.ecr-auth"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
},
{
  "name": "ecs.os-type",
  "value": "linux"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
},
{
  "name": "ecs.capability.task-eia"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
},
},
```

```
{
  "name": "ecs.capability.private-registry-
authentication.secretsmanager"
},
{
  "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
},
{
  "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
},
{
  "name": "ecs.capability.execution-role-awslogs"
},
{
  "name": "ecs.vpc-id",
  "value": "vpc-1234"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
},
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
},
{
  "name": "ecs.capability.task-eni"
},
{
  "name": "ecs.capability.execution-role-ecr-pull"
},
{
  "name": "ecs.capability.container-health-check"
},
{
  "name": "ecs.subnet-id",
  "value": "subnet-1234"
},
{
  "name": "ecs.instance-type",
  "value": "c5.large"
},
{
```

```

host"
    },
    {
      "name": "ecs.capability.container-ordering"
    },
    {
      "name": "ecs.capability.cni-plugin-version",
      "value": "91ccef8-2019.06.0"
    },
    {
      "name": "ecs.capability.pid-ipc-namespace-sharing"
    },
    {
      "name": "ecs.capability.secrets.ssm.environment-variables"
    },
    {
      "name": "com.amazonaws.ecs.capability.task-iam-role"
    }
  ],
  "registeredAt": 1560788724.507,
  "attachments": [],
  "tags": []
}
],
"failures": []
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateContainerInstancesState](#)」の「」を参照してください。

update-service-primary-task-set

次の例は、update-service-primary-task-set を使用する方法を説明しています。

AWS CLI

サービスのプライマリタスクセットを更新するには

次のupdate-service-primary-task-set例では、指定されたサービスのプライマリタスクセットを更新します。

```
aws ecs update-service-primary-task-set \
```



```
--cluster MyCluster \  
--service MyService \  
--primary-task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789
```

出力:

```
{  
  "taskSet": {  
    "id": "ecs-svc/1234567890123456789",  
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
    "status": "PRIMARY",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sample-fargate:2",  
    "computedDesiredCount": 1,  
    "pendingCount": 0,  
    "runningCount": 0,  
    "createdAt": 1557128360.711,  
    "updatedAt": 1557129412.653,  
    "launchType": "EC2",  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "subnets": [  
          "subnet-12344321"  
        ],  
        "securityGroups": [  
          "sg-12344312"  
        ],  
        "assignPublicIp": "DISABLED"  
      }  
    },  
    "loadBalancers": [],  
    "serviceRegistries": [],  
    "scale": {  
      "value": 50.0,  
      "unit": "PERCENT"  
    },  
    "stabilityStatus": "STABILIZING",  
    "stabilityStatusAt": 1557129279.914  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateServicePrimaryTaskSet](#)」の「」を参照してください。

update-service

次の例は、update-service を使用する方法を説明しています。

AWS CLI

例 1: サービスで使用されるタスク定義を変更するには

次の update-service の例は、amazon-ecs-sample タスク定義を使用するように my-http-service サービスを更新します。

```
aws ecs update-service --service my-http-service --task-definition amazon-ecs-sample
```

例 2: サービス内のタスク数を変更するには

次の update-service の例は、my-http-service サービスのタスクの数を 3 に更新します。

```
aws ecs update-service --service my-http-service --desired-count 3
```

詳細については、「Amazon ECSデベロッパーガイド」の「[サービスの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateService](#)」の「」を参照してください。

update-task-set

次の例は、update-task-set を使用する方法を説明しています。

AWS CLI

タスクセットを更新するには

次のupdate-task-set例では、タスクセットを更新してスケールを調整します。

```
aws ecs update-task-set \
```

```
--cluster MyCluster \  
--service MyService \  
--task-set arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/MyService/ecs-  
svc/1234567890123456789 \  
--scale value=50,unit=PERCENT
```

出力:

```
{  
  "taskSet": {  
    "id": "ecs-svc/1234567890123456789",  
    "taskSetArn": "arn:aws:ecs:us-west-2:123456789012:task-set/MyCluster/  
MyService/ecs-svc/1234567890123456789",  
    "status": "ACTIVE",  
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
sample-fargate:2",  
    "computedDesiredCount": 0,  
    "pendingCount": 0,  
    "runningCount": 0,  
    "createdAt": 1557128360.711,  
    "updatedAt": 1557129279.914,  
    "launchType": "EC2",  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "subnets": [  
          "subnet-12344321"  
        ],  
        "securityGroups": [  
          "sg-12344321"  
        ],  
        "assignPublicIp": "DISABLED"  
      }  
    },  
    "loadBalancers": [],  
    "serviceRegistries": [],  
    "scale": {  
      "value": 50.0,  
      "unit": "PERCENT"  
    },  
    "stabilityStatus": "STABILIZING",  
    "stabilityStatusAt": 1557129279.914  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTaskSet](#)」の「」を参照してください。

を使用した Amazon EFS の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますEFS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-file-system

次の例は、create-file-system を使用する方法を説明しています。

AWS CLI

暗号化されたファイルシステムを作成するには

次のcreate-file-system例では、デフォルトの を使用して暗号化されたファイルシステムを作成しますCMK。タグ も追加されますName=my-file-system。

```
aws efs create-file-system \  
  --performance-mode generalPurpose \  
  --throughput-mode bursting \  
  --encrypted \  
  --tags Key=Name,Value=my-file-system
```

出力:

```
{
```

```
"OwnerId": "123456789012",
"CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",
"FileSystemId": "fs-c7a0456e",
"FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/
fs-48499b4d",
"CreationTime": 1595286880.0,
"LifecycleState": "creating",
"Name": "my-file-system",
"NumberOfMountTargets": 0,
"SizeInBytes": {
  "Value": 0,
  "ValueInIA": 0,
  "ValueInStandard": 0
},
"PerformanceMode": "generalPurpose",
"Encrypted": true,
"KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-
adcf-30d92example",
"ThroughputMode": "bursting",
"Tags": [
  {
    "Key": "Name",
    "Value": "my-file-system"
  }
]
}
```

詳細については、[「Amazon Elastic EFS File System ユーザーガイド」の「Amazon ファイルシステムの作成」](#)を参照してください。 Amazon Elastic File System

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFileSystem](#)」の「」を参照してください。

create-mount-target

次のコード例は、create-mount-target を使用する方法を示しています。

AWS CLI

マウントターゲットを作成するには

次のcreate-mount-target例では、指定されたファイルシステムのマウントターゲットを作成します。

```
aws efs create-mount-target \  
  --file-system-id fs-c7a0456e \  
  --subnet-id subnet-02bf4c428bexample \  
  --security-groups sg-068f739363example
```

出力:

```
{  
  "OwnerId": "123456789012",  
  "MountTargetId": "fsmt-f9a14450",  
  "FileSystemId": "fs-c7a0456e",  
  "SubnetId": "subnet-02bf4c428bexample",  
  "LifeCycleState": "creating",  
  "IpAddress": "10.0.1.24",  
  "NetworkInterfaceId": "eni-02d542216aexample",  
  "AvailabilityZoneId": "use2-az2",  
  "AvailabilityZoneName": "us-east-2b",  
  "VpcId": "vpc-0123456789abcdef0"  
}
```

詳細については、Amazon Elastic File System ユーザーガイド」の「[マウントターゲットの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateMountTarget](#)」の「」を参照してください。

delete-file-system

次の例は、delete-file-system を使用する方法を説明しています。

AWS CLI

ファイルシステムを削除するには

次のdelete-file-system例では、指定されたファイルシステムを削除します。

```
aws efs delete-file-system \  
  --file-system-id fs-c7a0456e
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Elastic EFS File System ユーザーガイド](#)」の「[Amazon ファイルシステムの削除](#)」を参照してください。 Amazon Elastic File System

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFileSystem](#)」の「」を参照してください。

delete-mount-target

次の例は、delete-mount-target を使用する方法を説明しています。

AWS CLI

マウントターゲットを削除するには

次のdelete-mount-target例では、指定されたマウントターゲットを削除します。

```
aws efs delete-mount-target \  
  --mount-target-id fsmt-f9a14450
```

このコマンドでは何も出力されません。

詳細については、Amazon Elastic File System ユーザーガイド」の「[マウントターゲットの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMountTarget](#)」の「」を参照してください。

describe-file-systems

次の例は、describe-file-systems を使用する方法を説明しています。

AWS CLI

ファイルシステムを記述するには

次のdescribe-file-systems例では、指定されたファイルシステムについて説明します。

```
aws efs describe-file-systems \  
  --file-system-id fs-c7a0456e
```

出力:

```
{
  "FileSystems": [
    {
      "OwnerId": "123456789012",
      "CreationToken": "console-d7f56c5f-e433-41ca-8307-9d9c0example",
      "FileSystemId": "fs-c7a0456e",
      "FileSystemArn": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-
system/fs-48499b4d",
      "CreationTime": 1595286880.0,
      "LifecycleState": "available",
      "Name": "my-file-system",
      "NumberOfMountTargets": 3,
      "SizeInBytes": {
        "Value": 6144,
        "Timestamp": 1600991437.0,
        "ValueInIA": 0,
        "ValueInStandard": 6144
      },
      "PerformanceMode": "generalPurpose",
      "Encrypted": true,
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/a59b3472-e62c-42e4-
adcf-30d92example",
      "ThroughputMode": "bursting",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-file-system"
        }
      ]
    }
  ]
}
```

詳細については、[「Amazon Elastic EFS File System ユーザーガイド」の「Amazon ファイルシステムの管理」](#)を参照してください。Amazon Elastic File System

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFileSystems](#)」の「」を参照してください。

describe-mount-targets

次のコード例は、describe-mount-targets を使用する方法を示しています。

AWS CLI

マウントターゲットを記述するには

次のdescribe-mount-targets例では、指定されたマウントターゲットについて説明します。

```
aws efs describe-mount-targets \  
--mount-target-id fsmt-f9a14450
```

出力:

```
{  
  "MountTargets": [  
    {  
      "OwnerId": "123456789012",  
      "MountTargetId": "fsmt-f9a14450",  
      "FileSystemId": "fs-c7a0456e",  
      "SubnetId": "subnet-02bf4c428bexample",  
      "LifecycleState": "creating",  
      "IpAddress": "10.0.1.24",  
      "NetworkInterfaceId": "eni-02d542216aexample",  
      "AvailabilityZoneId": "use2-az2",  
      "AvailabilityZoneName": "us-east-2b",  
      "VpcId": "vpc-0123456789abcdef0"  
    }  
  ]  
}
```

詳細については、Amazon Elastic File System ユーザーガイド」の「[マウントターゲットの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeMountTargets](#)」の「」を参照してください。

describe-tags

次の例は、describe-tags を使用する方法を説明しています。

AWS CLI

ファイルシステムのタグを記述するには

次のdescribe-tags例では、指定されたファイルシステムのタグについて説明します。

```
aws efs describe-tags \  
  --file-system-id fs-c7a0456e
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "my-file-system"  
    },  
    {  
      "Key": "Department",  
      "Value": "Business Intelligence"  
    }  
  ]  
}
```

詳細については、Amazon Elastic File System [「ファイルシステムタグの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを取得するには

次のlist-tags-for-resource例では、指定されたファイルシステムに関連付けられたタグを取得します。

```
aws efs list-tags-for-resource \  
  --resource-id fs-c7a0456e
```

出力:

```
{
  "Tags": [
    {
      "Key": "Name",
      "Value": "my-file-system"
    },
    {
      "Key": "Department",
      "Value": "Business Intelligence"
    }
  ]
}
```

詳細については、Amazon Elastic File System [「ファイルシステムタグの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、Department=Business Intelligence指定されたファイルシステムにタグを追加します。

```
aws efs tag-resource \
  --resource-id fs-c7a0456e \
  --tags Key=Department,Value="Business Intelligence"
```

このコマンドでは何も出力されません。

詳細については、Amazon Elastic File System [「ファイルシステムタグの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグを削除するには

次のuntag-resource例では、指定されたファイルシステムからタグキーを持つDepartmentタグを削除します。

```
aws efs untag-resource \  
  --resource-id fs-c7a0456e \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、Amazon Elastic File System [「ファイルシステムタグの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

を使用した Amazon EKS の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますEKS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-encryption-config

次のコード例は、associate-encryption-config を使用する方法を示しています。

AWS CLI

暗号化設定を既存のクラスターに関連付けるには

次のassociate-encryption-config例では、暗号化が有効になっていない既存のEKSクラスターでの暗号化を有効にします。

```
aws eks associate-encryption-config \
  --cluster-name my-eks-cluster \
  --encryption-config '[{"resources":["secrets"],"provider":
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

出力:

```
{
  "update": {
    "id": "3141b835-8103-423a-8e68-12c2521ffa4d",
    "status": "InProgress",
    "type": "AssociateEncryptionConfig",
    "params": [
      {
        "type": "EncryptionConfig",
        "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":
\\\"arn:aws:kms:region-code:account:key/key\\\"}]}"
      }
    ],
    "createdAt": "2024-03-14T11:01:26.297000-04:00",
    "errors": []
  }
}
```

詳細については、「Amazon EKSユーザーガイド」の [「既存のクラスターでシークレット暗号化を有効にする」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateEncryptionConfig](#)」の「」を参照してください。

associate-identity-provider-config

次の例は、associate-identity-provider-config を使用する方法を説明しています。

AWS CLI

ID プロバイダーを Amazon EKS クラスターに関連付ける

次のassociate-identity-provider-config例では、ID プロバイダーを Amazon EKS クラスターに関連付けます。

```
aws eks associate-identity-provider-config \  
  --cluster-name my-eks-cluster \  
  --oidc 'identityProviderConfigName=my-identity-provider,issuerUrl=https://  
oidc.eks.us-east-2.amazonaws.com/  
id/38D6A4619A0A69E342B113ED7F1A7652,clientId=kubernetes,usernameClaim=email,usernamePrefix=m  
username-prefix,groupsClaim=my-claim,groupsPrefix=my-groups-  
prefix,requiredClaims={Claim1=value1,Claim2=value2}' \  
  --tags env=dev
```

出力:

```
{  
  "update": {  
    "id": "8c6c1bef-61fe-42ac-a242-89412387b8e7",  
    "status": "InProgress",  
    "type": "AssociateIdentityProviderConfig",  
    "params": [  
      {  
        "type": "IdentityProviderConfig",  
        "value": "[{\"type\": \"oidc\", \"name\": \"my-identity-provider\"}]"  
      }  
    ],  
    "createdAt": "2024-04-11T13:46:49.648000-04:00",  
    "errors": []  
  },  
  "tags": {  
    "env": "dev"  
  }  
}
```

詳細については、「[Amazon EKSユーザーガイド](#)」の[OpenID Connect ID プロバイダーからクラスタのユーザーを認証する - OIDC ID プロバイダーを関連付ける](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateIdentityProviderConfig](#)」の「」を参照してください。

create-addon

次の例は、create-addon を使用する方法を説明しています。

AWS CLI

例 1: 各EKSクラスターバージョンのデフォルトの互換性バージョンで Amazon EKSアドオンを作成するには

次のコマンドcreate-addon例では、各EKSクラスターバージョンのデフォルトの互換性バージョンを持つ Amazon EKS アドオンを作成します。

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name
```

出力:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.15.1-eksbuild.1",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/1ec71ee1-b9c2-8915-4e17-e8be0a55a149",  
    "createdAt": "2024-03-14T12:20:03.264000-04:00",  
    "modifiedAt": "2024-03-14T12:20:03.283000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {}  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKSアドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

例 2: 特定のEKSアドオンバージョンで Amazon アドオンを作成するには

次のコマンドcreate-addon例では、特定のEKSアドオンバージョンを使用して Amazon アドオンを作成します。

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
  --addon-version v1.16.4-eksbuild.2
```

出力:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/34c71ee6-7738-6c8b-c6bd-3921a176b5ff",  
    "createdAt": "2024-03-14T12:30:24.507000-04:00",  
    "modifiedAt": "2024-03-14T12:30:24.521000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {}  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKSアドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

例 3: カスタム設定値を使用して Amazon EKS アドオンを作成し、競合の詳細を解決するには

次のコマンドcreate-addon例では、カスタム設定値を使用して Amazon EKS アドオンを作成し、競合の詳細を解決します。

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
  --addon-version v1.16.4-eksbuild.2
```



```
--service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
--addon-version v1.16.4-eksbuild.2 \  
--configuration-values '{"resources":{"limits":{"cpu":"100m"}}}' \  
--resolve-conflicts OVERWRITE
```

出力:

```
{  
  "addon": {  
    "addonName": "my-eks-addon",  
    "clusterName": "my-eks-cluster",  
    "status": "CREATING",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-  
addon/a6c71ee9-0304-9237-1be8-25af1b0f1ffb",  
    "createdAt": "2024-03-14T12:35:58.313000-04:00",  
    "modifiedAt": "2024-03-14T12:35:58.327000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "tags": {},  
    "configurationValues": "{\"resources\":{\"limits\":{\"cpu\":\"100m\"}}}"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

例 4: カスタムJSON設定値ファイルを使用して Amazon EKS アドオンを作成するには

次のコマンド `create-addon` 例では、カスタム設定値を使用して Amazon EKS アドオンを作成し、競合の詳細を解決します。

```
aws eks create-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \  
  --addon-version v1.16.4-eksbuild.2 \  
  --configuration-values 'file://configuration-values.json' \  
  --resolve-conflicts OVERWRITE \  
  --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

configuration-values.json の内容:

```
{
  "resources": {
    "limits": {
      "cpu": "150m"
    }
  },
  "env": {
    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
  }
}
```

出力:

```
{
  "addon": {
    "addonName": "my-eks-addon",
    "clusterName": "my-eks-cluster",
    "status": "CREATING",
    "addonVersion": "v1.16.4-eksbuild.2",
    "health": {
      "issues": []
    },
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/d8c71ef8-fbd8-07d0-fb32-6a7be19eeced",
    "createdAt": "2024-03-14T13:10:51.763000-04:00",
    "modifiedAt": "2024-03-14T13:10:51.777000-04:00",
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {
      "eks-addon-key-1": "value-1",
      "eks-addon-key-2": "value-2"
    },
    "configurationValues": "{\n  \"resources\": {\n    \"limits\": {\n      \"cpu\": \"150m\"\n    }\n  },\n  \"env\": {\n    \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR\"\n  }\n}"
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

例 5: カスタムYAML設定値ファイルを使用して Amazon EKS アドオンを作成するには

次のコマンド `create-addon` 例では、カスタム設定値を使用して Amazon EKS アドオンを作成し、競合の詳細を解決します。

```
aws eks create-addon \
  --cluster-name my-eks-cluster \
  --addon-name my-eks-addon \
  --service-account-role-arn arn:aws:iam::111122223333:role/role-name \
  --addon-version v1.16.4-eksbuild.2 \
  --configuration-values 'file://configuration-values.yaml' \
  --resolve-conflicts OVERWRITE \
  --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

`configuration-values.yaml` の内容:

```
resources:
  limits:
    cpu: '100m'
env:
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

出力:

```
{
  "addon": {
    "addonName": "my-eks-addon",
    "clusterName": "my-eks-cluster",
    "status": "CREATING",
    "addonVersion": "v1.16.4-eksbuild.2",
    "health": {
      "issues": []
    },
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/d4c71efb-3909-6f36-a548-402cd4b5d59e",
    "createdAt": "2024-03-14T13:15:45.220000-04:00",
    "modifiedAt": "2024-03-14T13:15:45.237000-04:00",
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {
      "eks-addon-key-3": "value-3",
      "eks-addon-key-4": "value-4"
    },
    "configurationValues": "resources:\n      limits:\n          cpu: '100m'\nenv:\n  AWS_VPC_K8S_CNI_LOGLEVEL: 'INFO'"
  }
}
```

```
}  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateAddon](#)」の「」を参照してください。

create-cluster

次のコード例は、create-cluster を使用する方法を示しています。

AWS CLI

新しいクラスターを作成するには

このコマンド例は、デフォルトリージョンに prod という名前のクラスターを作成します。

コマンド:

```
aws eks create-cluster --name prod \  
--role-arn arn:aws:iam::012345678910:role/eks-service-role-  
AWSServiceRoleForAmazonEKS-J70NKE3BQ4PI \  
--resources-vpc-config subnetIds=subnet-6782e71e,subnet-  
e7e761ac,securityGroupIds=sg-6979fe18
```

出力:

```
{  
  "cluster": {  
    "name": "prod",  
    "arn": "arn:aws:eks:us-west-2:012345678910:cluster/prod",  
    "createdAt": 1527808069.147,  
    "version": "1.10",  
    "roleArn": "arn:aws:iam::012345678910:role/eks-service-role-  
AWSServiceRoleForAmazonEKS-J70NKE3BQ4PI",  
    "resourcesVpcConfig": {  
      "subnetIds": [  
        "subnet-6782e71e",  
        "subnet-e7e761ac"  
      ],  
    },  
  },  
}
```

```

        "securityGroupIds": [
            "sg-6979fe18"
        ],
        "vpcId": "vpc-950809ec"
    },
    "status": "CREATING",
    "certificateAuthority": {}
}
}

```

プライベートエンドポイントアクセスを含むクラスターを作成してロギングを埋め込むには

このコマンド例は、パブリックエンドポイントアクセスを無効にし、プライベートエンドポイントアクセスを有効にして、すべてのロギングタイプが有効な、example という名前のクラスターをデフォルトのリージョンに作成します。

コマンド:

```

aws eks create-cluster --name example --kubernetes-version 1.12 \
--role-arn arn:aws:iam::012345678910:role/example-cluster-ServiceRole-1XWBQWYSFRE2Q \
--resources-vpc-
config subnetIds=subnet-0a188dccd2f9a632f,subnet-09290d93da4278664,subnet-0f21dd86e0e91134a, \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'

```

出力:

```

{
  "cluster": {
    "name": "example",
    "arn": "arn:aws:eks:us-west-2:012345678910:cluster/example",
    "createdAt": 1565804921.901,
    "version": "1.12",
    "roleArn": "arn:aws:iam::012345678910:role/example-cluster-
ServiceRole-1XWBQWYSFRE2Q",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-0a188dccd2f9a632f",
        "subnet-09290d93da4278664",
        "subnet-0f21dd86e0e91134a",
        "subnet-0173dead68481a583",

```

```
        "subnet-051f70a57ed6fcab6",
        "subnet-01322339c5c7de9b4"
    ],
    "securityGroupIds": [
        "sg-0c5b580845a031c10"
    ],
    "vpcId": "vpc-0f622c01f68d4afec",
    "endpointPublicAccess": false,
    "endpointPrivateAccess": true
},
"logging": {
    "clusterLogging": [
        {
            "types": [
                "api",
                "audit",
                "authenticator",
                "controllerManager",
                "scheduler"
            ],
            "enabled": true
        }
    ]
},
"status": "CREATING",
"certificateAuthority": {},
"platformVersion": "eks.3"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateCluster](#)」の「」を参照してください。

create-fargate-profile

次の例は、create-fargate-profile を使用する方法を説明しています。

AWS CLI

例 1: 名前空間を持つセクタの EKS Fargate プロファイルを作成する

次のcreate-fargate-profile例では、名前空間を持つセクタの EKS Fargate プロファイルを作成します。

```
aws eks create-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \  
  --fargate-profile-name my-fargate-profile \  
  --selectors '[{"namespace": "default"}]'
```

出力:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/a2c72bca-318e-abe8-8ed1-27c6d4892e9e",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T12:38:47.368000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default"  
      }  
    ],  
    "status": "CREATING",  
    "tags": {}  
  }  
}
```

詳細については、「Amazon EKSユーザーガイド」の[AWS「Fargate プロファイルの作成」](#)を参照してください。

例 2: 名前空間とラベルを持つセレクタの EKS Fargate プロファイルを作成する

次のcreate-fargate-profile例では、名前空間とラベルを持つセレクタの EKS Fargate プロファイルを作成します。

```
aws eks create-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \  
  --selectors '[{"namespace": "default", "label": "my-label"}]'
```

```
--fargate-profile-name my-fargate-profile \  
--selectors '[{"namespace": "default", "labels": {"labelname1":  
"labelvalue1"}}]'
```

出力:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/88c72bc7-e8a4-fa34-44e4-2f1397224bb3",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T12:33:48.125000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default",  
        "labels": {  
          "labelname1": "labelvalue1"  
        }  
      }  
    ],  
    "status": "CREATING",  
    "tags": {}  
  }  
}
```

詳細については、「Amazon EKSユーザーガイド」の[AWS 「Fargate プロファイルの作成」](#)を参照してください。

例 3: 名前空間とラベル、および Pod を起動するIDsサブネットを持つセレクタの EKS Fargate プロファイルを作成します。

次のcreate-fargate-profile例では、名前空間とラベル、および Pod を起動するIDsサブネットを持つセレクタの EKS Fargate Profile を作成します。

```
aws eks create-fargate-profile \  
--cluster-name my-eks-cluster \  

```



```
--pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \  
--fargate-profile-name my-fargate-profile \  
--selectors '[{"namespace": "default", "labels": {"labelname1":  
"labelvalue1"}}]' \  
--subnets ["subnet-09d912bb63ef21b9a", "subnet-04ad87f71c6e5ab4d",  
"subnet-0e2907431c9988b72"]
```

出力:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/e8c72bc8-e87b-5eb6-57cb-ed4fe57577e3",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T12:35:58.640000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default",  
        "labels": {  
          "labelname1": "labelvalue1"  
        }  
      }  
    ],  
    "status": "CREATING",  
    "tags": {}  
  }  
}
```

詳細については、「Amazon EKSユーザーガイド」の[AWS「Fargate プロファイルの作成」](#)を参照してください。

例 4: 複数の名前空間とラベルを持つセレクタの EKS Fargate プロファイルを作成し、IDs のサブネットを使用して Pod を起動する

次のcreate-fargate-profile例では、複数の名前空間とラベル、および Pod を起動するIDs サブネットを持つセレクタの EKS Fargate プロファイルを作成します。

```
aws eks create-fargate-profile \
  --cluster-name my-eks-cluster \
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
  --fargate-profile-name my-fargate-profile \
  --selectors '[{"namespace": "default1", "labels": {"labelname1": "labelvalue1",
"labelname2": "labelvalue2"}}, {"namespace": "default2", "labels": {"labelname1":
"labelvalue1", "labelname2": "labelvalue2"}}]' \
  --subnets ['subnet-09d912bb63ef21b9a', "subnet-04ad87f71c6e5ab4d",
"subnet-0e2907431c9988b72"] \
  --tags '{"eks-fargate-profile-key-1": "value-1" , "eks-fargate-profile-key-2":
"value-2"}'
```

出力:

```
{
  "fargateProfile": {
    "fargateProfileName": "my-fargate-profile",
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-
eks-cluster/my-fargate-profile/4cc72bbf-b766-8ee6-8d29-e62748feb3cd",
    "clusterName": "my-eks-cluster",
    "createdAt": "2024-03-19T12:15:55.271000-04:00",
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "subnets": [
      "subnet-09d912bb63ef21b9a",
      "subnet-04ad87f71c6e5ab4d",
      "subnet-0e2907431c9988b72"
    ],
    "selectors": [
      {
        "namespace": "default1",
        "labels": {
          "labelname2": "labelvalue2",
          "labelname1": "labelvalue1"
        }
      },
      {
        "namespace": "default2",
        "labels": {
          "labelname2": "labelvalue2",
          "labelname1": "labelvalue1"
        }
      }
    ]
  },
}
```

```

    "status": "CREATING",
    "tags": {
      "eks-fargate-profile-key-2": "value-2",
      "eks-fargate-profile-key-1": "value-1"
    }
  }
}

```

詳細については、「Amazon EKSユーザーガイド」の[AWS「Fargate プロファイルの作成」](#)を参照してください。

例 5: 名前空間とラベル用のワイルドカードセレクタと、ポッドを起動するIDsサブネットを使用して EKS Fargate Profile を作成する

次のcreate-fargate-profile例では、複数の名前空間とラベル、および Pod を起動するIDsサブネットを持つセレクタの EKS Fargate プロファイルを作成します。

```

aws eks create-fargate-profile \
  --cluster-name my-eks-cluster \
  --pod-execution-role-arn arn:aws:iam::111122223333:role/role-name \
  --fargate-profile-name my-fargate-profile \
  --selectors '[{"namespace": "prod*", "labels": {"labelname*": "*value1"}}, {"namespace": "*dev*", "labels": {"labelname*": "*value*"}}]' \
  --subnets ['subnet-09d912bb63ef21b9a', 'subnet-04ad87f71c6e5ab4d', 'subnet-0e2907431c9988b72'] \
  --tags '{"eks-fargate-profile-key-1": "value-1" , "eks-fargate-profile-key-2": "value-2"}'

```

出力:

```

{
  "fargateProfile": {
    "fargateProfileName": "my-fargate-profile",
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/my-fargate-profile/e8c72bd6-5966-0bfe-b77b-1802893e5a6f",
    "clusterName": "my-eks-cluster",
    "createdAt": "2024-03-19T13:05:20.550000-04:00",
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "subnets": [
      "subnet-09d912bb63ef21b9a",
      "subnet-04ad87f71c6e5ab4d",
      "subnet-0e2907431c9988b72"
    ]
  }
}

```

```
    ],
    "selectors": [
      {
        "namespace": "prod*",
        "labels": {
          "labelname*?": "*value1"
        }
      },
      {
        "namespace": "*dev*",
        "labels": {
          "labelname*?": "*value*"
        }
      }
    ],
    "status": "CREATING",
    "tags": {
      "eks-fargate-profile-key-2": "value-2",
      "eks-fargate-profile-key-1": "value-1"
    }
  }
}
```

詳細については、「Amazon EKSユーザーガイド」の[AWS「Fargate プロファイルの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateFargateProfile](#)」の「」を参照してください。

create-nodegroup

次のコード例は、create-nodegroup を使用する方法を示しています。

AWS CLI

例 1: Amazon EKS クラスターのマネージドノードグループを作成する

次のcreate-nodegroup例では、Amazon EKS クラスターのマネージドノードグループを作成します。

```
aws eks create-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
```

```
--node-role arn:aws:iam::111122223333:role/role-name \  
--  
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a"  
\  
--scaling-config minSize=1,maxSize=3,desiredSize=1 \  
--region us-east-2
```

出力:

```
{  
  "nodegroup": {  
    "nodegroupName": "my-eks-nodegroup",  
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-  
cluster/my-eks-nodegroup/bac7550f-b8b8-5fbb-4f3e-7502a931119e",  
    "clusterName": "my-eks-cluster",  
    "version": "1.26",  
    "releaseVersion": "1.26.12-20240329",  
    "createdAt": "2024-04-04T13:19:32.260000-04:00",  
    "modifiedAt": "2024-04-04T13:19:32.260000-04:00",  
    "status": "CREATING",  
    "capacityType": "ON_DEMAND",  
    "scalingConfig": {  
      "minSize": 1,  
      "maxSize": 3,  
      "desiredSize": 1  
    },  
    "instanceTypes": [  
      "t3.medium"  
    ],  
    "subnets": [  
      "subnet-0e2907431c9988b72, subnet-04ad87f71c6e5ab4d,  
subnet-09d912bb63ef21b9a"  
    ],  
    "amiType": "AL2_x86_64",  
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",  
    "diskSize": 20,  
    "health": {  
      "issues": []  
    },  
    "updateConfig": {  
      "maxUnavailable": 1  
    },  
    "tags": {}  
  }
```

```
}
}
```

詳細については、「Amazon ユーザーガイド」の「[マネージドノードグループの作成](#)」を参照してください。 EKS

例 2: カスタムインスタンスタイプとディスクサイズを持つ Amazon EKS クラスターのマネージドノードグループを作成する

次の `create-nodegroup` 例では、カスタムインスタンスタイプとディスクサイズを持つ Amazon EKS クラスターのマネージドノードグループを作成します。

```
aws eks create-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --node-role arn:aws:iam::111122223333:role/role-name \
  --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a" \
\
  --scaling-config minSize=1,maxSize=3,desiredSize=1 \
  --capacity-type ON_DEMAND \
  --instance-types 'm5.large' \
  --disk-size 50 \
  --region us-east-2
```

出力:

```
{
  "nodegroup": {
    "nodegroupName": "my-eks-nodegroup",
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-nodegroup/c0c7551b-e4f9-73d9-992c-a450fdb82322",
    "clusterName": "my-eks-cluster",
    "version": "1.26",
    "releaseVersion": "1.26.12-20240329",
    "createdAt": "2024-04-04T13:46:07.595000-04:00",
    "modifiedAt": "2024-04-04T13:46:07.595000-04:00",
    "status": "CREATING",
    "capacityType": "ON_DEMAND",
    "scalingConfig": {
      "minSize": 1,
      "maxSize": 3,
```

```

        "desiredSize": 1
    },
    "instanceTypes": [
        "m5.large"
    ],
    "subnets": [
        "subnet-0e2907431c9988b72",
        "subnet-04ad87f71c6e5ab4d",
        "subnet-09d912bb63ef21b9a"
    ],
    "amiType": "AL2_x86_64",
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",
    "diskSize": 50,
    "health": {
        "issues": []
    },
    "updateConfig": {
        "maxUnavailable": 1
    },
    "tags": {}
}
}

```

詳細については、「Amazon ユーザーガイド」の [「マネージドノードグループの作成」](#) を参照してください。 EKS

例 3: カスタムインスタンスタイプ、ディスクサイズ、ami タイプ、容量タイプ、update-config、ラベル、テイント、タグを持つ Amazon EKS クラスターのマネージドノードグループを作成します。

次の create-nodegroup 例では、カスタムインスタンスタイプ、ディスクサイズ、ami タイプ、キャパシティタイプ、update-config、ラベル、テイント、タグを持つ Amazon EKS クラスターのマネージドノードグループを作成します。

```

aws eks create-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --node-role arn:aws:iam::111122223333:role/role-name \
  --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a" \
  \
  --scaling-config minSize=1,maxSize=5,desiredSize=4 \
  --instance-types 't3.large' \

```

```
--disk-size 50 \  
--ami-type AL2_x86_64 \  
--capacity-type SPOT \  
--update-config maxUnavailable=2 \  
--labels '{"my-eks-nodegroup-label-1": "value-1" , "my-eks-nodegroup-label-2":  
"value-2"}' \  
--taints '{"key": "taint-key-1" , "value": "taint-value-1", "effect":  
"NO_EXECUTE"}' \  
--tags '{"my-eks-nodegroup-key-1": "value-1" , "my-eks-nodegroup-key-2":  
"value-2"}'
```

出力:

```
{  
  "nodegroup": {  
    "nodegroupName": "my-eks-nodegroup",  
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-  
cluster/my-eks-nodegroup/88c75524-97af-0cb9-a9c5-7c0423ab5314",  
    "clusterName": "my-eks-cluster",  
    "version": "1.26",  
    "releaseVersion": "1.26.12-20240329",  
    "createdAt": "2024-04-04T14:05:07.940000-04:00",  
    "modifiedAt": "2024-04-04T14:05:07.940000-04:00",  
    "status": "CREATING",  
    "capacityType": "SPOT",  
    "scalingConfig": {  
      "minSize": 1,  
      "maxSize": 5,  
      "desiredSize": 4  
    },  
    "instanceTypes": [  
      "t3.large"  
    ],  
    "subnets": [  
      "subnet-0e2907431c9988b72",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-09d912bb63ef21b9a"  
    ],  
    "amiType": "AL2_x86_64",  
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",  
    "labels": {  
      "my-eks-nodegroup-label-2": "value-2",  
      "my-eks-nodegroup-label-1": "value-1"  
    }  
  }  
}
```



```
    },
    "taints": [
      {
        "key": "taint-key-1",
        "value": "taint-value-1",
        "effect": "NO_EXECUTE"
      }
    ],
    "diskSize": 50,
    "health": {
      "issues": []
    },
    "updateConfig": {
      "maxUnavailable": 2
    },
    "tags": {
      "my-eks-nodegroup-key-1": "value-1",
      "my-eks-nodegroup-key-2": "value-2"
    }
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「マネージドノードグループの作成EKS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateNodegroup](#)」の「」を参照してください。

delete-addon

次のコード例は、delete-addon を使用する方法を示しています。

AWS CLI

例 1。Amazon EKS アドオンを削除し、EKS クラスターでアドオンソフトウェアを保持するには

次のコマンド delete-addon 例では、Amazon EKS アドオンを削除しますが EKS、アドオンソフトウェアはクラスターに保持されます。

```
aws eks delete-addon \
  --cluster-name my-eks-cluster \
  --addon-name my-eks-addon \
```

```
--preserve
```

出力:

```
{
  "addon": {
    "addonName": "my-eks-addon",
    "clusterName": "my-eks-cluster",
    "status": "DELETING",
    "addonVersion": "v1.9.3-eksbuild.7",
    "health": {
      "issues": []
    },
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-addon/a8c71ed3-944e-898b-9167-c763856af4b8",
    "createdAt": "2024-03-14T11:49:09.009000-04:00",
    "modifiedAt": "2024-03-14T12:03:49.776000-04:00",
    "tags": {}
  }
}
```

詳細については、[「Amazon EKS アドオンの管理 - Amazon でのアドオンの削除」](#)を参照してください。 EKS

例 2。Amazon EKS アドオンを削除し、EKS クラスターからアドオンソフトウェアを削除するには

次のコマンド `delete-addon` 例では、Amazon EKS アドオンを削除し、EKS クラスターからアドオンソフトウェアも削除します。

```
aws eks delete-addon \
  --cluster-name my-eks-cluster \
  --addon-name my-eks-addon
```

出力:

```
{
  "addon": {
    "addonName": "my-eks-addon",
    "clusterName": "my-eks-cluster",
    "status": "DELETING",
    "addonVersion": "v1.15.1-eksbuild.1",
```

```
    "health": {
      "issues": []
    },
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/my-eks-
addon/bac71ed1-ec43-3bb6-88ea-f243cdb58954",
    "createdAt": "2024-03-14T11:45:31.983000-04:00",
    "modifiedAt": "2024-03-14T11:58:40.136000-04:00",
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {}
  }
}
```

詳細については、[「Amazon EKS アドオンの管理 - Amazon でのアドオンの削除」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAddon](#)」の「」を参照してください。

delete-cluster

次の例は、delete-cluster を使用する方法を説明しています。

AWS CLI

Amazon EKS クラスターコントロールプレーンを削除する

次の delete-cluster 例では、Amazon EKS クラスターコントロールプレーンを削除します。

```
aws eks delete-cluster \
  --name my-eks-cluster
```

出力:

```
{
  "cluster": {
    "name": "my-eks-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
    "createdAt": "2024-03-14T11:31:44.348000-04:00",
    "version": "1.27",
    "endpoint": "https://DALSJ343KE23J3RN45653DSKJTT647TYD.y14.us-
east-2.eks.amazonaws.com",
    "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
```

```
"resourcesVpcConfig": {
  "subnetIds": [
    "subnet-0fb75d2d8401716e7",
    "subnet-02184492f67a3d0f9",
    "subnet-04098063527aab776",
    "subnet-0e2907431c9988b72",
    "subnet-04ad87f71c6e5ab4d",
    "subnet-09d912bb63ef21b9a"
  ],
  "securityGroupIds": [
    "sg-0c1327f6270afbb36"
  ],
  "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
  "vpcId": "vpc-0012b8e1cc0abb17d",
  "endpointPublicAccess": true,
  "endpointPrivateAccess": true,
  "publicAccessCidrs": [
    "0.0.0.0/0"
  ]
},
"kubernetesNetworkConfig": {
  "serviceIpv4Cidr": "10.100.0.0/16",
  "ipFamily": "ipv4"
},
"logging": {
  "clusterLogging": [
    {
      "types": [
        "api",
        "audit",
        "authenticator",
        "controllerManager",
        "scheduler"
      ],
      "enabled": true
    }
  ]
},
"identity": {
  "oidc": {
    "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
DALSJ343KE23J3RN45653DSKJTT647TYD"
  }
},
```

```
"status": "DELETING",
"certificateAuthority": {
  "data": "XXX_CA_DATA_XXX"
},
"platformVersion": "eks.16",
"tags": {
  "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
  "alpha.eksctl.io/cluster-name": "my-eks-cluster",
  "karpenter.sh/discovery": "my-eks-cluster",
  "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
  "auto-delete": "no",
  "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
  "EKS-Cluster-Name": "my-eks-cluster",
  "alpha.eksctl.io/cluster-oidc-enabled": "true",
  "aws:cloudformation:logical-id": "ControlPlane",
  "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
  "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
},
"accessConfig": {
  "authenticationMode": "API_AND_CONFIG_MAP"
}
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS クラスターの削除」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCluster](#)」の「」を参照してください。

delete-fargate-profile

次のコード例は、delete-fargate-profile を使用する方法を示しています。

AWS CLI

例 1: 名前空間を持つセクタの EKS Fargate プロファイルを作成する

次のdelete-fargate-profile例では、名前空間を持つセクタの EKS Fargate プロファイルを作成します。

```
aws eks delete-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --fargate-profile-name my-fargate-profile
```

出力:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-  
eks-cluster/my-fargate-profile/1ac72bb3-3fc6-2631-f1e1-98bff53bed62",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-03-19T11:48:39.975000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/role-name",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "default",  
        "labels": {  
          "foo": "bar"  
        }  
      }  
    ],  
    "status": "DELETING",  
    "tags": {}  
  }  
}
```

詳細については、「Amazon EKSユーザーガイド」の[AWS「Fargate プロファイル - Fargate の削除」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteFargateProfile](#)」の「」を参照してください。

delete-nodegroup

次の例は、delete-nodegroup を使用する方法を説明しています。

AWS CLI

例 1: Amazon EKS クラスターのマネージドノードグループを削除する

次の `delete-nodegroup` 例では、Amazon EKS クラスターのマネージドノードグループを削除します。

```
aws eks delete-nodegroup \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup
```

出力:

```
{  
  "nodegroup": {  
    "nodegroupName": "my-eks-nodegroup",  
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-nodegroup/1ec75f5f-0e21-dcc0-b46e-f9c442685cd8",  
    "clusterName": "my-eks-cluster",  
    "version": "1.26",  
    "releaseVersion": "1.26.12-20240329",  
    "createdAt": "2024-04-08T13:25:15.033000-04:00",  
    "modifiedAt": "2024-04-08T13:25:31.252000-04:00",  
    "status": "DELETING",  
    "capacityType": "SPOT",  
    "scalingConfig": {  
      "minSize": 1,  
      "maxSize": 5,  
      "desiredSize": 4  
    },  
    "instanceTypes": [  
      "t3.large"  
    ],  
    "subnets": [  
      "subnet-0e2907431c9988b72",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-09d912bb63ef21b9a"  
    ],  
    "amiType": "AL2_x86_64",  
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",  
    "labels": {  
      "my-eks-nodegroup-label-2": "value-2",  
      "my-eks-nodegroup-label-1": "value-1"  
    }  
  }  
}
```

```
    },
    "taints": [
      {
        "key": "taint-key-1",
        "value": "taint-value-1",
        "effect": "NO_EXECUTE"
      }
    ],
    "diskSize": 50,
    "health": {
      "issues": []
    },
    "updateConfig": {
      "maxUnavailable": 2
    },
    "tags": {
      "my-eks-nodegroup-key-1": "value-1",
      "my-eks-nodegroup-key-2": "value-2"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteNodegroup](#)」の「」を参照してください。

deregister-cluster

次の例は、`deregister-cluster` を使用する方法を説明しています。

AWS CLI

接続されたクラスターを登録解除して Amazon EKSコントロールプレーンから削除するには

次の `deregister-cluster` 例では、接続されたクラスターを登録解除して Amazon EKS コントロールプレーンから削除します。

```
aws eks deregister-cluster \
  --name my-eks-anywhere-cluster
```

出力:

```
{
```



```

"cluster": {
  "name": "my-eks-anywhere-cluster",
  "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
  "createdAt": "2024-04-12T12:38:37.561000-04:00",
  "status": "DELETING",
  "tags": {},
  "connectorConfig": {
    "activationId": "dfb5ad28-13c3-4e26-8a19-5b2457638c74",
    "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
    "provider": "EKS_ANYWHERE",
    "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSCoordinatorAgentRole"
  }
}
}
}

```

詳細については、「Amazon EKSユーザーガイド」の「[クラスターの登録解除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterCluster](#)」の「」を参照してください。

describe-addon-configuration

次のコード例は、describe-addon-configuration を使用する方法を示しています。

AWS CLI

例 1: Amazon vpc-cni を作成または更新するときに使用できる設定オプション AddOns

次のdescribe-addon-configuration例では、vpc-cni アドオンのアドオンが各バージョンで作成または更新されたときに使用するすべての使用可能な設定スキーマを返します。

```

aws eks describe-addon-configuration \
  --addon-name vpc-cni \
  --addon-version v1.15.1-eksbuild.1

```

出力:

```

{
  "addonName": "vpc-cni",
  "addonVersion": "v1.15.1-eksbuild.1",
  "configurationSchema": "{\"$ref\":\"#/definitions/VpcCni\",\"$schema\":\"http://json-schema.org/draft-06/schema#\",\"definitions\":{\"Affinity\":{\"type\":

```

```
[\"object\\\", \"null\\\"]}, \"EniConfig\": {\"additionalProperties\": false, \"properties
\": {\"create\": {\"type\": \"boolean\"}, \"region\": {\"type\": \"string\"}, \"subnets\":
\": {\"additionalProperties\": {\"additionalProperties\": false, \"properties\": {\"id\":
\": {\"type\": \"string\"}, \"securityGroups\": {\"items\": {\"type\": \"string\"}, \"type\":
\": \"array\"}}, \"required\": [\"id\"], \"type\": \"object\"}, \"minProperties\": 1, \"type
\": \"object\"}}, \"required\": [\"create\", \"region\", \"subnets\"], \"type\": \"object
\"}, \"Env\": {\"additionalProperties\": false, \"properties\": {\"ADDITIONAL_ENI_TAGS
\": {\"type\": \"string\"}, \"ANNOTATE_POD_IP\": {\"format\": \"boolean\", \"type\":
\": \"string\"}, \"AWS_EC2_ENDPOINT\": {\"type\": \"string\"}, \"AWS_EXTERNAL_SERVICE_CIDRS
\": {\"type\": \"string\"}, \"AWS_MANAGE_ENIS_NON_SCHEDULABLE\": {\"format\": \"boolean
\", \"type\": \"string\"}, \"AWS_VPC_CNI_NODE_PORT_SUPPORT\": {\"format\": \"boolean
\", \"type\": \"string\"}, \"AWS_VPC_ENI_MTU\": {\"format\": \"integer\", \"type\":
\": \"string\"}, \"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\": {\"format\": \"boolean\", \"type
\": \"string\"}, \"AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS\": {\"type\": \"string\"},
\": \"AWS_VPC_K8S_CNI_EXTERNALSNAT\": {\"format\": \"boolean\", \"type\": \"string\"},
\": \"AWS_VPC_K8S_CNI_LOGLEVEL\": {\"type\": \"string\"}, \"AWS_VPC_K8S_CNI_LOG_FILE\":
\": {\"type\": \"string\"}, \"AWS_VPC_K8S_CNI_RANDOMIZESNAT\": {\"type\": \"string\"},
\": \"AWS_VPC_K8S_CNI_VETHPREFIX\": {\"type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_FILE
\": {\"type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\": {\"type\": \"string
\"}, \"CLUSTER_ENDPOINT\": {\"type\": \"string\"}, \"DISABLE_INTROSPECTION\":
\": {\"format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_LEAKED_ENI_CLEANUP\":
\": {\"format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_METRICS\": {\"format
\": \"boolean\", \"type\": \"string\"}, \"DISABLE_NETWORK_RESOURCE_PROVISIONING
\": {\"format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_POD_V6\": {\"format
\": \"boolean\", \"type\": \"string\"}, \"ENABLE_BANDWIDTH_PLUGIN\": {\"format\":
\": \"boolean\", \"type\": \"string\"}, \"ENABLE_POD_ENI\": {\"format\": \"boolean\",
\": \"string\"}, \"ENABLE_PREFIX_DELEGATION\": {\"format\": \"boolean\",
\": \"string\"}, \"ENABLE_V4_EGRESS\": {\"format\": \"boolean\", \"type\":
\": \"string\"}, \"ENABLE_V6_EGRESS\": {\"format\": \"boolean\", \"type\": \"string\"},
\": \"ENI_CONFIG_ANNOTATION_DEF\": {\"type\": \"string\"}, \"ENI_CONFIG_LABEL_DEF\":
\": {\"type\": \"string\"}, \"INTROSPECTION_BIND_ADDRESS\": {\"type\": \"string\"},
\": \"IP_COOLDOWN_PERIOD\": {\"format\": \"integer\", \"type\": \"string\"}, \"MAX_ENI
\": {\"format\": \"integer\", \"type\": \"string\"}, \"MINIMUM_IP_TARGET\": {\"format
\": \"integer\", \"type\": \"string\"}, \"POD_SECURITY_GROUP_ENFORCING_MODE\":
\": {\"type\": \"string\"}, \"WARM_ENI_TARGET\": {\"format\": \"integer\", \"type\":
\": \"string\"}, \"WARM_IP_TARGET\": {\"format\": \"integer\", \"type\": \"string\"},
\": \"WARM_PREFIX_TARGET\": {\"format\": \"integer\", \"type\": \"string\"}}, \"title
\": \"Env\", \"type\": \"object\"}, \"Init\": {\"additionalProperties\": false,
\": \"properties\": {\"env\": {\"$ref\": \"#/definitions/InitEnv\"}}, \"title\": \"Init
\", \"type\": \"object\"}, \"InitEnv\": {\"additionalProperties\": false, \"properties
\": {\"DISABLE_TCP_EARLY_DEMUX\": {\"format\": \"boolean\", \"type\": \"string\"},
\": \"ENABLE_V6_EGRESS\": {\"format\": \"boolean\", \"type\": \"string\"}}, \"title\":
\": \"InitEnv\", \"type\": \"object\"}, \"Limits\": {\"additionalProperties\": false,
\": \"properties\": {\"cpu\": {\"type\": \"string\"}, \"memory\": {\"type\": \"string\"}},
```

```

"title\":"Limits\","type\":"object\"},"NodeAgent\":{"additionalProperties
\":"false","properties\":{"enableCloudWatchLogs\":{"format\":"boolean\","
"type\":"string\"},"enablePolicyEventLogs\":{"format\":"boolean\","type\":"
string\"},"healthProbeBindAddr\":{"format\":"integer\","type\":"string
\"},"metricsBindAddr\":{"format\":"integer\","type\":"string\}}},"title\":"
NodeAgent\","type\":"object\"},"Resources\":{"additionalProperties\":"false,
"properties\":{"limits\":{"$ref\":"#/definitions/Limits\"},"requests\":"
{"$ref\":"#/definitions/Limits\}}},"title\":"Resources\","type\":"object
\"},"Tolerations\":{"additionalProperties\":"false","items\":{"type\":"object
\"},"type\":"array\"},"VpcCni\":{"additionalProperties\":"false","properties
\":{"affinity\":{"$ref\":"#/definitions/Affinity\"},"enableNetworkPolicy\":"
format\":"boolean\","type\":"string\"},"enableWindowsIpam\":{"format\":"
boolean\","type\":"string\"},"eniConfig\":{"$ref\":"#/definitions/EniConfig
\"},"env\":{"$ref\":"#/definitions/Env\"},"init\":{"$ref\":"#/definitions/Init
\"},"livenessProbeTimeoutSeconds\":{"type\":"integer\"},"nodeAgent\":{"$ref\":"
#/definitions/NodeAgent\"},"readinessProbeTimeoutSeconds\":{"type\":"integer
\"},"resources\":{"$ref\":"#/definitions/Resources\"},"tolerations\":{"$ref
\":"#/definitions/Tolerations\}}},"title\":"VpcCni\","type\":"object\}},
"description\":"vpc-cni\}"
}

```

例 2: Amazon CoreDNS を作成または更新するときに使用できる設定オプション AddOns

次のdescribe-addon-configuration例では、それぞれのバージョンで coreDNS アドオンのアドオンを作成または更新するときに使用するすべての使用可能な設定スキーマを返します。

```

aws eks describe-addon-configuration \
  --addon-name coredns \
  --addon-version v1.8.7-eksbuild.4

```

出力:

```

{
  "addonName": "coredns",
  "addonVersion": "v1.8.7-eksbuild.4",
  "configurationSchema": "{\"$ref\":"#/definitions/CoreDNS\","$schema
\":"http://json-schema.org/draft-06/schema#\","definitions\":{"CoreDNS\":"
{"additionalProperties\":"false","properties\":{"computeType\":{"type\":"
string\"},"corefile\":{"description\":"Entire corefile contents to use with
installation\","type\":"string\"},"nodeSelector\":{"additionalProperties\":"
type\":"string\"},"type\":"object\"},"replicaCount\":{"type\":"integer
\"},"resources\":{"$ref\":"#/definitions/Resources\}}},"title\":"CoreDNS\","
type\":"object\"},"Limits\":{"additionalProperties\":"false","properties\":"

```

```
{\"cpu\":{\"type\":\"string\"},\"memory\":{\"type\":\"string\"}},\"title\":\"Limits
\", \"type\":\"object\"},\"Resources\":{\"additionalProperties\":false,\"properties
\":{\"limits\":{\"$ref\":\"#/definitions/Limits\"},\"requests\":{\"$ref\":\"#/
definitions/Limits\"}},\"title\":\"Resources\", \"type\":\"object\"}}"
}
```

詳細については、Amazon の [「Amazon EKS クラスターの kubeconfig ファイルの作成または更新」](#) を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAddonConfiguration](#)」 の「」 を参照してください。

describe-addon-versions

次の例は、describe-addon-versions を使用する方法を説明しています。

AWS CLI

例 1: EKS クラスターで使用できるすべてのアドオンを一覧表示する

次の describe-addon-versions 例では、使用可能なすべての AWS アドオンを一覧表示します。

```
aws eks describe-addon-versions \
  --query 'sort_by(addons &owner)[].{publisher: publisher, owner: owner,
  addonName: addonName, type: type}' \
  --output table
```

出力:

```
-----
|                                                                 DescribeAddonVersions
|                                                                 |
+-----+-----+-----+
+-----+-----+-----+
|          addonName          |          owner          |          publisher
|          type              |                          |
+-----+-----+-----+
+-----+-----+-----+
| vpc-cni                    | aws                     | eks
|   | networking              |                          |
| snapshot-controller        | aws                     | eks
|   | storage                  |                          |
+-----+-----+-----+
```

kube-proxy	aws	eks
networking		
eks-pod-identity-agent	aws	eks
security		
coredns	aws	eks
networking		
aws-mountpoint-s3-csi-driver	aws	s3
storage		
aws-guardduty-agent	aws	eks
security		
aws-efs-csi-driver	aws	eks
storage		
aws-ebs-csi-driver	aws	eks
storage		
amazon-cloudwatch-observability	aws	eks
observability		
adot	aws	eks
observability		
upwind-security_upwind-operator	aws-marketplace	Upwind Security
security		
upbound_universal-crossplane	aws-marketplace	upbound
infra-management		
tetrade-io_istio-distro	aws-marketplace	tetrade-io
policy-management		
teleport_teleport	aws-marketplace	teleport
policy-management		
stormforge_optimize-live	aws-marketplace	StormForge
cost-management		
splunk_splunk-otel-collector-chart	aws-marketplace	Splunk
monitoring		
solo-io_istio-distro	aws-marketplace	Solo.io
service-mesh		
rafay-systems_rafay-operator	aws-marketplace	rafay-systems
kubernetes-management		
new-relic_kubernetes-operator	aws-marketplace	New Relic
observability		
netapp_trident-operator	aws-marketplace	NetApp Inc.
storage		
leaksignal_leakagent	aws-marketplace	leaksignal
monitoring		
kubecost_kubecost	aws-marketplace	kubecost
cost-management		
kong_konnect-ri	aws-marketplace	kong
ingress-service-type		

```

| kasten_k10 | aws-marketplace | Kasten by Veeam
| | data-protection |
| haproxy-technologies_kubernetes-ingress-ee | aws-marketplace | HAProxy
Technologies | ingress-controller |
| groundcover_agent | aws-marketplace | groundcover
| | monitoring |
| grafana-labs_kubernetes-monitoring | aws-marketplace | Grafana Labs
| | monitoring |
| factorhouse_kpow | aws-marketplace | factorhouse
| | monitoring |
| dynatrace_dynatrace-operator | aws-marketplace | dynatrace
| | monitoring |
| datree_engine-pro | aws-marketplace | datree
| | policy-management |
| datadog_operator | aws-marketplace | Datadog
| | monitoring |
| cribl_cribledge | aws-marketplace | Cribl
| | observability |
| calyptia_fluent-bit | aws-marketplace | Calyptia Inc
| | observability |
| accuknox_kubearmor | aws-marketplace | AccuKnox
| | security |
+-----+-----+
+-----+-----+

```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

例 2: 指定された Kubernetes バージョンで使用可能なすべてのアドオンを一覧表示する EKS

次の describe-addon-versions 例では、にサポートされた指定された Kubernetes バージョンで使用可能なすべてのアドオンを一覧表示します EKS。

```

aws eks describe-addon-versions \
  --kubernetes-version=1.26 \
  --query 'sort_by(addons &owner)[].{publisher: publisher, owner: owner,
addonName: addonName, type: type}' \
  --output table

```

出力:

DescribeAddonVersions		
addonName	owner	publisher
type		
vpc-cni	aws	eks
networking		
snapshot-controller	aws	eks
storage		
kube-proxy	aws	eks
networking		
eks-pod-identity-agent	aws	eks
security		
coredns	aws	eks
networking		
aws-mountpoint-s3-csi-driver	aws	s3
storage		
aws-guardduty-agent	aws	eks
security		
aws-efs-csi-driver	aws	eks
storage		
aws-ebs-csi-driver	aws	eks
storage		
amazon-cloudwatch-observability	aws	eks
observability		
adot	aws	eks
observability		
upwind-security_upwind-operator	aws-marketplace	Upwind Security
security		
tetrade-io_istio-distro	aws-marketplace	tetrade-io
policy-management		
stormforge_optimize-live	aws-marketplace	StormForge
cost-management		
splunk_splunk-otel-collector-chart	aws-marketplace	Splunk
monitoring		
solo-io_istio-distro	aws-marketplace	Solo.io
service-mesh		
rafay-systems_rafay-operator	aws-marketplace	rafay-systems
kubernetes-management		
new-relic_kubernetes-operator	aws-marketplace	New Relic
observability		

```

| netapp_trident-operator | aws-marketplace | NetApp Inc.
|   | storage |
| leaksignal_leakagent | aws-marketplace | leaksignal
|   | monitoring |
| kubecost_kubecost | aws-marketplace | kubecost
|   | cost-management |
| kong_konnect-ri | aws-marketplace | kong
|   | ingress-service-type |
| haproxy-technologies_kubernetes-ingress-ee | aws-marketplace | HAProxy
Technologies | ingress-controller |
| groundcover_agent | aws-marketplace | groundcover
|   | monitoring |
| grafana-labs_kubernetes-monitoring | aws-marketplace | Grafana Labs
|   | monitoring |
| dynatrace_dynatrace-operator | aws-marketplace | dynatrace
|   | monitoring |
| datadog_operator | aws-marketplace | Datadog
|   | monitoring |
| cribl_criblemage | aws-marketplace | Cribl
|   | observability |
| calyptia_fluent-bit | aws-marketplace | Calyptia Inc
|   | observability |
| accuknox_kubearmor | aws-marketplace | AccuKnox
|   | security |
+-----+-----+
+-----+-----+

```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKSアドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

例 3: でサポートされている指定された Kubernetes バージョンで使用可能なすべての vpc-cni アドオンバージョンを一覧表示する EKS

次のdescribe-addon-versions例では、 にサポートされた指定された Kubernetes バージョンで使用可能なすべての vpc-cni アドオンバージョンを一覧表示しますEKS。

```

aws eks describe-addon-versions \
  --kubernetes-version=1.26 \
  --addon-name=vpc-cni \
  --query='addons[].addonVersions[].addonVersion'

```

出力:


```
[
  "v1.18.0-eksbuild.1",
  "v1.17.1-eksbuild.1",
  "v1.16.4-eksbuild.2",
  "v1.16.3-eksbuild.2",
  "v1.16.2-eksbuild.1",
  "v1.16.0-eksbuild.1",
  "v1.15.5-eksbuild.1",
  "v1.15.4-eksbuild.1",
  "v1.15.3-eksbuild.1",
  "v1.15.1-eksbuild.1",
  "v1.15.0-eksbuild.2",
  "v1.14.1-eksbuild.1",
  "v1.14.0-eksbuild.3",
  "v1.13.4-eksbuild.1",
  "v1.13.3-eksbuild.1",
  "v1.13.2-eksbuild.1",
  "v1.13.0-eksbuild.1",
  "v1.12.6-eksbuild.2",
  "v1.12.6-eksbuild.1",
  "v1.12.5-eksbuild.2",
  "v1.12.0-eksbuild.2"
]
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKSアドオンの管理 - アドオンの作成」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAddonVersions](#)」の「」を参照してください。

describe-addon

次のコード例は、describe-addon を使用する方法を示しています。

AWS CLI

Amazon EKSクラスターでアクティブに実行されているEKSアドオンを説明する

次のdescribe-addon例では、Amazon EKSクラスターでEKSアドオンをアクティブに実行しています。

```
aws eks describe-addon \
```

```
--cluster-name my-eks-cluster \  
--addon-name vpc-cni
```

出力:

```
{  
  "addon": {  
    "addonName": "vpc-cni",  
    "clusterName": "my-eks-cluster",  
    "status": "ACTIVE",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f",  
    "createdAt": "2024-03-14T13:18:45.417000-04:00",  
    "modifiedAt": "2024-03-14T13:18:49.557000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-Yfakrq0C1UTm",  
    "tags": {  
      "eks-addon-key-3": "value-3",  
      "eks-addon-key-4": "value-4"  
    },  
    "configurationValues": "resources:\n      limits:\n        cpu: '100m'\n      nenv:\n        AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAddon](#)」の「」を参照してください。

describe-cluster

次のコード例は、describe-cluster を使用する方法を示しています。

AWS CLI

Amazon EKS クラスターでアクティブに実行されている EKS アドオンの説明

次の describe-cluster 例では、Amazon EKS クラスターで EKS アドオンをアクティブに実行しています。

```
aws eks describe-cluster \  
--cluster-name my-eks-cluster
```

出力:

```
{  
  "cluster": {  
    "name": "my-eks-cluster",  
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",  
    "createdAt": "2024-03-14T11:31:44.348000-04:00",  
    "version": "1.26",  
    "endpoint": "https://JSA79429HJDASKJDJ8223829MNDNASW.y14.us-  
east-2.eks.amazonaws.com",  
    "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-  
ServiceRole-zMF6CBakwbbW",  
    "resourcesVpcConfig": {  
      "subnetIds": [  
        "subnet-0fb75d2d8401716e7",  
        "subnet-02184492f67a3d0f9",  
        "subnet-04098063527aab776",  
        "subnet-0e2907431c9988b72",  
        "subnet-04ad87f71c6e5ab4d",  
        "subnet-09d912bb63ef21b9a"  
      ],  
      "securityGroupIds": [  
        "sg-0c1327f6270afbb36"  
      ],  
      "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",  
      "vpcId": "vpc-0012b8e1cc0abb17d",  
      "endpointPublicAccess": true,  
      "endpointPrivateAccess": true,  
      "publicAccessCidrs": [  
        "22.19.18.2/32"  
      ]  
    },  
    "kubernetesNetworkConfig": {  
      "serviceIpv4Cidr": "10.100.0.0/16",  
      "ipFamily": "ipv4"  
    },  
    "logging": {  
      "clusterLogging": [  
        {  
          "types": [  

```

```
        "api",
        "audit",
        "authenticator",
        "controllerManager",
        "scheduler"
    ],
    "enabled": true
  }
]
},
"identity": {
  "oidc": {
    "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
JSA79429HJDASKJDJ8223829MNDNASW"
  }
},
"status": "ACTIVE",
"certificateAuthority": {
  "data": "CA_DATA_STRING..."
},
"platformVersion": "eks.14",
"tags": {
  "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
  "alpha.eksctl.io/cluster-name": "my-eks-cluster",
  "karpenter.sh/discovery": "my-eks-cluster",
  "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-
beae-0a9599c8c7ed",
  "auto-delete": "no",
  "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
  "EKS-Cluster-Name": "my-eks-cluster",
  "alpha.eksctl.io/cluster-oidc-enabled": "true",
  "aws:cloudformation:logical-id": "ControlPlane",
  "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z",
  "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
},
"health": {
  "issues": []
},
"accessConfig": {
  "authenticationMode": "API_AND_CONFIG_MAP"
}
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCluster](#)」の「」を参照してください。

describe-fargate-profile

次のコード例は、describe-fargate-profile を使用する方法を示しています。

AWS CLI

Fargate プロファイルの説明

次のdescribe-fargate-profile例では、Fargate プロファイルについて説明します。

```
aws eks describe-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --fargate-profile-name my-fargate-profile
```

出力:

```
{  
  "fargateProfile": {  
    "fargateProfileName": "my-fargate-profile",  
    "fargateProfileArn": "arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/my-fargate-profile/96c766ce-43d2-f9c9-954c-647334391198",  
    "clusterName": "my-eks-cluster",  
    "createdAt": "2024-04-11T10:42:52.486000-04:00",  
    "podExecutionRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-farga-FargatePodExecutionRole-1htfAaJdJUE0",  
    "subnets": [  
      "subnet-09d912bb63ef21b9a",  
      "subnet-04ad87f71c6e5ab4d",  
      "subnet-0e2907431c9988b72"  
    ],  
    "selectors": [  
      {  
        "namespace": "prod*",  
        "labels": {  
          "labelname*?": "*value1"  
        }  
      }  
    ],  
  },  
}
```

```
    {
      "namespace": "*dev*",
      "labels": {
        "labelname*?": "*value*"
      }
    }
  ],
  "status": "ACTIVE",
  "tags": {
    "eks-fargate-profile-key-2": "value-2",
    "eks-fargate-profile-key-1": "value-1"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFargateProfile](#)」の「」を参照してください。

describe-identity-provider-config

次のコード例は、describe-identity-provider-config を使用する方法を示しています。

AWS CLI

Amazon クラスターに関連付けられた ID EKS プロバイダー設定を説明する

次のdescribe-identity-provider-config例では、Amazon クラスターに関連付けられた ID EKS プロバイダー設定について説明します。

```
aws eks describe-identity-provider-config \
  --cluster-name my-eks-cluster \
  --identity-provider-config type=oidc,name=my-identity-provider
```

出力:

```
{
  "identityProviderConfig": {
    "oidc": {
      "identityProviderConfigName": "my-identity-provider",
      "identityProviderConfigArn": "arn:aws:eks:us-
east-2:111122223333:identityproviderconfig/my-eks-cluster/oidc/my-identity-
provider/8ac76722-78e4-cec1-ed76-d49eea058622",
```

```
    "clusterName": "my-eks-cluster",
    "issuerUrl": "https://oidc.eks.us-east-2.amazonaws.com/
id/38D6A4619A0A69E342B113ED7F1A7652",
    "clientId": "kubernetes",
    "usernameClaim": "email",
    "usernamePrefix": "my-username-prefix",
    "groupsClaim": "my-claim",
    "groupsPrefix": "my-groups-prefix",
    "requiredClaims": {
      "Claim1": "value1",
      "Claim2": "value2"
    },
    "tags": {
      "env": "dev"
    },
    "status": "ACTIVE"
  }
}
```

詳細については、「[Amazon EKSユーザーガイド](#)」の[OpenID Connect ID プロバイダーからクラスタのユーザーを認証する](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeIdentityProviderConfig](#)」の「」を参照してください。

describe-nodegroup

次のコード例は、describe-nodegroup を使用する方法を示しています。

AWS CLI

Amazon EKS クラスターのマネージドノードグループの説明

次のdescribe-nodegroup例では、Amazon EKS クラスターのマネージドノードグループについて説明します。

```
aws eks describe-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup
```

出力:

```
{
  "nodegroup": {
    "nodegroupName": "my-eks-nodegroup",
    "nodegroupArn": "arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-
cluster/my-eks-nodegroup/a8c75f2f-df78-a72f-4063-4b69af3de5b1",
    "clusterName": "my-eks-cluster",
    "version": "1.26",
    "releaseVersion": "1.26.12-20240329",
    "createdAt": "2024-04-08T11:42:10.555000-04:00",
    "modifiedAt": "2024-04-08T11:44:12.402000-04:00",
    "status": "ACTIVE",
    "capacityType": "ON_DEMAND",
    "scalingConfig": {
      "minSize": 1,
      "maxSize": 3,
      "desiredSize": 1
    },
    "instanceTypes": [
      "t3.medium"
    ],
    "subnets": [
      "subnet-0e2907431c9988b72",
      "subnet-04ad87f71c6e5ab4d",
      "subnet-09d912bb63ef21b9a"
    ],
    "amiType": "AL2_x86_64",
    "nodeRole": "arn:aws:iam::111122223333:role/role-name",
    "labels": {},
    "resources": {
      "autoScalingGroups": [
        {
          "name": "eks-my-eks-nodegroup-a8c75f2f-df78-
a72f-4063-4b69af3de5b1"
        }
      ]
    },
    "diskSize": 20,
    "health": {
      "issues": []
    },
    "updateConfig": {
      "maxUnavailable": 1
    },
  },
}
```



```
    "tags": {}
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeNodegroup](#)」の「」を参照してください。

describe-update

次のコード例は、describe-update を使用する方法を示しています。

AWS CLI

例 1: クラスターの更新を記述するには

次のdescribe-update例では、 という名前のクラスターの更新について説明します。

```
aws eks describe-update \  
  --name my-eks-cluster \  
  --update-id 10bddb13-a71b-425a-b0a6-71cd03e59161
```

出力:

```
{  
  "update": {  
    "id": "10bddb13-a71b-425a-b0a6-71cd03e59161",  
    "status": "Successful",  
    "type": "EndpointAccessUpdate",  
    "params": [  
      {  
        "type": "EndpointPublicAccess",  
        "value": "false"  
      },  
      {  
        "type": "EndpointPrivateAccess",  
        "value": "true"  
      }  
    ],  
    "createdAt": "2024-03-14T10:01:26.297000-04:00",  
    "errors": []  
  }  
}
```

```
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKSクラスター Kubernetes バージョンの更新」](#)を参照してください。 EKS

例 2: クラスターの更新を記述するには

次のdescribe-update例では、 という名前のクラスターの更新について説明します。

```
aws eks describe-update \  
  --name my-eks-cluster \  
  --update-id e4994991-4c0f-475a-a040-427e6da52966
```

出力:

```
{  
  "update": {  
    "id": "e4994991-4c0f-475a-a040-427e6da52966",  
    "status": "Successful",  
    "type": "AssociateEncryptionConfig",  
    "params": [  
      {  
        "type": "EncryptionConfig",  
        "value": "[{\"resources\":[\"secrets\"],\"provider\":{\"keyArn\":  
\"arn:aws:kms:region-code:account:key/key\"}}]"  
      }  
    ],  
    "createdAt": "2024-03-14T11:01:26.297000-04:00",  
    "errors": []  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKSクラスター Kubernetes バージョンの更新」](#)を参照してください。 EKS

例 3: クラスターの更新を記述するには

次のdescribe-update例では、 という名前のクラスターの更新について説明します。

```
aws eks describe-update \  
  --name my-eks-cluster \  
  --update-id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

出力:

```
{
  "update": {
    "id": "b5f0ba18-9a87-4450-b5a0-825e6e84496f",
    "status": "Successful",
    "type": "VersionUpdate",
    "params": [
      {
        "type": "Version",
        "value": "1.29"
      },
      {
        "type": "PlatformVersion",
        "value": "eks.1"
      }
    ],
    "createdAt": "2024-03-14T12:05:26.297000-04:00",
    "errors": []
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS クラスター Kubernetes バージョンの更新」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUpdate](#)」の「」を参照してください。

disassociate-identity-provider-config

次のコード例は、disassociate-identity-provider-config を使用する方法を示しています。

AWS CLI

ID プロバイダーと Amazon EKS クラスターの関連付けを解除する

次のdisassociate-identity-provider-config例では、ID プロバイダーと Amazon EKS クラスターの関連付けを解除します。

```
aws eks disassociate-identity-provider-config \
  --cluster-name my-eks-cluster \
```

```
--identity-provider-config 'type=oidc,name=my-identity-provider'
```

出力:

```
{
  "update": {
    "id": "5f78d14e-c57b-4857-a3e4-cf664ae20949",
    "status": "InProgress",
    "type": "DisassociateIdentityProviderConfig",
    "params": [
      {
        "type": "IdentityProviderConfig",
        "value": "[]"
      }
    ],
    "createdAt": "2024-04-11T13:53:43.314000-04:00",
    "errors": []
  }
}
```

詳細については、「[Amazon EKSユーザーガイド](#)」の[OpenID Connect ID プロバイダーからのクラスタのユーザーの認証](#) - クラスタから OIDC ID プロバイダーの関連付けを解除する「」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDisassociateIdentityProviderConfig](#)」の「」を参照してください。

get-token

次のコード例は、get-token を使用する方法を示しています。

AWS CLI

例 1: my-eks-cluster 「」という名前の Amazon EKS クラスタの認証トークンを取得する

次のget-token例では、という名前の Amazon EKS クラスタの認証トークンを取得します my-eks-cluster。

```
aws eks get-token \
  --cluster-name my-eks-cluster
```

出力:

```
{
  "kind": "ExecCredential",
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "spec": {},
  "status": {
    "expirationTimestamp": "2024-04-11T20:59:56Z",
    "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
  }
}
```

例 2: トークンの署名時に、このロールARNを認証情報として引き受けることで、my-eks-cluster「」という名前の Amazon EKS クラスターの認証トークンを取得します

次のget-token例では、トークンの署名時に認証情報のこのロールARNを引き受けることで、という名前 my-eks-clusterの Amazon EKS クラスターの認証トークンを取得します。

```
aws eks get-token \
  --cluster-name my-eks-cluster \
  --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-ServiceRole-j1k7AfTIQtnM
```

出力:

```
{
  "kind": "ExecCredential",
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "spec": {},
  "status": {
    "expirationTimestamp": "2024-04-11T21:05:26Z",
    "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetToken](#)」の「」を参照してください。

list-addons

次のコード例は、list-addons を使用する方法を示しています。

AWS CLI

my-eks-cluster 「」という名前の Amazon EKS クラスターにインストールされているすべてのアドオンを一覧表示します。

次の `list-addons` 例では、という名前の Amazon EKS クラスターにインストールされているすべてのアドオンを一覧表示します my-eks-cluster。

```
aws eks list-addons \
  --cluster-name my-eks-cluster
```

出力:

```
{
  "addons": [
    "kube-proxy",
    "vpc-cni"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAddons](#)」の「」を参照してください。

list-clusters

次のコード例は、`list-clusters` を使用する方法を示しています。

AWS CLI

my-eks-cluster 「」という名前の Amazon EKS クラスターにインストールされているすべてのアドオンを一覧表示するには

次の `list-clusters` 例では、という名前の Amazon EKS クラスターにインストールされているすべてのアドオンを一覧表示します my-eks-cluster。

```
aws eks list-clusters
```

出力:

```
{
```

```
"clusters": [
  "prod",
  "qa",
  "stage",
  "my-eks-cluster"
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListClusters](#)」の「」を参照してください。

list-fargate-profiles

次の例は、list-fargate-profiles を使用する方法を説明しています。

AWS CLI

my-eks-cluster 「」という名前の Amazon EKS クラスター内のすべてのファーゲートプロフィールを一覧表示するには

次のlist-fargate-profiles例では、という名前の Amazon EKS クラスター内のすべてのファーゲートプロフィールを一覧表示します my-eks-cluster。

```
aws eks list-fargate-profiles \
  --cluster-name my-eks-cluster
```

出力:

```
{
  "fargateProfileNames": [
    "my-fargate-profile"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFargateProfiles](#)」の「」を参照してください。

list-identity-provider-configs

次のコード例は、list-identity-provider-configs を使用する方法を示しています。

AWS CLI

Amazon クラスターに関連付けられている ID EKS プロバイダーを一覧表示する

次のlist-identity-provider-configs例では、Amazon クラスターに関連付けられている ID EKS プロバイダーを一覧表示します。

```
aws eks list-identity-provider-configs \  
  --cluster-name my-eks-cluster
```

出力:

```
{  
  "identityProviderConfigs": [  
    {  
      "type": "oidc",  
      "name": "my-identity-provider"  
    }  
  ]  
}
```

詳細については、「[Amazon EKSユーザーガイド](#)」の[OpenID Connect ID プロバイダーからクラスターのユーザーを認証する](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListIdentityProviderConfigs](#)」の「」を参照してください。

list-nodegroups

次の例は、list-nodegroups を使用する方法を説明しています。

AWS CLI

Amazon EKSクラスター内のすべてのノードグループを一覧表示する

次のlist-nodegroups例では、Amazon EKSクラスター内のすべてのノードグループを一覧表示します。

```
aws eks list-nodegroups \  
  --cluster-name my-eks-cluster
```

出力:


```
{
  "nodegroups": [
    "my-eks-managed-node-group",
    "my-eks-nodegroup"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListNodegroups](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

例 1: Amazon EKS クラスターのすべてのタグを一覧表示するには ARN

次のlist-tags-for-resource例では、Amazon EKSクラスター のすべてのタグを一覧表示しますARN。

```
aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

出力:

```
{
  "tags": {
    "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-cluster",
    "alpha.eksctl.io/cluster-name": "my-eks-cluster",
    "karpenter.sh/discovery": "my-eks-cluster",
    "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-east-2:111122223333:stack/eksctl-my-eks-cluster-cluster/e752ea00-e217-11ee-beae-0a9599c8c7ed",
    "auto-delete": "no",
    "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
    "EKS-Cluster-Name": "my-eks-cluster",
    "alpha.eksctl.io/cluster-oidc-enabled": "true",
    "aws:cloudformation:logical-id": "ControlPlane",
    "alpha.eksctl.io/eksctl-version": "0.173.0-dev+7ee89342.2024-03-01T03:40:57Z",
  }
}
```

```

    "Name": "eksctl-my-eks-cluster-cluster/ControlPlane"
  }
}

```

例 2: Amazon EKS Node グループのすべてのタグを一覧表示するには ARN

次の `list-tags-for-resource` 例では、Amazon EKS Node グループ のすべてのタグを一覧表示しますARN。

```

aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c

```

出力:

```

{
  "tags": {
    "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-nodegroup-my-eks-managed-node-group",
    "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-east-2:111122223333:stack/eksctl-my-eks-cluster-nodegroup-my-eks-managed-node-group/ea20310-e219-11ee-b851-0ab9ad8228ff",
    "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
    "EKS-Cluster-Name": "my-eks-cluster",
    "alpha.eksctl.io/nodegroup-type": "managed",
    "NodeGroup Name 1": "my-eks-managed-node-group",
    "k8s.io/cluster-autoscaler/enabled": "true",
    "nodegroup-role": "worker",
    "alpha.eksctl.io/cluster-name": "my-eks-cluster",
    "alpha.eksctl.io/nodegroup-name": "my-eks-managed-node-group",
    "karpenter.sh/discovery": "my-eks-cluster",
    "NodeGroup Name 2": "AmazonLinux-Linux-Managed-NG-v1-26-v1",
    "auto-delete": "no",
    "k8s.io/cluster-autoscaler/my-eks-cluster": "owned",
    "aws:cloudformation:logical-id": "ManagedNodeGroup",
    "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z"
  }
}

```

例 3: Amazon EKS Fargate プロファイルのすべてのタグを一覧表示するには ARN

次のlist-tags-for-resource例では、Amazon EKS Fargate プロファイル のすべてのタグを一覧表示しますARN。

```
aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:fargateprofile/my-eks-cluster/my-fargate-profile/d6c76780-e541-0725-c816-36754cab734b
```

出力:

```
{
  "tags": {
    "eks-fargate-profile-key-2": "value-2",
    "eks-fargate-profile-key-1": "value-1"
  }
}
```

例 4: Amazon EKS アドオンのすべてのタグを一覧表示するには ARN

次のlist-tags-for-resource例では、Amazon EKS アドオン のすべてのタグを一覧表示しますARN。

```
aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f
```

出力:

```
{
  "tags": {
    "eks-addon-key-2": "value-2",
    "eks-addon-key-1": "value-1"
  }
}
```

例 5: Amazon EKS OIDC ID プロバイダーのすべてのタグを一覧表示するには ARN

次のlist-tags-for-resource例では、Amazon EKS OIDC ID プロバイダー のすべてのタグを一覧表示しますARN。

```
aws eks list-tags-for-resource \
```

```
--resource-arn arn:aws:eks:us-east-2:111122223333:identityproviderconfig/my-eks-cluster/oidc/my-identity-provider/8ac76722-78e4-cec1-ed76-d49eea058622
```

出力:

```
{
  "tags": {
    "my-identity-provider": "test"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-update

次の例は、list-update を使用する方法を説明しています。

AWS CLI

例 1: Amazon EKS クラスター名に関連付けられた更新を一覧表示するには

次のlist-updates例では、Amazon EKS クラスター名のすべての更新IDsを一覧表示します。

```
aws eks list-updates \
  --name my-eks-cluster
```

出力:

```
{
  "updateIds": [
    "5f78d14e-c57b-4857-a3e4-cf664ae20949",
    "760e5a3f-adad-48c7-88d3-7ac283c09c26",
    "cd4ec863-bc55-47d5-a377-3971502f529b",
    "f12657ce-e869-4f17-b158-a82ab8b7d937"
  ]
}
```

例 2: Amazon EKS Node グループのすべての更新IDsを一覧表示するには

次のlist-updates例では、Amazon EKS Node グループのすべての更新IDsを一覧表示します。

```
aws eks list-updates \  
  --name my-eks-cluster \  
  --nodegroup-name my-eks-managed-node-group
```

出力:

```
{  
  "updateIds": [  
    "8c6c1bef-61fe-42ac-a242-89412387b8e7"  
  ]  
}
```

例 3: Amazon EKS Add-on のすべての更新を一覧表示IDsするには

次のlist-updates例では、Amazon EKS アドオンのすべての更新IDsを一覧表示します。

```
aws eks list-updates \  
  --name my-eks-cluster \  
  --addon-name vpc-cni
```

出力:

```
{  
  "updateIds": [  
    "9cdba8d4-79fb-3c83-afe8-00b508d33268"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUpdate](#)」の「」を参照してください。

list-updates

次の例は、list-updates を使用する方法を説明しています。

AWS CLI

クラスターの更新を一覧表示するには

このコマンド例では、デフォルトのリージョンexampleで という名前のクラスターの現在の更新を一覧表示します。

コマンド:

```
aws eks list-updates --name example
```

出力:

```
{
  "updateIds": [
    "10bddb13-a71b-425a-b0a6-71cd03e59161"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUpdates](#)」の「」を参照してください。

register-cluster

次のコード例は、register-cluster を使用する方法を示しています。

AWS CLI

例 1: 外部 EKS_ANYWHERE Kubernetes クラスターを Amazon に登録する EKS

次のregister-cluster例では、外部 EKS_ANYWHERE Kubernetes クラスターを Amazon に登録しますEKS。

```
aws eks register-cluster \
  --name my-eks-anywhere-cluster \
  --connector-config 'roleArn=arn:aws:iam::111122223333:role/AmazonEKSCoordinatorAgentRole,provider=EKS_ANYWHERE'
```

出力:

```
{
  "cluster": {
    "name": "my-eks-anywhere-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
    "createdAt": "2024-04-12T12:38:37.561000-04:00",
    "status": "PENDING",
    "tags": {},
    "connectorConfig": {
      "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
    }
  }
}
```

```
    "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
    "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
    "provider": "EKS_ANYWHERE",
    "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSCoordinatorAgentRole"
  }
}
```

詳細については、「Amazon EKSユーザーガイド」の「[外部クラスターの接続](#)」を参照してください。

例 2: 外部 Kubernetes クラスターを Amazon に登録する EKS

次のregister-cluster例では、外部 EKS_ANYWHERE Kubernetes クラスターを Amazon に登録しますEKS。

```
aws eks register-cluster \
  --name my-eks-anywhere-cluster \
  --connector-config 'roleArn=arn:aws:iam::111122223333:role/AmazonEKSCoordinatorAgentRole,provider=OTHER'
```

出力:

```
{
  "cluster": {
    "name": "my-onprem-k8s-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-onprem-k8s-cluster",
    "createdAt": "2024-04-12T12:42:10.861000-04:00",
    "status": "PENDING",
    "tags": {},
    "connectorConfig": {
      "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
      "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
      "activationExpiry": "2024-04-15T12:42:10.339000-04:00",
      "provider": "OTHER",
      "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSCoordinatorAgentRole"
    }
  }
}
```

詳細については、「Amazon EKSユーザーガイド」の「[外部クラスターの接続](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterCluster](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

例 1: 指定されたタグを Amazon EKS クラスターに追加するには

次のtag-resource例では、指定されたタグを Amazon EKS クラスターに追加します。

```
aws eks tag-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \  
  --tag 'my-eks-cluster-test-1=test-value-1,my-eks-cluster-dev-1=dev-value-2'
```

このコマンドでは何も出力されません。

例 2: 指定されたタグを Amazon EKS Node グループに追加するには

次のtag-resource例では、指定されたタグを Amazon EKS Node グループに追加します。

```
aws eks tag-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \  
  --tag 'my-eks-nodegroup-test-1=test-value-1,my-eks-nodegroup-dev-1=dev-value-2'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

例 1: Amazon EKS クラスターから指定されたタグを削除するには

次のuntag-resource例では、Amazon EKS クラスターから指定されたタグを削除します。


```
aws eks untag-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster \  
  --tag-keys "my-eks-cluster-test-1" "my-eks-cluster-dev-1"
```

このコマンドでは何も出力されません。

例 2: Amazon EKS Node グループから指定されたタグを削除するには

次の `untag-resource` 例では、Amazon EKS Node グループから指定されたタグを削除します。

```
aws eks untag-resource \  
  --resource-arn arn:aws:eks:us-east-2:111122223333:nodegroup/my-eks-cluster/my-eks-managed-node-group/60c71ed2-2cfb-020f-a5f4-ad32477f198c \  
  --tag-keys "my-eks-nodegroup-test-1" "my-eks-nodegroup-dev-1"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-addon

次のコード例は、`update-addon` を使用する方法を示しています。

AWS CLI

例 1。サービスアカウントロールで Amazon EKS アドオンを更新するには ARN

次のコマンド `update-addon` 例では、サービスアカウントロールを使用して Amazon EKS アドオンを更新しますARN。

```
aws eks update-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name vpc-cni \  
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-Yfakrq0C1UTm
```

出力:

```
{  
  "update": {  
    "id": "c00d2de2-c2e4-3d30-929e-46b8edec2ce4",
```

```
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "ServiceAccountRoleArn",
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      }
    ],
    "updatedAt": "2024-04-12T16:04:55.614000-04:00",
    "errors": []
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの更新」](#)を参照してください。 EKS

例 2。特定のEKSアドオンバージョンで Amazon アドオンを更新するには

次のコマンドupdate-addon例では、Amazon EKS アドオンを特定のアドオンバージョンで更新します。

```
aws eks update-addon \
  --cluster-name my-eks-cluster \
  --addon-name vpc-cni \
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm \
  --addon-version v1.16.4-eksbuild.2
```

出力:

```
{
  "update": {
    "id": "f58dc0b0-2b18-34bd-bc6a-e4abc0011f36",
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "AddonVersion",
        "value": "v1.16.4-eksbuild.2"
      },
      {
        "type": "ServiceAccountRoleArn",
```

```

        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      }
    ],
    "createdAt": "2024-04-12T16:07:16.550000-04:00",
    "errors": []
  }
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの更新」](#)を参照してください。 EKS

例 3。Amazon EKS アドオンをカスタム設定値で更新し、競合の詳細を解決するには

次のコマンド `update-addon` 例では、Amazon EKS アドオンをカスタム設定値で更新し、競合の詳細を解決します。

```

aws eks update-addon \
  --cluster-name my-eks-cluster \
  --addon-name vpc-cni \
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm \
  --addon-version v1.16.4-eksbuild.2 \
  --configuration-values '{"resources": {"limits":{"cpu":"100m"}, "requests":
{"cpu":"50m"}}}' \
  --resolve-conflicts PRESERVE

```

出力:

```

{
  "update": {
    "id": "cd9f2173-a8d8-3004-a90f-032f14326520",
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "AddonVersion",
        "value": "v1.16.4-eksbuild.2"
      },
      {
        "type": "ServiceAccountRoleArn",
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      }
    ]
  }
}

```

```

    },
    {
      "type": "ResolveConflicts",
      "value": "PRESERVE"
    },
    {
      "type": "ConfigurationValues",
      "value": "{\"resources\": {\"limits\": {\"cpu\": \"100m\"}, \"requests\": {\"cpu\": \"50m\"}}}"
    }
  ],
  "createdAt": "2024-04-12T16:16:27.363000-04:00",
  "errors": []
}
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの更新」](#)を参照してください。 EKS

例 4. カスタムJSON設定値ファイルを使用して Amazon EKS アドオンを更新するには

次のコマンドupdate-addon例では、Amazon EKS アドオンをカスタムJSON設定値で更新し、競合の詳細を解決します。

```

aws eks update-addon \
  --cluster-name my-eks-cluster \
  --addon-name vpc-cni \
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-Yfakrq0C1UTm \
  --addon-version v1.17.1-eksbuild.1 \
  --configuration-values 'file://configuration-values.json' \
  --resolve-conflicts PRESERVE

```

configuration-values.json の内容:

```

{
  "resources": {
    "limits": {
      "cpu": "100m"
    },
    "requests": {
      "cpu": "50m"
    }
  }
}

```

```

    }
  },
  "env": {
    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
  }
}

```

出力:

```

{
  "update": {
    "id": "6881a437-174f-346b-9a63-6e91763507cc",
    "status": "InProgress",
    "type": "AddonUpdate",
    "params": [
      {
        "type": "AddonVersion",
        "value": "v1.17.1-eksbuild.1"
      },
      {
        "type": "ServiceAccountRoleArn",
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-
addon-vpc-cni-Role1-Yfakrq0C1UTm"
      },
      {
        "type": "ResolveConflicts",
        "value": "PRESERVE"
      },
      {
        "type": "ConfigurationValues",
        "value": "{\n  \"resources\": {\n    \"limits\": {\n
      \"cpu\": \"100m\"\n    },\n    \"requests\": {\n      \"cpu\": \"50m
      }\n    },\n  \"env\": {\n    \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR
      }\n  }\n}"
      }
    ],
    "createdAt": "2024-04-12T16:22:55.519000-04:00",
    "errors": []
  }
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの更新」](#)を参照してください。 EKS

例 5。カスタムYAML設定値ファイルを使用して Amazon EKS アドオンを更新するには

次のコマンドupdate-addon例では、Amazon EKS アドオンをカスタムYAML設定値で更新し、競合の詳細を解決します。

```
aws eks update-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name vpc-cni \  
  --service-account-role-arn arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-  
addon-vpc-cni-Role1-Yfakrq0C1UTm \  
  --addon-version v1.18.0-eksbuild.1 \  
  --configuration-values 'file://configuration-values.yaml' \  
  --resolve-conflicts PRESERVE
```

configuration-values.yaml の内容:

```
resources:  
  limits:  
    cpu: '100m'  
  requests:  
    cpu: '50m'  
env:  
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

出力:

```
{  
  "update": {  
    "id": "a067a4c9-69d0-3769-ace9-d235c5b16701",  
    "status": "InProgress",  
    "type": "AddonUpdate",  
    "params": [  
      {  
        "type": "AddonVersion",  
        "value": "v1.18.0-eksbuild.1"  
      },  
      {  
        "type": "ServiceAccountRoleArn",  
        "value": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-  
addon-vpc-cni-Role1-Yfakrq0C1UTm"  
      },  
      {
```

```

        "type": "ResolveConflicts",
        "value": "PRESERVE"
      },
      {
        "type": "ConfigurationValues",
        "value": "resources:\n      limits:\n          cpu: '100m'\nrequests:\n      cpu: '50m'\nenv:\n      AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
      }
    ],
    "createdAt": "2024-04-12T16:25:07.212000-04:00",
    "errors": []
  }
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS アドオンの管理 - アドオンの更新」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateAddon](#)」の「」を参照してください。

update-cluster-config

次の例は、update-cluster-config を使用する方法を説明しています。

AWS CLI

クラスターエンドポイントアクセスを更新するには

この例では、クラスターを更新してエンドポイントのパブリックアクセスを無効にし、プライベートエンドポイントアクセスを有効にします。

コマンド:

```
aws eks update-cluster-config --name example \
--resources-vpc-config endpointPublicAccess=false,endpointPrivateAccess=true
```

出力:

```
{
  "update": {
    "id": "ec883c93-2e9e-407c-a22f-8f6fa6e67d4f",
    "status": "InProgress",
  }
}
```

```

    "type": "EndpointAccessUpdate",
    "params": [
      {
        "type": "EndpointPublicAccess",
        "value": "false"
      },
      {
        "type": "EndpointPrivateAccess",
        "value": "true"
      }
    ],
    "createdAt": 1565806986.506,
    "errors": []
  }
}

```

クラスターのログ記録を有効にするには

このコマンド例では、`example` という名前のクラスターのすべてのクラスターコントロールプレーンログ記録タイプを有効にします `example`。

コマンド:

```

aws eks update-cluster-config --name example \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'

```

出力:

```

{
  "update": {
    "id": "7551c64b-1d27-4b1e-9f8e-c45f056eb6fd",
    "status": "InProgress",
    "type": "LoggingUpdate",
    "params": [
      {
        "type": "ClusterLogging",
        "value": "{\"clusterLogging\":{\"types\":[\"api\",\"audit\",
\"authenticator\",\"controllerManager\",\"scheduler\"],\"enabled\":true}}"
      }
    ],
    "createdAt": 1565807210.37,
  }
}

```



```
    "errors": []
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateClusterConfig](#)」の「」を参照してください。

update-cluster-version

次のコード例は、update-cluster-version を使用する方法を示しています。

AWS CLI

my-eks-cluster 「」という名前の Amazon EKS クラスターを指定された Kubernetes バージョンに更新するには

次のupdate-cluster-version例では、Amazon EKS クラスターを指定された Kubernetes バージョンに更新します。

```
aws eks update-cluster-version \
  --name my-eks-cluster \
  --kubernetes-version 1.27
```

出力:

```
{
  "update": {
    "id": "e4091a28-ea14-48fd-a8c7-975aeb469e8a",
    "status": "InProgress",
    "type": "VersionUpdate",
    "params": [
      {
        "type": "Version",
        "value": "1.27"
      },
      {
        "type": "PlatformVersion",
        "value": "eks.16"
      }
    ],
    "createdAt": "2024-04-12T16:56:01.082000-04:00",
```

```
    "errors": []
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS クラスター Kubernetes バージョンの更新」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateClusterVersion](#)」の「」を参照してください。

update-kubeconfig

次のコード例は、update-kubeconfig を使用する方法を示しています。

AWS CLI

例 1: my-eks-cluster 「」という名前の Amazon EKS クラスターに接続できるように kubeconfig を作成または更新して kubectl を設定します。

次のupdate-kubeconfig例では、という名前の Amazon クラスターに接続できるように kubeconfig を作成または更新して EKS kubectl を設定します my-eks-cluster。

```
aws eks update-kubeconfig \
  --name my-eks-cluster
```

出力:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/xxx/.kube/config
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS クラスターの kubeconfig ファイルの作成または更新」](#)を参照してください。 EKS

例 2: my-eks-cluster 「」という名前の Amazon クラスターに接続できるように、kubeconfig (クラスター認証のロールを引き受ける role-arn オプションを使用) を作成または更新して kubectl EKS を設定します。

次のupdate-kubeconfig例では、という名前の Amazon クラスターに接続できるように、kubeconfig (クラスター認証のロールを引き受ける role-arn オプションを使用) を作成または更新することで EKS kubectl を設定します my-eks-cluster。

```
aws eks update-kubeconfig \  
  --name my-eks-cluster \  
  --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-  
cluster-ServiceRole-j1k7AfTIQtnM
```

出力:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/  
xxx/.kube/config
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS クラスターの kubeconfig ファイルの作成または更新」](#)を参照してください。 EKS

例 3: my-eks-cluster 「」という名前の Amazon クラスターに接続できるように、kubeconfig (カスタムクラスターエイリアスとユーザーエイリアスとともにクラスター認証のロールを引き受ける role-arn オプションを使用) を作成または更新して EKS、kubectl を設定します。

次の update-kubeconfig 例では、という名前の Amazon クラスターに接続できるように、kubeconfig (カスタムクラスターエイリアスとユーザーエイリアスとともにクラスター認証のロールを引き受ける role-arn オプションを使用) を作成または更新して EKS kubectl を設定します my-eks-cluster。

```
aws eks update-kubeconfig \  
  --name my-eks-cluster \  
  --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-  
cluster-ServiceRole-j1k7AfTIQtnM \  
  --alias stage-eks-cluster \  
  --user-alias john
```

出力:

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS クラスターの kubeconfig ファイルの作成または更新」](#)を参照してください。 EKS

例 4: レビュー用に kubeconfig ファイルエントリを印刷し、my-eks-cluster 「」という名前の Amazon EKS クラスターに接続できるように kubectl を設定します。

次のupdate-kubeconfig例では、という名前の Amazon クラスターに接続できるように、kubeconfig (カスタムクラスターエイリアスとユーザーエイリアスとともにクラスター認証のロールを引き受ける role-arn オプションを使用) を作成または更新して EKS kubectl を設定します my-eks-cluster。

```
aws eks update-kubeconfig \  
  --name my-eks-cluster \  
  --role-arn arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-  
cluster-ServiceRole-j1k7AfTIQtnM \  
  --alias stage-eks-cluster \  
  --user-alias john \  
  --verbose
```

出力:

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config  
Entries:  
  
context:  
cluster: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster  
user: john  
name: stage-eks-cluster  
  
name: john  
user:  
exec:  
  apiVersion: client.authentication.k8s.io/v1beta1  
  args:  
  - --region  
  - us-east-2  
  - eks  
  - get-token  
  - --cluster-name  
  - my-eks-cluster  
  - --output  
  - json  
  - --role  
  - arn:aws:iam::111122223333:role/eksctl-EKS-Linux-Cluster-v1-24-cluster-  
ServiceRole-j1k7AfTIQtnM  
  command: aws  
  
cluster:
```

```
certificate-authority-data: xxx_CA_DATA_xxx
server: https://DALSJ343KE23J3RN45653DSKJTT647TYD.y14.us-east-2.eks.amazonaws.com
name: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

詳細については、[「Amazon ユーザーガイド」の「Amazon EKS クラスターの kubeconfig ファイルの作成または更新」](#)を参照してください。 EKS

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateKubeconfig](#)」の「」を参照してください。

update-nodegroup-config

次のコード例は、update-nodegroup-config を使用する方法を示しています。

AWS CLI

例 1: マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードに新しいラベルとテイントを追加する

次の update-nodegroup-config 例では、マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードに新しいラベルとテイントを追加します。

```
aws eks update-nodegroup-config \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --labels 'add0rUpdateLabels={my-eks-nodegroup-label-1=value-1,my-eks-nodegroup-label-2=value-2}' \
  --taints 'add0rUpdateTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'
```

出力:

```
{
  "update": {
    "id": "e66d21d3-bd8b-3ad1-a5aa-b196dc08c7c1",
    "status": "InProgress",
    "type": "ConfigUpdate",
    "params": [
      {
        "type": "LabelsToAdd",
        "value": "{\"my-eks-nodegroup-label-2\":\"value-2\",\"my-eks-nodegroup-label-1\":\"value-1\"}"
      }
    ]
  }
}
```

```

    },
    {
      "type": "TaintsToAdd",
      "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\"},
\\\"key\":\"taint-key-1\"]"
    }
  ],
  "createdAt": "2024-04-08T12:05:19.161000-04:00",
  "errors": []
}
}

```

詳細については、「[Amazon EKSユーザーガイド](#)」の「[マネージドノードグループの更新](#)」を参照してください。

例 2: マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードのラベルとテイントを削除する

次の `update-nodegroup-config` 例では、マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードのラベルとテイントを削除します。

```

aws eks update-nodegroup-config \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --labels 'removeLabels=my-eks-nodegroup-label-1, my-eks-nodegroup-label-2' \
  --taints 'removeTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'

```

出力:

```

{
  "update": {
    "id": "67a08692-9e59-3ace-a916-13929f44cec3",
    "status": "InProgress",
    "type": "ConfigUpdate",
    "params": [
      {
        "type": "LabelsToRemove",
        "value": "[\"my-eks-nodegroup-label-1\",\"my-eks-nodegroup-label-2\"]"
      }
    ]
  },
  {

```

```

        "type": "TaintsToRemove",
        "value": "[{\"effect\":\"NO_EXECUTE\",\"value\":\"taint-value-1\"},
\\\"key\\\":\\\"taint-key-1\\\"}]"
    }
  ],
  "createdAt": "2024-04-08T12:17:31.817000-04:00",
  "errors": []
}
}

```

詳細については、「[Amazon EKSユーザーガイド](#)」の「[マネージドノードグループの更新](#)」を参照してください。

例 3: マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードのラベルとテイントを削除および追加する

次の `update-nodegroup-config` 例では、マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードのラベルとテイントを削除および追加します。

```

aws eks update-nodegroup-config \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --labels 'addOrUpdateLabels={my-eks-nodegroup-new-label-1=new-value-1,my-eks-nodegroup-new-label-2=new-value-2},removeLabels=my-eks-nodegroup-label-1, my-eks-nodegroup-label-2' \
  --taints 'addOrUpdateTaints=[{key=taint-new-key-1,value=taint-new-value-1,effect=PREFER_NO_SCHEDULE}],removeTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'

```

出力:

```

{
  "update": {
    "id": "4a9c8c45-6ac7-3115-be71-d6412a2339b7",
    "status": "InProgress",
    "type": "ConfigUpdate",
    "params": [
      {
        "type": "LabelsToAdd",
        "value": "{\"my-eks-nodegroup-new-label-1\":\"new-value-1\",\"my-eks-nodegroup-new-label-2\":\"new-value-2\"}"
      },
      {

```

```

        "type": "LabelsToRemove",
        "value": "[\"my-eks-nodegroup-label-1\", \"my-eks-nodegroup-
label-2\"]"
      },
      {
        "type": "TaintsToAdd",
        "value": "[{\"effect\": \"PREFER_NO_SCHEDULE\", \"value\": \"taint-new-
value-1\", \"key\": \"taint-new-key-1\"}]"
      },
      {
        "type": "TaintsToRemove",
        "value": "[{\"effect\": \"NO_EXECUTE\", \"value\": \"taint-value-1\",
\"key\": \"taint-key-1\"}]"
      }
    ],
    "createdAt": "2024-04-08T12:30:55.486000-04:00",
    "errors": []
  }
}

```

詳細については、「Amazon EKSユーザーガイド」の[「マネージドノードグループの更新」](#)を参照してください。

例 4: マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードの scaling-config と update-config を更新する

次の update-nodegroup-config 例では、マネージドノードグループを更新して、Amazon EKS クラスターの EKS ワーカーノードの scaling-config と update-config を更新します。

```

aws eks update-nodegroup-config \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --scaling-config minSize=1,maxSize=5,desiredSize=2 \
  --update-config maxUnavailable=2

```

出力:

```

{
  "update": {
    "id": "a977160f-59bf-3023-805d-c9826e460aea",
    "status": "InProgress",
    "type": "ConfigUpdate",
    "params": [

```



```
{
  {
    "type": "MinSize",
    "value": "1"
  },
  {
    "type": "MaxSize",
    "value": "5"
  },
  {
    "type": "DesiredSize",
    "value": "2"
  },
  {
    "type": "MaxUnavailable",
    "value": "2"
  }
],
"createdAt": "2024-04-08T12:35:17.036000-04:00",
"errors": []
}
}
```

詳細については、「[Amazon EKSユーザーガイド](#)」の「[マネージドノードグループの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateNodegroupConfig](#)」の「」を参照してください。

update-nodegroup-version

次のコード例は、update-nodegroup-version を使用する方法を示しています。

AWS CLI

例 1: Amazon EKS マネージドノードグループの Kubernetes バージョンまたはAMIバージョンを更新する

次のupdate-nodegroup-version例では、Amazon EKS マネージドノードグループの Kubernetes バージョンまたはAMIバージョンを、Kubernetes クラスターで利用可能な最新バージョンに更新します。

```
aws eks update-nodegroup-version \
```

```
--cluster-name my-eks-cluster \  
--nodegroup-name my-eks-nodegroup \  
--no-force
```

出力:

```
{  
  "update": {  
    "id": "a94ebfc3-6bf8-307a-89e6-7dbaa36421f7",  
    "status": "InProgress",  
    "type": "VersionUpdate",  
    "params": [  
      {  
        "type": "Version",  
        "value": "1.26"  
      },  
      {  
        "type": "ReleaseVersion",  
        "value": "1.26.12-20240329"  
      }  
    ],  
    "createdAt": "2024-04-08T13:16:00.724000-04:00",  
    "errors": []  
  }  
}
```

詳細については、「[Amazon EKSユーザーガイド](#)」の「[マネージドノードグループの更新](#)」を参照してください。

例 2: Amazon EKS マネージドノードグループの Kubernetes バージョンまたはAMIバージョンを更新する

次のupdate-nodegroup-version例では、Amazon EKS マネージドノードグループの Kubernetes バージョンまたはAMIバージョンを指定されたAMIリリースバージョンに更新します。

```
aws eks update-nodegroup-version \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup \  
  --kubernetes-version '1.26' \  
  --release-version '1.26.12-20240307' \  
  --no-force
```

出力:

```
{
  "update": {
    "id": "4db06fe1-088d-336b-bdcd-3fdb94995fb7",
    "status": "InProgress",
    "type": "VersionUpdate",
    "params": [
      {
        "type": "Version",
        "value": "1.26"
      },
      {
        "type": "ReleaseVersion",
        "value": "1.26.12-20240307"
      }
    ],
    "createdAt": "2024-04-08T13:13:58.595000-04:00",
    "errors": []
  }
}
```

詳細については、「Amazon EKSユーザーガイド」の「マネージドノードグループの更新」 - <https://docs.aws.amazon.com/eks/latest/userguide/update-managed-node-group.html>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateNodegroupVersion](#)」の「」を参照してください。

を使用した Elastic Beanstalk の例 AWS CLI

次のコード例は、Elastic Beanstalk AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

abort-environment-update

次のコード例は、`abort-environment-update` を使用する方法を示しています。

AWS CLI

デプロイを中止するには

次のコマンドは、`my-env` という名前の環境に対して実行中のアプリケーションバージョンデプロイを中止します。

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AbortEnvironmentUpdate](#)」の「」を参照してください。

check-dns-availability

次の例は、`check-dns-availability` を使用する方法を説明しています。

AWS CLI

の可用性を確認するには CNAME

次のコマンドは、サブドメイン `my-cname.elasticbeanstalk.com` の可用性をチェックします。

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

出力:

```
{
  "Available": true,
  "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CheckDnsAvailability](#)」の「」を参照してください。

create-application-version

次の例は、create-application-version を使用する方法を説明しています。

AWS CLI

新しいアプリケーションバージョンを作成するには

次のコマンドは、「」という名前のアプリケーションの新しいバージョン「v1MyApp」を作成します。

```
aws elasticbeanstalk create-application-version --application-name MyApp
--version-label v1 --description MyAppv1 --source-bundle S3Bucket="my-
bucket",S3Key="sample.war" --auto-create-application
```

オプションにより auto-create-application、アプリケーションがまだ存在しない場合、アプリケーションは自動的に作成されます。ソースバンドルは、Apache Tomcat サンプルアプリケーションを含む「my-bucket」という名前の s3 バケットに保存されている .war ファイルです。

出力:

```
{
  "ApplicationVersion": {
    "ApplicationName": "MyApp",
    "VersionLabel": "v1",
    "Description": "MyAppv1",
    "DateCreated": "2015-02-03T23:01:25.412Z",
    "DateUpdated": "2015-02-03T23:01:25.412Z",
    "SourceBundle": {
      "S3Bucket": "my-bucket",
      "S3Key": "sample.war"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateApplicationVersion](#)」の「」を参照してください。

create-application

次のコード例は、create-application を使用する方法を示しています。

AWS CLI

新しいアプリケーションを作成するには

次のコマンドは、MyApp「」という名前の新しいアプリケーションを作成します。

```
aws elasticbeanstalk create-application --application-name MyApp --description "my application"
```

create-application コマンドは、アプリケーションの名前と説明のみを設定します。アプリケーションのソースコードをアップロードするには、を使用してアプリケーションの初期バージョンを作成します create-application-version。には、1つのステップでアプリケーションとアプリケーションバージョンを作成できる auto-create-application オプション create-application-version もあります。

出力:

```
{
  "Application": {
    "ApplicationName": "MyApp",
    "ConfigurationTemplates": [],
    "DateUpdated": "2015-02-12T18:32:21.181Z",
    "Description": "my application",
    "DateCreated": "2015-02-12T18:32:21.181Z"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateApplication](#)」の「」を参照してください。

create-configuration-template

次のコード例は、create-configuration-template を使用する方法を示しています。

AWS CLI

設定テンプレートを作成するには

次のコマンドは、ID が の環境に適用される設定my-app-v1から という名前の設定テンプレートを作成しますe-rpqsewtp2j。

```
aws elasticbeanstalk create-configuration-template --application-name my-app --  
template-name my-app-v1 --environment-id e-rpqsewtp2j
```

出力:

```
{  
  "ApplicationName": "my-app",  
  "TemplateName": "my-app-v1",  
  "DateCreated": "2015-08-12T18:40:39Z",  
  "DateUpdated": "2015-08-12T18:40:39Z",  
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConfigurationTemplate](#)」の「」を参照してください。

create-environment

次の例は、create-environment を使用する方法を説明しています。

AWS CLI

アプリケーションの新しい環境を作成するには

次のコマンドは、「my-app」という名前の java アプリケーションのバージョン「v1」用の新しい環境を作成します。

```
aws elasticbeanstalk create-environment --application-name my-app --environment-  
name my-env --cname-prefix my-app --version-label v1 --solution-stack-name "64bit  
Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
```

出力:

```
{  
  "ApplicationName": "my-app",  
  "EnvironmentName": "my-env",
```

```
"VersionLabel": "v1",
"Status": "Launching",
"EnvironmentId": "e-izqpassy4h",
"SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
"CNAME": "my-app.elasticbeanstalk.com",
"Health": "Grey",
"Tier": {
  "Type": "Standard",
  "Name": "WebServer",
  "Version": " "
},
"DateUpdated": "2015-02-03T23:04:54.479Z",
"DateCreated": "2015-02-03T23:04:54.479Z"
}
```

v1 は、 で以前にアップロードされたアプリケーションバージョンのラベルです create-application-version。

環境設定オプションを定義するJSONファイルを指定するには

次のcreate-environmentコマンドは、 という名前のJSONファイルを使用して、ソリューションスタックまたは設定テンプレートから取得した値を上書きmyoptions.jsonするように指定します。

```
aws elasticbeanstalk create-environment --environment-name sample-env --application-name sampleapp --option-settings file://myoptions.json
```

myoptions.json は、いくつかの設定を定義するJSONオブジェクトです。

```
[
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Interval",
    "Value": "15"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Timeout",
    "Value": "8"
  },
  {
    "Namespace": "aws:elb:healthcheck",
```



```
"OptionName": "HealthyThreshold",
  "Value": "2"
},
{
  "Namespace": "aws:elb:healthcheck",
  "OptionName": "UnhealthyThreshold",
  "Value": "3"
}
]
```

詳細については、AWS Elastic Beanstalk デベロッパーガイドの「オプション値」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEnvironment](#)」の「」を参照してください。

create-storage-location

次のコード例は、create-storage-location を使用する方法を示しています。

AWS CLI

ストレージロケーションを作成するには

次のコマンドは、Amazon S3 にストレージロケーションを作成します。

```
aws elasticbeanstalk create-storage-location
```

出力:

```
{
  "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStorageLocation](#)」の「」を参照してください。

delete-application-version

次のコード例は、delete-application-version を使用する方法を示しています。

AWS CLI

アプリケーションバージョンを削除するには

次のコマンドは、 という名前のアプリケーション22a0-stage-150819_182129に という名前のアプリケーションバージョンを削除しますmy-app。

```
aws elasticbeanstalk delete-application-version --version-label 22a0-stage-150819_182129 --application-name my-app
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApplicationVersion](#)」の「」を参照してください。

delete-application

次のコード例は、delete-application を使用する方法を示しています。

AWS CLI

アプリケーションを削除するには

次のコマンドは、 という名前のアプリケーションを削除しますmy-app。

```
aws elasticbeanstalk delete-application --application-name my-app
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApplication](#)」の「」を参照してください。

delete-configuration-template

次のコード例は、delete-configuration-template を使用する方法を示しています。

AWS CLI

設定テンプレートを削除するには

次のコマンドは、 という名前のアプリケーションの my-template という名前の設定テンプレートを削除しますmy-app。

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --application-name my-app
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConfigurationTemplate](#)」の「」を参照してください。

delete-environment-configuration

次の例は、delete-environment-configuration を使用する方法を説明しています。

AWS CLI

ドラフト設定を削除するには

次のコマンドは、 という名前の環境のドラフト設定を削除しますmy-env。

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --application-name my-app
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEnvironmentConfiguration](#)」の「」を参照してください。

describe-application-versions

次のコード例は、describe-application-versions を使用する方法を示しています。

AWS CLI

アプリケーションバージョンに関する情報を表示するには

次のコマンドは、 というラベルのアプリケーションバージョンに関する情報を取得しますv2。

```
aws elasticbeanstalk describe-application-versions --application-name my-app --version-label "v2"
```

出力:

```
{
  "ApplicationVersions": [
    {
      "ApplicationName": "my-app",
      "VersionLabel": "v2",
      "Description": "update cover page",
```

```
    "DateCreated": "2015-07-23T01:32:26.079Z",
    "DateUpdated": "2015-07-23T01:32:26.079Z",
    "SourceBundle": {
      "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",
      "S3Key": "my-app/5026-stage-150723_224258.war"
    }
  },
  {
    "ApplicationName": "my-app",
    "VersionLabel": "v1",
    "Description": "initial version",
    "DateCreated": "2015-07-23T22:26:10.816Z",
    "DateUpdated": "2015-07-23T22:26:10.816Z",
    "SourceBundle": {
      "S3Bucket": "elasticbeanstalk-us-west-2-015321684451",
      "S3Key": "my-app/5026-stage-150723_222618.war"
    }
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeApplicationVersions](#)」の「」を参照してください。

describe-applications

次の例は、describe-applications を使用する方法を説明しています。

AWS CLI

アプリケーションのリストを表示するには

次のコマンドは、現在のリージョンのアプリケーションに関する情報を取得します。

```
aws elasticbeanstalk describe-applications
```

出力:

```
{
  "Applications": [
    {
      "ApplicationName": "ruby",
```

```

    "ConfigurationTemplates": [],
    "DateUpdated": "2015-08-13T21:05:44.376Z",
    "Versions": [
      "Sample Application"
    ],
    "DateCreated": "2015-08-13T21:05:44.376Z"
  },
  {
    "ApplicationName": "pythonsample",
    "Description": "Application created from the EB CLI using \"eb init\"",
    "Versions": [
      "Sample Application"
    ],
    "DateCreated": "2015-08-13T19:05:43.637Z",
    "ConfigurationTemplates": [],
    "DateUpdated": "2015-08-13T19:05:43.637Z"
  },
  {
    "ApplicationName": "nodejs-example",
    "ConfigurationTemplates": [],
    "DateUpdated": "2015-08-06T17:50:02.486Z",
    "Versions": [
      "add elasticache",
      "First Release"
    ],
    "DateCreated": "2015-08-06T17:50:02.486Z"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeApplications](#)」の「」を参照してください。

describe-configuration-options

次の例は、describe-configuration-options を使用する方法を説明しています。

AWS CLI

環境の設定オプションを表示するには

次のコマンドは、という名前の環境で使用できるすべての設定オプションの説明を取得しますmy-env。

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --  
application-name my-app
```

出力 (省略):

```
{  
  "Options": [  
    {  
      "Name": "JVMOptions",  
      "UserDefined": false,  
      "DefaultValue": "Xms=256m,Xmx=256m,XX:MaxPermSize=64m,JVM Options=",  
      "ChangeSeverity": "RestartApplicationServer",  
      "Namespace": "aws:cloudformation:template:parameter",  
      "ValueType": "KeyValueList"  
    },  
    {  
      "Name": "Interval",  
      "UserDefined": false,  
      "DefaultValue": "30",  
      "ChangeSeverity": "NoInterruption",  
      "Namespace": "aws:elb:healthcheck",  
      "MaxValue": 300,  
      "MinValue": 5,  
      "ValueType": "Scalar"  
    },  
    ...  
    {  
      "Name": "LowerThreshold",  
      "UserDefined": false,  
      "DefaultValue": "2000000",  
      "ChangeSeverity": "NoInterruption",  
      "Namespace": "aws:autoscaling:trigger",  
      "MinValue": 0,  
      "ValueType": "Scalar"  
    },  
    {  
      "Name": "ListenerEnabled",  
      "UserDefined": false,  
      "DefaultValue": "true",  
      "ChangeSeverity": "Unknown",  
      "Namespace": "aws:elb:listener",  
      "ValueType": "Boolean"  
    }  
  ]  
}
```

```
]
}
```

使用可能な設定オプションは、プラットフォームと設定バージョンによって異なります。名前空間とサポートされているオプションの詳細については、AWS Elastic Beanstalk デベロッパーガイドの「オプション値」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeConfigurationOptions](#)」の「」を参照してください。

describe-configuration-settings

次のコード例は、describe-configuration-settings を使用する方法を示しています。

AWS CLI

環境の設定を表示するには

次のコマンドは、 という名前の環境の設定を取得しますmy-env。

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --
application-name my-app
```

出力 (省略):

```
{
  "ConfigurationSettings": [
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Description": "Environment created from the EB CLI using \"eb create
      \",
      "DeploymentStatus": "deployed",
      "DateCreated": "2015-08-13T19:16:25Z",
      "OptionSettings": [
        {
          "OptionName": "Availability Zones",
          "ResourceName": "AWSEBAutoScalingGroup",
          "Namespace": "aws:autoscaling:asg",
          "Value": "Any"
        },
        {
          "OptionName": "Cooldown",
```

```

        "ResourceName": "AWSEBAutoScalingGroup",
        "Namespace": "aws:autoscaling:asg",
        "Value": "360"
    },
    ...
    {
        "OptionName": "ConnectionDrainingTimeout",
        "ResourceName": "AWSEBLoadBalancer",
        "Namespace": "aws:elb:policies",
        "Value": "20"
    },
    {
        "OptionName": "ConnectionSettingIdleTimeout",
        "ResourceName": "AWSEBLoadBalancer",
        "Namespace": "aws:elb:policies",
        "Value": "60"
    }
],
    "DateUpdated": "2015-08-13T23:30:07Z",
    "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
Java 8"
    }
]
}

```

名前空間とサポートされているオプションの詳細については、AWS Elastic Beanstalk デベロッパーガイドの「オプション値」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeConfigurationSettings](#)」の「」を参照してください。

describe-environment-health

次のコード例は、describe-environment-health を使用する方法を示しています。

AWS CLI

環境のヘルスを表示するには

次のコマンドは、 という名前の環境の全体的なヘルス情報を取得します my-env。

```
aws elasticbeanstalk describe-environment-health --environment-name my-env --
attribute-names ALL
```


出力:

```
{
  "Status": "Ready",
  "EnvironmentName": "my-env",
  "Color": "Green",
  "ApplicationMetrics": {
    "Duration": 10,
    "Latency": {
      "P99": 0.004,
      "P75": 0.002,
      "P90": 0.003,
      "P95": 0.004,
      "P85": 0.003,
      "P10": 0.001,
      "P999": 0.004,
      "P50": 0.001
    },
    "RequestCount": 45,
    "StatusCodes": {
      "Status3xx": 0,
      "Status2xx": 45,
      "Status5xx": 0,
      "Status4xx": 0
    }
  },
  "RefreshedAt": "2015-08-20T21:09:18Z",
  "HealthStatus": "Ok",
  "InstancesHealth": {
    "Info": 0,
    "Ok": 1,
    "Unknown": 0,
    "Severe": 0,
    "Warning": 0,
    "Degraded": 0,
    "NoData": 0,
    "Pending": 0
  },
  "Causes": []
}
```

ヘルス情報は、拡張ヘルスレポートが有効になっている環境でのみ使用できます。詳細については、AWS Elastic Beanstalk デベロッパーガイドの「拡張ヘルスレポートとモニタリング」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEnvironmentHealth](#)」の「」を参照してください。

describe-environment-resources

次の例は、describe-environment-resources を使用する方法を説明しています。

AWS CLI

環境内の AWS リソースに関する情報を表示するには

次のコマンドは、`my-env` という名前の環境内のリソースに関する情報を取得します。

```
aws elasticbeanstalk describe-environment-resources --environment-name my-env
```

出力:

```
{
  "EnvironmentResources": {
    "EnvironmentName": "my-env",
    "AutoScalingGroups": [
      {
        "Name": "awseb-e-qu3fyyjyjs-stack-AWSEBAutoScalingGroup-
QSB2Z088SXZT"
      }
    ],
    "Triggers": [],
    "LoadBalancers": [
      {
        "Name": "awseb-e-q-AWSEBLoa-1EEPZ0K98BIF0"
      }
    ],
    "Queues": [],
    "Instances": [
      {
        "Id": "i-0c91c786"
      }
    ],
    "LaunchConfigurations": [
```

```
{
  "Name": "awseb-e-qu3fyyjyjs-stack-
  AWSEBAutoScalingLaunchConfiguration-1UUVQIBC96TQ2"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEnvironmentResources](#)」の「」を参照してください。

describe-environments

次のコード例は、describe-environments を使用する方法を示しています。

AWS CLI

環境に関する情報を表示するには

次のコマンドは、 という名前の環境に関する情報を取得しますmy-env。

```
aws elasticbeanstalk describe-environments --environment-names my-env
```

出力:

```
{
  "Environments": [
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "VersionLabel": "7f58-stage-150812_025409",
      "Status": "Ready",
      "EnvironmentId": "e-rpqsewtp2j",
      "EndpointURL": "awseb-e-w-AWSEBLoa-1483140XB0Q4L-109QXY8121.us-
      west-2.elb.amazonaws.com",
      "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8
      Java 8",
      "CNAME": "my-env.elasticbeanstalk.com",
      "Health": "Green",
      "AbortableOperationInProgress": false,
      "Tier": {
        "Version": " ",

```

```
        "Type": "Standard",
        "Name": "WebServer"
    },
    "DateUpdated": "2015-08-12T18:16:55.019Z",
    "DateCreated": "2015-08-07T20:48:49.599Z"
}
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEnvironments](#)」の「」を参照してください。

describe-events

次の例は、describe-events を使用する方法を説明しています。

AWS CLI

環境のイベントを表示するには

次のコマンドは、 という名前の環境のイベントを取得しますmy-env。

```
aws elasticbeanstalk describe-events --environment-name my-env
```

出力 (省略):

```
{
  "Events": [
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Message": "Environment health has transitioned from Info to Ok.",
      "EventDate": "2015-08-20T07:06:53.535Z",
      "Severity": "INFO"
    },
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Severity": "INFO",
      "RequestId": "b7f3960b-4709-11e5-ba1e-07e16200da41",
      "Message": "Environment update completed successfully.",
      "EventDate": "2015-08-20T07:06:02.049Z"
    }
  ]
}
```

```
    },
    ...
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Severity": "INFO",
      "RequestId": "ca8dfbf6-41ef-11e5-988b-651aa638f46b",
      "Message": "Using elasticbeanstalk-us-west-2-012445113685 as Amazon S3
storage bucket for environment data.",
      "EventDate": "2015-08-13T19:16:27.561Z"
    },
    {
      "ApplicationName": "my-app",
      "EnvironmentName": "my-env",
      "Severity": "INFO",
      "RequestId": "cdfba8f6-41ef-11e5-988b-65638f41aa6b",
      "Message": "createEnvironment is starting.",
      "EventDate": "2015-08-13T19:16:26.581Z"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

describe-instances-health

次のコード例は、describe-instances-health を使用する方法を示しています。

AWS CLI

環境のヘルスを表示するには

次のコマンドは、 という名前の環境内のインスタンスのヘルス情報を取得しますmy-env。

```
aws elasticbeanstalk describe-instances-health --environment-name my-env --
attribute-names ALL
```

出力:

```
{
  "InstanceHealthList": [
```

```
{
  "InstanceId": "i-08691cc7",
  "ApplicationMetrics": {
    "Duration": 10,
    "Latency": {
      "P99": 0.006,
      "P75": 0.002,
      "P90": 0.004,
      "P95": 0.005,
      "P85": 0.003,
      "P10": 0.0,
      "P999": 0.006,
      "P50": 0.001
    },
    "RequestCount": 48,
    "StatusCodes": {
      "Status3xx": 0,
      "Status2xx": 47,
      "Status5xx": 0,
      "Status4xx": 1
    }
  },
  "System": {
    "LoadAverage": [
      0.0,
      0.02,
      0.05
    ],
    "CPUUtilization": {
      "SoftIRQ": 0.1,
      "IOWait": 0.2,
      "System": 0.3,
      "Idle": 97.8,
      "User": 1.5,
      "IRQ": 0.0,
      "Nice": 0.1
    }
  },
  "Color": "Green",
  "HealthStatus": "Ok",
  "LaunchedAt": "2015-08-13T19:17:09Z",
  "Causes": []
},
],
```

```
"RefreshedAt": "2015-08-20T21:09:08Z"  
}
```

ヘルス情報は、拡張ヘルスレポートが有効になっている環境でのみ使用できます。詳細については、AWS Elastic Beanstalk デベロッパーガイドの「拡張ヘルスレポートとモニタリング」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstancesHealth](#)」の「」を参照してください。

list-available-solution-stacks

次のコード例は、list-available-solution-stacks を使用する方法を示しています。

AWS CLI

ソリューションスタックを表示するには

次のコマンドは、現在利用可能なすべてのプラットフォーム設定と、過去に使用したことのあるすべてのソリューションスタックを一覧表示します。

```
aws elasticbeanstalk list-available-solution-stacks
```

出力 (省略):

```
{  
  "SolutionStacks": [  
    "64bit Amazon Linux 2015.03 v2.0.0 running Node.js",  
    "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.6",  
    "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.5",  
    "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.4",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Python 3.4",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Python 2.7",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Python",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Puma)",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Passenger Standalone)",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Puma)",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Passenger Standalone)",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Puma)",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Passenger Standalone)",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 1.9.3",  
    "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",  
  ]  
}
```

```

    "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 7",
    "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 6",
    "64bit Windows Server Core 2012 R2 running IIS 8.5",
    "64bit Windows Server 2012 R2 running IIS 8.5",
    "64bit Windows Server 2012 running IIS 8",
    "64bit Windows Server 2008 R2 running IIS 7.5",
    "64bit Amazon Linux 2015.03 v2.0.0 running Docker 1.6.2",
    "64bit Amazon Linux 2015.03 v2.0.0 running Multi-container Docker 1.6.2
(Generic)",
    "64bit Debian jessie v2.0.0 running GlassFish 4.1 Java 8 (Preconfigured -
Docker)",
    "64bit Debian jessie v2.0.0 running GlassFish 4.0 Java 7 (Preconfigured -
Docker)",
    "64bit Debian jessie v2.0.0 running Go 1.4 (Preconfigured - Docker)",
    "64bit Debian jessie v2.0.0 running Go 1.3 (Preconfigured - Docker)",
    "64bit Debian jessie v2.0.0 running Python 3.4 (Preconfigured - Docker)",
  ],
  "SolutionStackDetails": [
    {
      "PermittedFileTypes": [
        "zip"
      ],
      "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Node.js"
    },
    ...
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAvailableSolutionStacks](#)」の「」を参照してください。

rebuild-environment

次のコード例は、rebuild-environment を使用する方法を示しています。

AWS CLI

環境を再構築するには

次のコマンドは、 という名前の環境でリソースを終了して再作成します `my-env`。

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```


- API 詳細については、AWS CLI 「コマンドリファレンス[RebuildEnvironment](#)」の「」を参照してください。

request-environment-info

次の例は、request-environment-info を使用する方法を説明しています。

AWS CLI

テールログをリクエストするには

次のコマンドは、 という名前の環境からログをリクエストしますmy-env。

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-type tail
```

ログをリクエストしたら、 を使用してその場所を取得します retrieve-environment-info。

- API 詳細については、AWS CLI 「コマンドリファレンス[RequestEnvironmentInfo](#)」の「」を参照してください。

restart-app-server

次の例は、restart-app-server を使用する方法を説明しています。

AWS CLI

アプリケーションサーバーを再起動するには

次のコマンドは、 という環境内のすべてのインスタンスでアプリケーションサーバーを再起動しますmy-env。

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RestartAppServer](#)」の「」を参照してください。

retrieve-environment-info

次の例は、retrieve-environment-info を使用する方法を説明しています。

AWS CLI

テールログを取得するには

次のコマンドは、 という名前の環境からログへのリンクを取得しますmy-env。

```
aws elasticbeanstalk retrieve-environment-info --environment-name my-env --info-type tail
```

出力:

```
{
  "EnvironmentInfo": [
    {
      "SampleTimestamp": "2015-08-20T22:23:17.703Z",
      "Message": "https://elasticbeanstalk-us-west-2-0123456789012.s3.amazonaws.com/resources/environments/logs/tail/e-fyqyju3yjs/i-09c1c867/TailLogs-1440109397703.out?AWSAccessKeyId=AKGPT4J56IAJ2EUBL5CQ&Expires=1440195891&Signature=n%2BEa10V6A2HI0x4Rcfb7LT16bBM%3D",
      "InfoType": "tail",
      "Ec2InstanceId": "i-09c1c867"
    }
  ]
}
```

ブラウザでリンクを表示します。取得する前に、 でログをリクエストする必要がありますrequest-environment-info。

- API 詳細については、AWS CLI 「コマンドリファレンス[RetrieveEnvironmentInfo](#)」の「」を参照してください。

swap-environment-cnames

次のコード例は、swap-environment-cnames を使用する方法を示しています。

AWS CLI

環境をスワップするには CNAMEs

次のコマンドは、2つの環境の割り当てられたサブドメインをスワップします。

```
aws elasticbeanstalk swap-environment-cnames --source-environment-name my-env-blue
--destination-environment-name my-env-green
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SwapEnvironmentCnames](#)」の「」を参照してください。

terminate-environment

次の例は、terminate-environment を使用する方法を説明しています。

AWS CLI

環境を終了するには

次のコマンドは、 という名前の Elastic Beanstalk 環境を終了しますmy-env。

```
aws elasticbeanstalk terminate-environment --environment-name my-env
```

出力:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "Status": "Terminating",
  "EnvironmentId": "e-fh2eravpns",
  "EndpointURL": "awseb-e-f-AWSEBLoa-1I9XUMP4-8492WNUP202574.us-
west-2.elb.amazonaws.com",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java
8",
  "CNAME": "my-env.elasticbeanstalk.com",
  "Health": "Grey",
  "AbortableOperationInProgress": false,
  "Tier": {
    "Version": " ",
    "Type": "Standard",
    "Name": "WebServer"
  },
  "DateUpdated": "2015-08-12T19:05:54.744Z",
  "DateCreated": "2015-08-12T18:52:53.622Z"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[TerminateEnvironment](#)」の「」を参照してください。

update-application-version

次のコード例は、update-application-version を使用する方法を示しています。

AWS CLI

アプリケーションバージョンの説明を変更するには

次のコマンドは、という名前のアプリケーションバージョンの説明を更新します22a0-stage-150819_185942。

```
aws elasticbeanstalk update-application-version --version-label 22a0-stage-150819_185942 --application-name my-app --description "new description"
```

出力:

```
{
  "ApplicationVersion": {
    "ApplicationName": "my-app",
    "VersionLabel": "22a0-stage-150819_185942",
    "Description": "new description",
    "DateCreated": "2015-08-19T18:59:17.646Z",
    "DateUpdated": "2015-08-20T22:53:28.871Z",
    "SourceBundle": {
      "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012",
      "S3Key": "my-app/22a0-stage-150819_185942.war"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateApplicationVersion](#)」の「」を参照してください。

update-application

次の例は、update-application を使用する方法を説明しています。

AWS CLI

アプリケーションの説明を変更するには

次のコマンドは、 という名前のアプリケーションの説明を更新しますmy-app。

```
aws elasticbeanstalk update-application --application-name my-app --description "my Elastic Beanstalk application"
```

出力:

```
{
  "Application": {
    "ApplicationName": "my-app",
    "Description": "my Elastic Beanstalk application",
    "Versions": [
      "2fba-stage-150819_234450",
      "bf07-stage-150820_214945",
      "93f8",
      "fd7c-stage-150820_000431",
      "22a0-stage-150819_185942"
    ],
    "DateCreated": "2015-08-13T19:15:50.449Z",
    "ConfigurationTemplates": [],
    "DateUpdated": "2015-08-20T22:34:56.195Z"
  }
}
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[UpdateApplication](#)」の「」を参照してください。

update-configuration-template

次のコード例は、update-configuration-template を使用する方法を示しています。

AWS CLI

設定テンプレートを更新するには

次のコマンドは、 という名前の保存済み設定テンプレートConfigDocumentから、設定されたCloudWatch カスタムヘルスマトリクス設定を削除しますmy-template。

```
aws elasticbeanstalk update-configuration-template --template-name my-template --application-name my-app --options-to-remove Namespace=aws:elasticbeanstalk:healthreporting:system,OptionName=ConfigDocument
```

出力:

```
{
  "ApplicationName": "my-app",
  "TemplateName": "my-template",
  "DateCreated": "2015-08-20T22:39:31Z",
  "DateUpdated": "2015-08-20T22:43:11Z",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8"
}
```

名前空間とサポートされているオプションの詳細については、AWS Elastic Beanstalk デベロッパーガイドの「オプション値」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateConfigurationTemplate](#)」の「」を参照してください。

update-environment

次の例は、update-environment を使用する方法を説明しています。

AWS CLI

環境を新しいバージョンに更新するには

次のコマンドは、「my-env」という名前の環境を、それが属するアプリケーションのバージョン「v2」に更新します。

```
aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

このコマンドでは、「my-env」環境がすでに存在し、「v2」というラベルが付いた有効なアプリケーションバージョンを持つアプリケーションに属している必要があります。

出力:

```
{
  "ApplicationName": "my-app",
```

```

"EnvironmentName": "my-env",
"VersionLabel": "v2",
"Status": "Updating",
"EnvironmentId": "e-szqipays4h",
"EndpointURL": "awseb-e-i-AWSEBLoa-1RD LX6TC9VUA0-0123456789.us-
west-2.elb.amazonaws.com",
"SolutionStackName": "64bit Amazon Linux running Tomcat 7",
"CNAME": "my-env.elasticbeanstalk.com",
"Health": "Grey",
"Tier": {
  "Version": " ",
  "Type": "Standard",
  "Name": "WebServer"
},
"DateUpdated": "2015-02-03T23:12:29.119Z",
"DateCreated": "2015-02-03T23:04:54.453Z"
}

```

環境変数を設定するには

次のコマンドは、「my-envPARAM1」環境の「」変数の値をParamValue「」に設定します。

```

aws elasticbeanstalk update-environment --environment-name my-env --option-
settings Namespace=aws:elasticbeanstalk:application:environment,OptionName=PARAM1,Value=Para

```

option-settings パラメータは、変数の名前と値に加えて名前空間を取得します。Elastic Beanstalk は、環境変数に加えて、オプション用のいくつかの名前空間をサポートしています。

ファイルからオプション設定を設定するには

次のコマンドは、ファイルからaws:elb:loadbalancer名前空間内のいくつかのオプションを設定します。

```

aws elasticbeanstalk update-environment --environment-name my-env --option-
settings file://options.json

```

options.json は、いくつかの設定を定義するJSONオブジェクトです。

```

[
  {
    "Namespace": "aws:elb:healthcheck",

```

```
    "OptionName": "Interval",
    "Value": "15"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "Timeout",
    "Value": "8"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "HealthyThreshold",
    "Value": "2"
  },
  {
    "Namespace": "aws:elb:healthcheck",
    "OptionName": "UnhealthyThreshold",
    "Value": "3"
  }
]
```

出力:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "7f58-stage-150812_025409",
  "Status": "Updating",
  "EnvironmentId": "e-wtp2rqpqsej",
  "EndpointURL": "awseb-e-w-AWSEBLoa-14XB83101Q4L-104QXY80921.sa-
east-1.elb.amazonaws.com",
  "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java
8",
  "CNAME": "my-env.elasticbeanstalk.com",
  "Health": "Grey",
  "AbortableOperationInProgress": true,
  "Tier": {
    "Version": " ",
    "Type": "Standard",
    "Name": "WebServer"
  },
  "DateUpdated": "2015-08-12T18:15:23.804Z",
  "DateCreated": "2015-08-07T20:48:49.599Z"
}
```


名前空間とサポートされているオプションの詳細については、AWS Elastic Beanstalk デベロッパーガイドの「オプション値」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateEnvironment](#)」の「」を参照してください。

validate-configuration-settings

次の例は、validate-configuration-settings を使用する方法を説明しています。

AWS CLI

設定を検証するには

次のコマンドは、CloudWatch カスタムメトリクス設定ドキュメントを検証します。

```
aws elasticbeanstalk validate-configuration-settings --application-name my-app --environment-name my-env --option-settings file://options.json
```

options.json は、検証する 1 つ以上の設定を含む JSON ドキュメントです。

```
[
  {
    "Namespace": "aws:elasticbeanstalk:healthreporting:system",
    "OptionName": "ConfigDocument",
    "Value": "{\"CloudWatchMetrics\": {\"Environment\":
  {\"ApplicationLatencyP99.9\": null,\"InstancesSevere\": 60,
  \"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": null,
  \"ApplicationLatencyP95\": 60,\"InstancesUnknown\": 60,\"ApplicationLatencyP85\":
  60,\"InstancesInfo\": null,\"ApplicationRequests2xx\": null,\"InstancesDegraded
  \": null,\"InstancesWarning\": 60,\"ApplicationLatencyP50\": 60,
  \"ApplicationRequestsTotal\": null,\"InstancesNoData\": null,\"InstancesPending
  \": 60,\"ApplicationLatencyP10\": null,\"ApplicationRequests5xx\": null,
  \"ApplicationLatencyP75\": null,\"Instances0k\": 60,\"ApplicationRequests3xx\":
  null,\"ApplicationRequests4xx\": null},\"Instance\": {\"ApplicationLatencyP99.9\":
  null,\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": null,
  \"ApplicationLatencyP95\": null,\"ApplicationLatencyP85\": null,\"CPUUser\": 60,
  \"ApplicationRequests2xx\": null,\"CPUIdle\": null,\"ApplicationLatencyP50\":
  null,\"ApplicationRequestsTotal\": 60,\"RootFilesystemUtil\": null,
  \"LoadAverage1min\": null,\"CPUIdle\": null,\"CPUNice\": 60,\"CPUiowait\": 60,
  \"ApplicationLatencyP10\": null,\"LoadAverage5min\": null,\"ApplicationRequests5xx
  \": null,\"ApplicationLatencyP75\": 60,\"CPUSystem\": 60,\"ApplicationRequests3xx\":
```

```
60,\"ApplicationRequests4xx\": null,\"InstanceHealth\": null,\"CPUSoftirq\": 60}},
\"Version\": 1}"
  }
]
```

指定したオプションが指定された環境に対して有効な場合、Elastic Beanstalk は空の Messages 配列を返します。

```
{
  "Messages": []
}
```

検証が失敗すると、レスポンスにはエラーに関する情報が含まれます。

```
{
  "Messages": [
    {
      "OptionName": "ConfigDocumet",
      "Message": "Invalid option specification (Namespace:
'aws:elasticbeanstalk:healthreporting:system', OptionName: 'ConfigDocumet'):
Unknown configuration setting.",
      "Namespace": "aws:elasticbeanstalk:healthreporting:system",
      "Severity": "error"
    }
  ]
}
```

名前空間とサポートされているオプションの詳細については、AWS Elastic Beanstalk デベロッパーガイドの「オプション値」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ValidateConfigurationSettings](#)」の「」を参照してください。

Elastic Load Balancing - を使用したバージョン 1 の例 AWS CLI

次のコード例は、Elastic Load Balancing - バージョン 1 AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags

次の例は、add-tags を使用する方法を説明しています。

AWS CLI

ロードバランサーにタグを追加するには

この例では、指定されたロードバランサーにタグを追加します。

コマンド:

```
aws elb add-tags --load-balancer-name my-load-balancer --  
tags "Key=project,Value=Lima" "Key=department,Value=digital-media"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTags](#)」の「」を参照してください。

apply-security-groups-to-load-balancer

次のコード例は、apply-security-groups-to-load-balancer を使用する方法を示しています。

AWS CLI

セキュリティグループを のロードバランサーに関連付けるには VPC

この例では、セキュリティグループを 内の指定されたロードバランサーに関連付けますVPC。

コマンド:

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer  
--security-groups sg-fc448899
```

出力:

```
{
  "SecurityGroups": [
    "sg-fc448899"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ApplySecurityGroupsToLoadBalancer](#)」の「」を参照してください。

attach-load-balancer-to-subnets

次のコード例は、attach-load-balancer-to-subnets を使用する方法を示しています。

AWS CLI

サブネットをロードバランサーにアタッチするには

この例では、指定されたロードバランサー用に設定されたサブネットのセットに、指定されたサブネットを追加します。

コマンド:

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

出力:

```
{
  "Subnets": [
    "subnet-15aaab61",
    "subnet-0ecac448"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachLoadBalancerToSubnets](#)」の「」を参照してください。

configure-health-check

次のコード例は、configure-health-check を使用する方法を示しています。

AWS CLI

バックエンドEC2インスタンスのヘルスチェック設定を指定するには

この例では、バックエンドEC2インスタンスのヘルスを評価するために使用されるヘルスチェック設定を指定します。

コマンド:

```
aws elb configure-health-check --load-balancer-name my-load-balancer --health-check Target=HTTP:80/png,Interval=30,UnhealthyThreshold=2,HealthyThreshold=2,Timeout=3
```

出力:

```
{
  "HealthCheck": {
    "HealthyThreshold": 2,
    "Interval": 30,
    "Target": "HTTP:80/png",
    "Timeout": 3,
    "UnhealthyThreshold": 2
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfigureHealthCheck](#)」の「」を参照してください。

create-app-cookie-stickiness-policy

次の例は、create-app-cookie-stickiness-policy を使用する方法を説明しています。

AWS CLI

HTTPSロードバランサーのスティッキーポリシーを生成するには

この例では、アプリケーション生成 Cookie のスティッキーセッションの存続期間に従うスティッキーポリシーを生成します。

コマンド:

```
aws elb create-app-cookie-stickness-policy --load-balancer-name my-load-balancer --  
policy-name my-app-cookie-policy --cookie-name my-app-cookie
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAppCookieStickinessPolicy](#)」の「」を参照してください。

create-lb-cookie-stickness-policy

次の例は、create-lb-cookie-stickness-policy を使用する方法を説明しています。

AWS CLI

HTTPSロードバランサーの期間ベースのスティビネスポリシーを生成するには

この例では、指定された有効期限によって制御されるスティッキーセッションの有効期間を持つスティッキーポリシーを生成します。

コマンド:

```
aws elb create-lb-cookie-stickness-policy --load-balancer-name my-load-balancer --  
policy-name my-duration-cookie-policy --cookie-expiration-period 60
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLbCookieStickinessPolicy](#)」の「」を参照してください。

create-load-balancer-listeners

次のコード例は、create-load-balancer-listeners を使用する方法を示しています。

AWS CLI

ロードバランサーのHTTPリスナーを作成するには

この例では、HTTPプロトコルを使用してポート 80 でロードバランサーのリスナーを作成します。

コマンド:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
```

ロードバランサーのHTTPSリスナーを作成するには

この例では、HTTPSプロトコルを使用してポート 443 でロードバランサーのリスナーを作成します。

コマンド:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTP,InstancePort=80"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoadBalancerListeners](#)」の「」を参照してください。

create-load-balancer-policy

次のコード例は、create-load-balancer-policy を使用する方法を示しています。

AWS CLI

ロードバランサーでプロキシプロトコルを有効にするポリシーを作成するには

この例では、指定されたロードバランサーでプロキシプロトコルを有効にするポリシーを作成します。

コマンド:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-  
name my-ProxyProtocol-policy --policy-type-name ProxyProtocolPolicyType --policy-  
attributes AttributeName=ProxyProtocol,AttributeValue=true
```

推奨されるセキュリティポリシーを使用してSSLネゴシエーションポリシーを作成するには

この例では、推奨されるセキュリティポリシーを使用して、指定されたHTTPSロードバランサーのSSLネゴシエーションポリシーを作成します。

コマンド:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-  
balancer --policy-name my-SSLNegotiation-policy --policy-type-
```

```
name SSLNegotiationPolicyType --policy-attributes AttributeName=Reference-Security-Policy,AttributeValue=ELBSecurityPolicy-2015-03
```

カスタムセキュリティポリシーを使用してSSLネゴシエーションポリシーを作成するには

この例では、プロトコルと暗号を有効にして、カスタムセキュリティポリシーを使用してHTTPSロードバランサーのSSLネゴシエーションポリシーを作成します。

コマンド:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-SSLNegotiation-policy --policy-type-name SSLNegotiationPolicyType --policy-attributes AttributeName=Protocol-SSLv3,AttributeValue=true AttributeName=Protocol-TLSv1.1,AttributeValue=true AttributeName=DHE-RSA-AES256-SHA256,AttributeValue=true AttributeName=Server-Defined-Cipher-Order,AttributeValue=true
```

パブリックキーポリシーを作成するには

この例では、パブリックキーポリシーを作成します。

コマンド:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-PublicKey-policy --policy-type-name PublicKeyPolicyType --policy-attributes AttributeName=PublicKey,AttributeValue=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ+dS74kj//c6x7R0tusUaeQCTgIUkayttRDWchuqo1pHC1u+n5xxXnBBE2ejbb2WRsKIQ5rXEeixsjFpFsojpSQKkzhVGI6mJVZBJDVKSHmswnwLBdofLhzvllpovBPTHe+o4haAWvDBALJU0pkSI1FecPHcs2hwx14zHoXy1e2k36A64nXW43wtfx5qcVSIxtCE0jnYRg7RPvybaGfQ+v6Iaxb/+7J5kEvZhTFQId+bSiJImF1FSUT1W1xwzBZPUbcUkkXDj45vC2s3Z8E+Lk7a3uZhvsQHLZnrFuWjBWGWvZ/MhZYgEXAMPLE
```

バックエンドサーバー認証ポリシーを作成するには

この例では、パブリックキーポリシーを使用してバックエンドインスタンスで認証を有効にするバックエンドサーバー認証ポリシーを作成します。

コマンド:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-authentication-policy --policy-
```



```
type-name BackendServerAuthenticationPolicyType --policy-attributes AttributeName=PublicKeyPolicyName,AttributeValue=my-PublicKey-policy
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoadBalancerPolicy](#)」の「」を参照してください。

create-load-balancer

次のコード例は、create-load-balancer を使用する方法を示しています。

AWS CLI

HTTPロードバランサーを作成するには

この例では、にHTTPリスナーを持つロードバランサーを作成しますVPC。

コマンド:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

出力:

```
{  
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"  
}
```

この例では、EC2-Classic でHTTPリスナーを持つロードバランサーを作成します。

コマンド:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" --availability-zones us-west-2a us-west-2b
```

出力:

```
{  
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"  
}
```

HTTPSロードバランサーを作成するには

この例では、 にHTTPSリスナーを持つロードバランサーを作成しますVPC。

コマンド:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTPS,InstancePort=443,certificate/my-server-cert" --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

出力:

```
{  
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"  
}
```

この例では、 EC2-Classic でHTTPSリスナーを持つロードバランサーを作成します。

コマンド:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTPS,InstancePort=443,certificate/my-server-cert" --availability-zones us-west-2a us-west-2b
```

出力:

```
{  
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"  
}
```

内部ロードバランサーを作成するには

この例では、 にHTTPリスナーを持つ内部ロードバランサーを作成しますVPC。

コマンド:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" --scheme internal --subnets subnet-a85db0df --security-groups sg-a61988c3
```

出力:

```
{
  "DNSName": "internal-my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoadBalancer](#)」の「」を参照してください。

delete-load-balancer-listeners

次の例は、delete-load-balancer-listeners を使用する方法を説明しています。

AWS CLI

ロードバランサーからリスナーを削除するには

この例では、指定されたロードバランサーから指定されたポートのリスナーを削除します。

コマンド:

```
aws elb delete-load-balancer-listeners --load-balancer-name my-load-balancer --load-balancer-ports 80
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLoadBalancerListeners](#)」の「」を参照してください。

delete-load-balancer-policy

次のコード例は、delete-load-balancer-policy を使用する方法を示しています。

AWS CLI

ロードバランサーからポリシーを削除するには

この例では、指定されたロードバランサーから指定されたポリシーを削除します。リスナーでポリシーを有効にしないでください。

コマンド:

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLoadBalancerPolicy](#)」の「」を参照してください。

delete-load-balancer

次の例は、delete-load-balancer を使用する方法を説明しています。

AWS CLI

ロードバランサーを削除するには

この例では、指定されたロードバランサーを削除します。

コマンド:

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLoadBalancer](#)」の「」を参照してください。

deregister-instances-from-load-balancer

次の例は、deregister-instances-from-load-balancer を使用する方法を説明しています。

AWS CLI

ロードバランサーからインスタンスの登録を解除するには

この例では、指定されたロードバランサーから指定されたインスタンスを登録解除します。

コマンド:

```
aws elb deregister-instances-from-load-balancer --load-balancer-name my-load-balancer --instances i-d6f6fae3
```

出力:

```
{
  "Instances": [
    {
      "InstanceId": "i-207d9717"
    }
  ]
}
```

```
    },
    {
      "InstanceId": "i-afefb49b"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterInstancesFromLoadBalancer](#)」の「」を参照してください。

describe-account-limits

次のコード例は、describe-account-limits を使用する方法を示しています。

AWS CLI

Classic Load Balancer の制限を記述するには

次のdescribe-account-limits例では、AWS アカウントの Classic Load Balancer の制限に関する詳細を表示します。

```
aws elb describe-account-limits
```

出力:

```
{
  "Limits": [
    {
      "Name": "classic-load-balancers",
      "Max": "20"
    },
    {
      "Name": "classic-listeners",
      "Max": "100"
    },
    {
      "Name": "classic-registered-instances",
      "Max": "1000"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAccountLimits](#)」の「」を参照してください。

describe-instance-health

次のコード例は、describe-instance-health を使用する方法を示しています。

AWS CLI

ロードバランサーのインスタンスのヘルスを記述するには

この例では、指定されたロードバランサーのインスタンスの状態について説明します。

コマンド:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer
```

出力:

```
{
  "InstanceStates": [
    {
      "InstanceId": "i-207d9717",
      "ReasonCode": "N/A",
      "State": "InService",
      "Description": "N/A"
    },
    {
      "InstanceId": "i-afefb49b",
      "ReasonCode": "N/A",
      "State": "InService",
      "Description": "N/A"
    }
  ]
}
```

ロードバランサーのインスタンスのヘルスを記述するには

この例では、指定されたロードバランサーの指定されたインスタンスのヘルスについて説明します。

コマンド:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer --  
instances i-7299c809
```

登録しているインスタンスのレスポンスの例を次に示します。

出力:

```
{  
  "InstanceStates": [  
    {  
      "InstanceId": "i-7299c809",  
      "ReasonCode": "ELB",  
      "State": "OutOfService",  
      "Description": "Instance registration is still in progress."  
    }  
  ]  
}
```

以下は、異常なインスタンスのレスポンスの例です。

出力:

```
{  
  "InstanceStates": [  
    {  
      "InstanceId": "i-7299c809",  
      "ReasonCode": "Instance",  
      "State": "OutOfService",  
      "Description": "Instance has failed at least the UnhealthyThreshold number  
of health checks consecutively."  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceHealth](#)」の「」を参照してください。

describe-load-balancer-attributes

次の例は、describe-load-balancer-attributes を使用する方法を説明しています。

AWS CLI

ロードバランサーの属性を記述するには

この例では、指定されたロードバランサーの属性について説明します。

コマンド:

```
aws elb describe-load-balancer-attributes --load-balancer-name my-load-balancer
```

出力:

```
{
  "LoadBalancerAttributes": {
    "ConnectionDraining": {
      "Enabled": false,
      "Timeout": 300
    },
    "CrossZoneLoadBalancing": {
      "Enabled": true
    },
    "ConnectionSettings": {
      "IdleTimeout": 30
    },
    "AccessLog": {
      "Enabled": false
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoadBalancerAttributes](#)」の「」を参照してください。

describe-load-balancer-policies

次のコード例は、describe-load-balancer-policies を使用する方法を示しています。

AWS CLI

ロードバランサーに関連付けられているすべてのポリシーを記述するには

この例では、指定されたロードバランサーに関連付けられているすべてのポリシーについて説明します。

コマンド:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer
```

出力:

```
{
  "PolicyDescriptions": [
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "ProxyProtocol",
          "AttributeValue": "true"
        }
      ],
      "PolicyName": "my-ProxyProtocol-policy",
      "PolicyTypeName": "ProxyProtocolPolicyType"
    },
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "CookieName",
          "AttributeValue": "my-app-cookie"
        }
      ],
      "PolicyName": "my-app-cookie-policy",
      "PolicyTypeName": "AppCookieStickinessPolicyType"
    },
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "CookieExpirationPeriod",
          "AttributeValue": "60"
        }
      ],
      "PolicyName": "my-duration-cookie-policy",
      "PolicyTypeName": "LBCookieStickinessPolicyType"
    },
    .
    .
    .
  ]
}
```

ロードバランサーに関連付けられた特定のポリシーを記述するには

この例では、指定されたロードバランサーに関連付けられた指定されたポリシーについて説明します。

コマンド:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer --  
policy-name my-authentication-policy
```

出力:

```
{  
  "PolicyDescriptions": [  
    {  
      "PolicyAttributeDescriptions": [  
        {  
          "AttributeName": "PublicKeyPolicyName",  
          "AttributeValue": "my-PublicKey-policy"  
        }  
      ],  
      "PolicyName": "my-authentication-policy",  
      "PolicyTypeName": "BackendServerAuthenticationPolicyType"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoadBalancerPolicies](#)」の「」を参照してください。

describe-load-balancer-policy-types

次の例は、describe-load-balancer-policy-types を使用する方法を説明しています。

AWS CLI

Elastic Load Balancing で定義されるロードバランサーポリシータイプを記述するには

この例では、ロードバランサーのポリシー設定を作成するために使用できるロードバランサーポリシータイプについて説明します。

コマンド:

```
aws elb describe-load-balancer-policy-types
```

出力:

```
{
  "PolicyTypeDescriptions": [
    {
      "PolicyAttributeTypeDescriptions": [
        {
          "Cardinality": "ONE",
          "AttributeName": "ProxyProtocol",
          "AttributeType": "Boolean"
        }
      ],
      "PolicyTypeName": "ProxyProtocolPolicyType",
      "Description": "Policy that controls whether to include the IP address and port of the originating request for TCP messages. This policy operates on TCP/SSL listeners only"
    },
    {
      "PolicyAttributeTypeDescriptions": [
        {
          "Cardinality": "ONE",
          "AttributeName": "PublicKey",
          "AttributeType": "String"
        }
      ],
      "PolicyTypeName": "PublicKeyPolicyType",
      "Description": "Policy containing a list of public keys to accept when authenticating the back-end server(s). This policy cannot be applied directly to back-end servers or listeners but must be part of a BackendServerAuthenticationPolicyType."
    },
    {
      "PolicyAttributeTypeDescriptions": [
        {
          "Cardinality": "ONE",
          "AttributeName": "CookieName",
          "AttributeType": "String"
        }
      ],
    }
  ],
}
```

```
    "PolicyTypeName": "AppCookieStickinessPolicyType",
    "Description": "Stickiness policy with session lifetimes controlled by the
lifetime of the application-generated cookie. This policy can be associated only
with HTTP/HTTPS listeners."
  },
  {
    "PolicyAttributeTypeDescriptions": [
      {
        "Cardinality": "ZERO_OR_ONE",
        "AttributeName": "CookieExpirationPeriod",
        "AttributeType": "Long"
      }
    ],
    "PolicyTypeName": "LBCookieStickinessPolicyType",
    "Description": "Stickiness policy with session lifetimes controlled by
the browser (user-agent) or a specified expiration period. This policy can be
associated only with HTTP/HTTPS listeners."
  },
  {
    "PolicyAttributeTypeDescriptions": [
      .
      .
      .
    ],
    "PolicyTypeName": "SSLNegotiationPolicyType",
    "Description": "Listener policy that defines the ciphers and protocols
that will be accepted by the load balancer. This policy can be associated only with
HTTPS/SSL listeners."
  },
  {
    "PolicyAttributeTypeDescriptions": [
      {
        "Cardinality": "ONE_OR_MORE",
        "AttributeName": "PublicKeyPolicyName",
        "AttributeType": "PolicyName"
      }
    ],
    "PolicyTypeName": "BackendServerAuthenticationPolicyType",
    "Description": "Policy that controls authentication to back-end server(s)
and contains one or more policies, such as an instance of a PublicKeyPolicyType.
This policy can be associated only with back-end servers that are using HTTPS/SSL."
  }
]
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoadBalancerPolicyTypes](#)」の「」を参照してください。

describe-load-balancers

次のコード例は、describe-load-balancers を使用する方法を示しています。

AWS CLI

ロードバランサーを記述するには

この例は、すべてのロードバランサーを記述します。

コマンド:

```
aws elb describe-load-balancers
```

ロードバランサーの 1 つを記述するには

この例は、指定されたロードバランサーを記述します。

コマンド:

```
aws elb describe-load-balancers --load-balancer-name my-load-balancer
```

次のレスポンス例は、のHTTPSロードバランサーの場合ですVPC。

出力:

```
{
  "LoadBalancerDescriptions": [
    {
      "Subnets": [
        "subnet-15aaab61"
      ],
      "CanonicalHostedZoneNameID": "Z3DZXE0EXAMPLE",
      "CanonicalHostedZoneName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com",
      "ListenerDescriptions": [
        {
          "Listener": {
```

```
        "InstancePort": 80,
        "LoadBalancerPort": 80,
        "Protocol": "HTTP",
        "InstanceProtocol": "HTTP"
    },
    "PolicyNames": []
},
{
    "Listener": {
        "InstancePort": 443,
        "SSLCertificateId": "arn:aws:iam::123456789012:server-certificate/
my-server-cert",
        "LoadBalancerPort": 443,
        "Protocol": "HTTPS",
        "InstanceProtocol": "HTTPS"
    },
    "PolicyNames": [
        "ELBSecurityPolicy-2015-03"
    ]
}
],
"HealthCheck": {
    "HealthyThreshold": 2,
    "Interval": 30,
    "Target": "HTTP:80/png",
    "Timeout": 3,
    "UnhealthyThreshold": 2
},
"VPCId": "vpc-a01106c2",
"BackendServerDescriptions": [
    {
        "InstancePort": 80,
        "PolicyNames": [
            "my-ProxyProtocol-policy"
        ]
    }
],
"Instances": [
    {
        "InstanceId": "i-207d9717"
    },
    {
        "InstanceId": "i-afefb49b"
    }
]
```

```
    ],
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com",
    "SecurityGroups": [
      "sg-a61988c3"
    ],
    "Policies": {
      "LBCookieStickinessPolicies": [
        {
          "PolicyName": "my-duration-cookie-policy",
          "CookieExpirationPeriod": 60
        }
      ],
      "AppCookieStickinessPolicies": [],
      "OtherPolicies": [
        "my-PublicKey-policy",
        "my-authentication-policy",
        "my-SSLNegotiation-policy",
        "my-ProxyProtocol-policy",
        "ELBSecurityPolicy-2015-03"
      ]
    },
    "LoadBalancerName": "my-load-balancer",
    "CreatedTime": "2015-03-19T03:24:02.650Z",
    "AvailabilityZones": [
      "us-west-2a"
    ],
    "Scheme": "internet-facing",
    "SourceSecurityGroup": {
      "OwnerAlias": "123456789012",
      "GroupName": "my-elb-sg"
    }
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLoadBalancers](#)」の「」を参照してください。

describe-tags

次のコード例は、describe-tags を使用する方法を示しています。

AWS CLI

ロードバランサーに割り当てられたタグを記述するには

この例では、指定されたロードバランサーに割り当てられたタグについて説明します。

コマンド:

```
aws elb describe-tags --load-balancer-name my-load-balancer
```

出力:

```
{
  "TagDescriptions": [
    {
      "Tags": [
        {
          "Value": "lima",
          "Key": "project"
        },
        {
          "Value": "digital-media",
          "Key": "department"
        }
      ],
      "LoadBalancerName": "my-load-balancer"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

detach-load-balancer-from-subnets

次の例は、detach-load-balancer-from-subnets を使用する方法を説明しています。

AWS CLI

サブネットからロードバランサーをデタッチするには

この例では、指定されたロードバランサーを指定されたサブネットからデタッチします。

コマンド:

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-load-balancer --  
subnets subnet-0ecac448
```

出力:

```
{  
  "Subnets": [  
    "subnet-15aaab61"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachLoadBalancerFromSubnets](#)」の「」を参照してください。

disable-availability-zones-for-load-balancer

次のコード例は、`disable-availability-zones-for-load-balancer` を使用方法を示しています。

AWS CLI

ロードバランサーのアベイラビリティーゾーンを無効にするには

この例では、指定されたアベイラビリティーゾーンを、指定されたロードバランサーのアベイラビリティーゾーンのセットから削除します。

コマンド:

```
aws elb disable-availability-zones-for-load-balancer --load-balancer-name my-load-  
balancer --availability-zones us-west-2a
```

出力:

```
{  
  "AvailabilityZones": [  
    "us-west-2b"  
  ]  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DisableAvailabilityZonesForLoadBalancer](#)」の「」を参照してください。

enable-availability-zones-for-load-balancer

次のコード例は、`enable-availability-zones-for-load-balancer` を使用する方法を示しています。

AWS CLI

ロードバランサーのアベイラビリティゾーンを有効にするには

この例では、指定されたアベイラビリティゾーンを指定されたロードバランサーに追加します。

コマンド:

```
aws elb enable-availability-zones-for-load-balancer --load-balancer-name my-Load-balancer --availability-zones us-west-2b
```

出力:

```
{
  "AvailabilityZones": [
    "us-west-2a",
    "us-west-2b"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [EnableAvailabilityZonesForLoadBalancer](#)」の「」を参照してください。

modify-load-balancer-attributes

次の例は、`modify-load-balancer-attributes` を使用する方法を説明しています。

AWS CLI

ロードバランサーの属性を変更するには

この例では、指定されたロードバランサーの `CrossZoneLoadBalancing` 属性を変更します。

コマンド:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

出力:

```
{
  "LoadBalancerAttributes": {
    "CrossZoneLoadBalancing": {
      "Enabled": true
    }
  },
  "LoadBalancerName": "my-load-balancer"
}
```

この例では、指定されたロードバランサーの `ConnectionDraining` 属性を変更します。

コマンド:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"ConnectionDraining\":{\"Enabled\":true,\"Timeout\":300}}"
```

出力:

```
{
  "LoadBalancerAttributes": {
    "ConnectionDraining": {
      "Enabled": true,
      "Timeout": 300
    }
  },
  "LoadBalancerName": "my-load-balancer"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyLoadBalancerAttributes](#)」の「」を参照してください。

register-instances-with-load-balancer

次のコード例は、`register-instances-with-load-balancer` を使用する方法を示しています。

AWS CLI

ロードバランサーにインスタンスを登録するには

この例では、指定されたインスタンスを指定されたロードバランサーに登録します。

コマンド:

```
aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer
--instances i-d6f6fae3
```

出力:

```
{
  "Instances": [
    {
      "InstanceId": "i-d6f6fae3"
    },
    {
      "InstanceId": "i-207d9717"
    },
    {
      "InstanceId": "i-afefb49b"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RegisterInstancesWithLoadBalancer](#)」の「」を参照してください。

remove-tags

次の例は、`remove-tags` を使用する方法を説明しています。

AWS CLI

ロードバランサーからタグを削除するには

この例では、指定されたロードバランサーからタグを削除します。

コマンド:

```
aws elb remove-tags --load-balancer-name my-load-balancer --tags project
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveTags](#)」の「」を参照してください。

set-load-balancer-listener-ssl-certificate

次の例は、set-load-balancer-listener-ssl-certificate を使用する方法を説明しています。

AWS CLI

HTTPSロードバランサーのSSL証明書を更新するには

この例では、指定されたHTTPSロードバランサーの既存のSSL証明書を置き換えます。

コマンド:

```
aws elb set-load-balancer-listener-ssl-certificate --load-balancer-name my-load-balancer --load-balancer-port 443 --ssl-certificate-id arn:aws:iam::123456789012:server-certificate/new-server-cert
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetLoadBalancerListenerSslCertificate](#)」の「」を参照してください。

set-load-balancer-policies-for-backend-server

次の例は、set-load-balancer-policies-for-backend-server を使用する方法を説明しています。

AWS CLI

バックエンドインスタンスのポートに関連付けられたポリシーを置き換えるには

この例では、現在指定されたポートに関連付けられているポリシーを置き換えます。

コマンド:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names my-ProxyProtocol-policy
```

バックエンドインスタンスのポートに現在関連付けられているすべてのポリシーを削除するにはこの例では、指定されたポートに関連付けられているすべてのポリシーを削除します。

コマンド:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names []
```

ポリシーが削除されていることを確認するには、`describe-load-balancer-policies` コマンドを使用します。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetLoadBalancerPoliciesForBackendServer](#)」の「」を参照してください。

set-load-balancer-policies-of-listener

次の例は、`set-load-balancer-policies-of-listener` を使用する方法を説明しています。

AWS CLI

リスナーに関連付けられたポリシーを置き換えるには

この例では、指定されたリスナーに現在関連付けられているポリシーを置き換えます。

コマンド:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer --load-balancer-port 443 --policy-names my-SSLNegotiation-policy
```

リスナーに関連付けられているすべてのポリシーを削除するには

この例では、指定されたリスナーに現在関連付けられているすべてのポリシーを削除します。

コマンド:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer --load-balancer-port 443 --policy-names []
```

ポリシーがロードバランサーから削除されていることを確認するには、`describe-load-balancer-policies` コマンドを使用します。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetLoadBalancerPoliciesOfListener](#)」の「」を参照してください。

Elastic Load Balancing - を使用したバージョン 2 の例 AWS CLI

次のコード例は、Elastic Load Balancing - バージョン 2 AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-listener-certificates

次の例は、`add-listener-certificates` を使用する方法を説明しています。

AWS CLI

セキュアリスナーに証明書を追加するには

この例では、指定された証明書を指定されたセキュアリスナーに追加します。

コマンド:

```
aws elbv2 add-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705
```

出力:

```
{
  "Certificates": [
    {
      "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
      "IsDefault": false
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddListenerCertificates](#)」の「」を参照してください。

add-tags

次のコード例は、add-tags を使用する方法を示しています。

AWS CLI

ロードバランサーにタグを追加するには

次のadd-tags例では、指定されたロードバランサーに project と department タグを追加します。

```
aws elbv2 add-tags \
  --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
  --tags "Key=project, Value=lima" "Key=department, Value=digital-media"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTags](#)」の「」を参照してください。

create-listener

次の例は、create-listener を使用する方法を説明しています。

AWS CLI

例 1: HTTPリスナーを作成するには

次のcreate-listener例では、指定されたターゲットグループにリクエストを転送する、指定された Application Load Balancer のHTTPリスナーを作成します。

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --protocol HTTP \  
  --port 80 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

詳細については、[Application Load Balancer ユーザーガイド](#)の「チュートリアル: [を使用して Application Load Balancer を作成する AWS CLI](#)」を参照してください。

例 2: HTTPSリスナーを作成するには

次のcreate-listener例では、指定されたターゲットグループにリクエストを転送する、指定された Application Load Balancer のHTTPSリスナーを作成します。HTTPS リスナーのSSL証明書を指定する必要があります。AWS Certificate Manager () を使用して証明書を作成および管理できますACM。または、SSL/TLS ツールを使用して証明書を作成し、認証局 (CA) によって署名された証明書を取得し、その証明書を AWS Identity and Access Management () にアップロードすることもできますIAM。

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --protocol HTTPS \  
  --port 443 \  
  --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \  
  --ssl-policy ELBSecurityPolicy-2016-08 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

詳細については、「[Application Load Balancer ユーザーガイド](#)」の[HTTPS 「リスナーの追加」](#)を参照してください。

例 3: TCPリスナーを作成するには

次のcreate-listener例では、指定されたターゲットグループにリクエストを転送する、指定された Network Load Balancer のTCPリスナーを作成します。

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \  
  --protocol TCP \  
  --port 80 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

詳細については、[「Network Load Balancer ユーザーガイド」の「チュートリアル: を使用して AWS CLI Network Load Balancer を作成する」](#)を参照してください。

例 4: TLSリスナーを作成するには

次のcreate-listener例では、指定されたターゲットグループにリクエストを転送する、指定された Network Load Balancer のTLSリスナーを作成します。TLS リスナーのSSL証明書を指定する必要があります。

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --protocol TLS \  
  --port 443 \  
  --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \  
  --ssl-policy ELBSecurityPolicy-2016-08 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

詳細については、[TLS 「Network Load Balancer ユーザーガイド」の「Network Load Balancer のリスナー」](#)を参照してください。

例 5: UDPリスナーを作成するには

次のcreate-listener例では、指定されたターゲットグループにリクエストを転送する、指定された Network Load Balancer のUDPリスナーを作成します。

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \  
  --protocol UDP \  
  --port 53 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

```
--default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

詳細については、[「Network Load Balancer のユーザーガイド」の「チュートリアル: を使用して AWS CLI Network Load Balancer を作成する」](#)を参照してください。

例 6: 指定されたゲートウェイのリスナーを作成して転送するには

次の create-listener の例は、指定されたターゲットグループにリクエストを転送する、指定された Gateway Load Balancer のリスナーを作成します。

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/e0f9b3d5c7f7d3d6 \  
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-east-1:850631746142:targetgroup/my-glb-targets/007ca469fae3bb1615
```

出力:

```
{  
  "Listeners": [  
    {  
      "ListenerArn": "arn:aws:elasticloadbalancing:us-east-1:850631746142:listener/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6/afc127db15f925de",  
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:850631746142:loadbalancer/gwy/my-agw-lb-example2/e0f9b3d5c7f7d3d6",  
      "DefaultActions": [  
        {  
          "Type": "forward",  
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615",  
          "ForwardConfig": {  
            "TargetGroups": [  
              {  
                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:850631746142:targetgroup/test-tg-agw-2/007ca469fae3bb1615"  
              }  
            ]  
          }  
        }  
      ]  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

詳細については、[「Gateway Load Balancer ユーザーガイド」の「を使用した Gateway Load Balancer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateListener](#)」の「」を参照してください。

create-load-balancer

次の例は、create-load-balancer を使用する方法を説明しています。

AWS CLI

例 1: インターネット向けロードバランサーを作成するには

次の create-load-balancer の例は、インターネット向けの Application Load Balancer を作成し、指定されたサブネットのアベイラビリティゾーンを有効化します。

```
aws elbv2 create-load-balancer \  
  --name my-load-balancer \  
  --subnets subnet-b7d581c0 subnet-8360a9e7
```

出力:

```
{  
  "LoadBalancers": [  
    {  
      "Type": "application",  
      "Scheme": "internet-facing",  
      "IpAddressType": "ipv4",  
      "VpcId": "vpc-3ac0fb5f",  
      "AvailabilityZones": [  
        {  
          "ZoneName": "us-west-2a",  
          "SubnetId": "subnet-8360a9e7"  
        },  
        {  
          "ZoneName": "us-west-2b",  
          "SubnetId": "subnet-b7d581c0"  
        }  
      ]  
    }  
  ]  
}
```

```

    }
  ],
  "CreatedTime": "2017-08-25T21:26:12.920Z",
  "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
  "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
  "SecurityGroups": [
    "sg-5943793c"
  ],
  "LoadBalancerName": "my-load-balancer",
  "State": {
    "Code": "provisioning"
  },
  "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"
}
]
}

```

詳細については、[Application Load Balancer ユーザーガイド](#)の「チュートリアル: [を使用して Application Load Balancer を作成する AWS CLI](#)」を参照してください。

例 2: 内部向けロードバランサーを作成するには

次の `create-load-balancer` の例は、内部向けの Application Load Balancer を作成し、指定されたサブネットのアベイラビリティゾーンを有効化します。

```

aws elbv2 create-load-balancer \
  --name my-internal-load-balancer \
  --scheme internal \
  --subnets subnet-b7d581c0 subnet-8360a9e7

```

出力:

```

{
  "LoadBalancers": [
    {
      "Type": "application",
      "Scheme": "internal",
      "IpAddressType": "ipv4",
      "VpcId": "vpc-3ac0fb5f",
      "AvailabilityZones": [
        {
          "ZoneName": "us-west-2a",

```

```

        "SubnetId": "subnet-8360a9e7"
      },
      {
        "ZoneName": "us-west-2b",
        "SubnetId": "subnet-b7d581c0"
      }
    ],
    "CreatedTime": "2016-03-25T21:29:48.850Z",
    "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
    "DNSName": "internal-my-internal-load-balancer-1529930873.us-
west-2.elb.amazonaws.com",
    "SecurityGroups": [
      "sg-5943793c"
    ],
    "LoadBalancerName": "my-internal-load-balancer",
    "State": {
      "Code": "provisioning"
    },
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-internal-load-balancer/5b49b8d4303115c2"
  }
]
}

```

詳細については、[Application Load Balancer ユーザーガイド](#)の「チュートリアル: を使用して Application Load Balancer を作成する AWS CLI」を参照してください。

例 3: Network Load Balancer を作成するには

次の `create-load-balancer` の例は、インターネット向けの Network Load Balancer を作成し、指定されたサブネットの Availability Zones を有効化します。サブネットマッピングを使用して、指定された Elastic IP アドレスを、Availability Zones のロードバランサーノードが使用するネットワークインターフェイスに関連付けます。

```

aws elbv2 create-load-balancer \
  --name my-network-load-balancer \
  --type network \
  --subnet-mappings SubnetId=subnet-b7d581c0,AllocationId=eipalloc-64d5890a

```

出力:

```
{
```

```
"LoadBalancers": [
  {
    "Type": "network",
    "Scheme": "internet-facing",
    "IpAddressType": "ipv4",
    "VpcId": "vpc-3ac0fb5f",
    "AvailabilityZones": [
      {
        "LoadBalancerAddresses": [
          {
            "IpAddress": "35.161.207.171",
            "AllocationId": "eipalloc-64d5890a"
          }
        ],
        "ZoneName": "us-west-2b",
        "SubnetId": "subnet-5264e837"
      }
    ],
    "CreatedTime": "2017-10-15T22:41:25.657Z",
    "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
    "DNSName": "my-network-load-balancer-5d1b75f4f1cee11e.elb.us-
west-2.amazonaws.com",
    "LoadBalancerName": "my-network-load-balancer",
    "State": {
      "Code": "provisioning"
    },
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e"
  }
]
```

詳細については、[「Network Load Balancer ユーザーガイド」の「を使用して Network Load Balancer を作成する AWS CLI」](#)を参照してください。

例 4: Gateway Load Balancer を作成するには

次の create-load-balancer の例は、Gateway Load Balancer を作成し、指定されたサブネットの Availability Zones を有効化します。

```
aws elbv2 create-load-balancer \
  --name my-gateway-load-balancer \
  --type gateway \
```

```
--subnets subnet-dc83f691 subnet-a62583f9
```

出力:

```
{
  "LoadBalancers": [
    {
      "Type": "gateway",
      "VpcId": "vpc-838475fe",
      "AvailabilityZones": [
        {
          "ZoneName": "us-east-1b",
          "SubnetId": "subnet-a62583f9"
        },
        {
          "ZoneName": "us-east-1a",
          "SubnetId": "subnet-dc83f691"
        }
      ],
      "CreatedTime": "2021-07-14T19:33:43.324000+00:00",
      "LoadBalancerName": "my-gateway-load-balancer",
      "State": {
        "Code": "provisioning"
      },
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:850631746142:loadbalancer/gwy/my-gateway-load-balancer/dfbb5a7d32cdee79"
    }
  ]
}
```

詳細については、[「Gateway Load Balancer ユーザーガイド」の「を使用した Gateway Load Balancer の開始方法 AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoadBalancer](#)」の「」を参照してください。

create-rule

次の例は、create-rule を使用する方法を説明しています。

AWS CLI

例 1: パス条件とフォワードアクションを使用してルールを作成するには

次のcreate-rule例では、に指定されたパターンURLが含まれている場合に、指定されたターゲットグループにリクエストを転送するルールを作成します。

```
aws elbv2 create-rule \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \  
  --priority 5 \  
  --conditions file://conditions-pattern.json \  
  --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

conditions-pattern.json の内容:

```
[  
  {  
    "Field": "path-pattern",  
    "PathPatternConfig": {  
      "Values": ["/images/*"]  
    }  
  }  
]
```

例 2: ホスト条件と固定レスポンスを使用してルールを作成するには

次のcreate-rule例では、ホストヘッダーのホスト名が指定されたホスト名と一致する場合に、固定レスポンスを提供するルールを作成します。

```
aws elbv2 create-rule \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \  
  --priority 10 \  
  --conditions file://conditions-host.json \  
  --actions file://actions-fixed-response.json
```

conditions-host.json の内容

```
[  
  {  
    "Field": "host-header",  
    "HostHeaderConfig": {  
      "Values": [ "*.example.com" ]  
    }  
  }  
]
```

```
}  
]
```

actions-fixed-response.json の内容

```
[  
  {  
    "Type": "fixed-response",  
    "FixedResponseConfig": {  
      "MessageBody": "Hello world",  
      "StatusCode": "200",  
      "ContentType": "text/plain"  
    }  
  }  
]
```

例 3: ソース IP アドレス条件、認証アクション、転送アクションを使用してルールを作成するには

次の `create-rule` 例では、送信元 IP アドレスが指定された IP アドレスと一致する場合にユーザーを認証するルールを作成し、認証が成功した場合にリクエストを指定されたターゲットグループに転送します。

```
aws elbv2 create-rule \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \  
  --priority 20 \  
  --conditions file://conditions-source-ip.json \  
  --actions file://actions-authenticate.json
```

conditions-source-ip.json の内容

```
[  
  {  
    "Field": "source-ip",  
    "SourceIpConfig": {  
      "Values": ["192.0.2.0/24", "198.51.100.10/32"]  
    }  
  }  
]
```

actions-authenticate.json の内容

```
[
  {
    "Type": "authenticate-oidc",
    "AuthenticateOidcConfig": {
      "Issuer": "https://idp-issuer.com",
      "AuthorizationEndpoint": "https://authorization-endpoint.com",
      "TokenEndpoint": "https://token-endpoint.com",
      "UserInfoEndpoint": "https://user-info-endpoint.com",
      "ClientId": "abcdefghijklmnopqrstuvwxy123456789",
      "ClientSecret": "123456789012345678901234567890",
      "SessionCookieName": "my-cookie",
      "SessionTimeout": 3600,
      "Scope": "email",
      "AuthenticationRequestExtraParams": {
        "display": "page",
        "prompt": "login"
      },
      "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
  },
  {
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:880185128111:targetgroup/cli-test/642a97ecb0e0f26b",
    "Order": 2
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRule](#)」の「」を参照してください。

create-target-group

次のコード例は、create-target-group を使用する方法を示しています。

AWS CLI

例 1: Application Load Balancer のターゲットグループを作成するには

次の create-target-group の例は、Application Load Balancer のターゲットグループを作成し、インスタンス ID (ターゲットタイプは instance) でターゲットを登録します。このター

ゲットグループは、HTTPターゲットグループのHTTPプロトコル、ポート 80、およびデフォルトのヘルスチェック設定を使用します。

```
aws elbv2 create-target-group \  
  --name my-targets \  
  --protocol HTTP \  
  --port 80 \  
  --target-type instance \  
  --vpc-id vpc-3ac0fb5f
```

出力:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",  
      "TargetGroupName": "my-targets",  
      "Protocol": "HTTP",  
      "Port": 80,  
      "VpcId": "vpc-3ac0fb5f",  
      "HealthCheckProtocol": "HTTP",  
      "HealthCheckPort": "traffic-port",  
      "HealthCheckEnabled": true,  
      "HealthCheckIntervalSeconds": 30,  
      "HealthCheckTimeoutSeconds": 5,  
      "HealthyThresholdCount": 5,  
      "UnhealthyThresholdCount": 2,  
      "HealthCheckPath": "/",  
      "Matcher": {  
        "HttpCode": "200"  
      },  
      "TargetType": "instance",  
      "ProtocolVersion": "HTTP1",  
      "IpAddressType": "ipv4"  
    }  
  ]  
}
```

詳細については、「Application Load Balancers のユーザーガイド」の「[ターゲットグループの作成](#)」を参照してください。

例 2: Application Load Balancer から Lambda 関数にトラフィックをルーティングするターゲットグループを作成するには

次の `create-target-group` の例は、Application Load Balancer のターゲットグループを作成します。ターゲットは Lambda 関数 (ターゲットタイプは `lambda`) です。このターゲットグループのヘルスチェックは、デフォルトでは無効化されています。

```
aws elbv2 create-target-group \  
  --name my-lambda-target \  
  --target-type lambda
```

出力:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-lambda-target/a3003e085dbb8ddc",  
      "TargetGroupName": "my-lambda-target",  
      "HealthCheckEnabled": false,  
      "HealthCheckIntervalSeconds": 35,  
      "HealthCheckTimeoutSeconds": 30,  
      "HealthyThresholdCount": 5,  
      "UnhealthyThresholdCount": 2,  
      "HealthCheckPath": "/",  
      "Matcher": {  
        "HttpCode": "200"  
      },  
      "TargetType": "lambda",  
      "IpAddressType": "ipv4"  
    }  
  ]  
}
```

詳細については、Application Load Balancer ユーザーガイドの [ターゲットとしての Lambda 関数](#) を参照してください。

例 3: Network Load Balancer のターゲットグループを作成するには

次の `create-target-group` の例は、Network Load Balancer のターゲットグループを作成し、IP アドレス (ターゲットタイプは `ip`) でターゲットを登録します。このターゲットグループ

は、TCPターゲットグループのTCPプロトコル、ポート 80、およびデフォルトのヘルスチェック設定を使用します。

```
aws elbv2 create-target-group \  
  --name my-ip-targets \  
  --protocol TCP \  
  --port 80 \  
  --target-type ip \  
  --vpc-id vpc-3ac0fb5f
```

出力:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-ip-targets/b6bba954d1361c78",  
      "TargetGroupName": "my-ip-targets",  
      "Protocol": "TCP",  
      "Port": 80,  
      "VpcId": "vpc-3ac0fb5f",  
      "HealthCheckEnabled": true,  
      "HealthCheckProtocol": "TCP",  
      "HealthCheckPort": "traffic-port",  
      "HealthCheckIntervalSeconds": 30,  
      "HealthCheckTimeoutSeconds": 10,  
      "HealthyThresholdCount": 5,  
      "UnhealthyThresholdCount": 2,  
      "TargetType": "ip",  
      "IpAddressType": "ipv4"  
    }  
  ]  
}
```

詳細については、「Network Load Balancer ユーザーガイド」の [「ターゲットグループの作成」](#) を参照してください。

例 4: Network Load Balancer から Application Load Balancer にトラフィックをルーティングするターゲットグループを作成するには

次のcreate-target-group例では、Application Load Balancer をターゲットとして登録する Network Application Load Balancer のターゲットグループを作成します (ターゲットタイプは)alb。

```
aws elbv2 create-target-group --name my-alb-target --protocol TCP --port 80 --target-type alb --vpc-id vpc-3ac0fb5f
```

出力:

```
{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-alb-target/a3003e085dbb8ddc",
      "TargetGroupName": "my-alb-target",
      "Protocol": "TCP",
      "Port": 80,
      "VpcId": "vpc-838475fe",
      "HealthCheckProtocol": "HTTP",
      "HealthCheckPort": "traffic-port",
      "HealthCheckEnabled": true,
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 6,
      "HealthyThresholdCount": 5,
      "UnhealthyThresholdCount": 2,
      "HealthCheckPath": "/",
      "Matcher": {
        "HttpCode": "200-399"
      },
      "TargetType": "alb",
      "IpAddressType": "ipv4"
    }
  ]
}
```

詳細については、「Network [Application Load Balancer ユーザーガイド](#)」の「[Application Load Balancer をターゲットとするターゲットグループを作成する](#)」を参照してください。

例 5: Gateway Load Balancer のターゲットグループを作成するには

次のcreate-target-group例では、Gateway Load Balancer のターゲットグループを作成します。ターゲットはインスタンスで、ターゲットグループプロトコルは ですGENEVE。

```
aws elbv2 create-target-group \  
  --name my-glb-targetgroup \  
  --protocol GENEVE \  
  --port 6081 \  
  --target-type instance \  
  --vpc-id vpc-838475fe
```

出力:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-glb-targetgroup/00c3d57eacd6f40b6f",  
      "TargetGroupName": "my-glb-targetgroup",  
      "Protocol": "GENEVE",  
      "Port": 6081,  
      "VpcId": "vpc-838475fe",  
      "HealthCheckProtocol": "TCP",  
      "HealthCheckPort": "80",  
      "HealthCheckEnabled": true,  
      "HealthCheckIntervalSeconds": 10,  
      "HealthCheckTimeoutSeconds": 5,  
      "HealthyThresholdCount": 5,  
      "UnhealthyThresholdCount": 2,  
      "TargetType": "instance"  
    }  
  ]  
}
```

詳細については、「Gateway Load Balancer ユーザーガイド」の「ターゲットグループの作成」
<<https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/create-target-group.html>>`_`
を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateTargetGroup](#)」の「」を参照してください。

delete-listener

次のコード例は、delete-listener を使用する方法を示しています。

AWS CLI

リスナーを削除するには

次のdelete-listener例では、指定されたリスナーを削除します。

```
aws elbv2 delete-listener \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteListener](#)」の「」を参照してください。

delete-load-balancer

次の例は、delete-load-balancer を使用する方法を説明しています。

AWS CLI

ロードバランサーを削除するには

次の delete-load-balancer の例は、指定されたロードバランサーを削除します。

```
aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLoadBalancer](#)」の「」を参照してください。

delete-rule

次の例は、delete-rule を使用する方法を説明しています。

AWS CLI

ルールを削除するには

次のdelete-rule例では、指定されたルールを削除します。

```
aws elbv2 delete-rule \  
  --rule-id my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

```
--rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRule](#)」の「」を参照してください。

delete-target-group

次の例は、delete-target-group を使用する方法を説明しています。

AWS CLI

ターゲットグループを削除するには

次の delete-target-group の例は、指定されたターゲットグループを削除します。

```
aws elbv2 delete-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

このコマンドでは何も出力されません。

詳細については、Application Load [Balancer ガイドの「ロードバランサーの削除」](#)を参照してください。 Application Load Balancer

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTargetGroup](#)」の「」を参照してください。

deregister-targets

次の例は、deregister-targets を使用する方法を説明しています。

AWS CLI

例 1: ターゲットグループからターゲットの登録を解除するには

次のderegister-targets例では、指定されたターゲットグループから指定されたインスタンスを削除します。

```
aws elbv2 deregister-targets \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \  
  --instances ...
```

```
--targets Id=i-1234567890abcdef0
```

例 2: ポートオーバーライドを使用して登録されたターゲットの登録を解除するには

次のderegister-targets例では、ポートオーバーライドを使用して登録されたターゲットグループからインスタンスを削除します。

```
aws elbv2 deregister-targets \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \  
  --targets Id=i-1234567890abcdef0,Port=80 Id=i-1234567890abcdef0,Port=766
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterTargets](#)」の「」を参照してください。

describe-account-limits

次のコード例は、describe-account-limits を使用する方法を示しています。

AWS CLI

Elastic Load Balancing の制限を記述するには

次のdescribe-account-limits例は、現在のリージョンの AWS アカウントの Elastic Load Balancing 制限を示しています。

```
aws elbv2 describe-account-limits
```

出力:

```
{  
  "Limits": [  
    {  
      "Name": "target-groups",  
      "Max": "3000"  
    },  
    {  
      "Name": "targets-per-application-load-balancer",  
      "Max": "1000"  
    },  
    {  
      "Name": "listeners-per-application-load-balancer",
```

```
    "Max": "50"
  },
  {
    "Name": "rules-per-application-load-balancer",
    "Max": "100"
  },
  {
    "Name": "network-load-balancers",
    "Max": "50"
  },
  {
    "Name": "targets-per-network-load-balancer",
    "Max": "3000"
  },
  {
    "Name": "targets-per-availability-zone-per-network-load-balancer",
    "Max": "500"
  },
  {
    "Name": "listeners-per-network-load-balancer",
    "Max": "50"
  },
  {
    "Name": "condition-values-per-alb-rule",
    "Max": "5"
  },
  {
    "Name": "condition-wildcards-per-alb-rule",
    "Max": "5"
  },
  {
    "Name": "target-groups-per-application-load-balancer",
    "Max": "100"
  },
  {
    "Name": "target-groups-per-action-on-application-load-balancer",
    "Max": "5"
  },
  {
    "Name": "target-groups-per-action-on-network-load-balancer",
    "Max": "1"
  },
  {
    "Name": "certificates-per-application-load-balancer",
```

```
    "Max": "25"
  },
  {
    "Name": "certificates-per-network-load-balancer",
    "Max": "25"
  },
  {
    "Name": "targets-per-target-group",
    "Max": "1000"
  },
  {
    "Name": "target-id-registrations-per-application-load-balancer",
    "Max": "1000"
  },
  {
    "Name": "network-load-balancer-enis-per-vpc",
    "Max": "1200"
  },
  {
    "Name": "application-load-balancers",
    "Max": "50"
  },
  {
    "Name": "gateway-load-balancers",
    "Max": "100"
  },
  {
    "Name": "gateway-load-balancers-per-vpc",
    "Max": "100"
  },
  {
    "Name": "geneve-target-groups",
    "Max": "100"
  },
  {
    "Name": "targets-per-availability-zone-per-gateway-load-balancer",
    "Max": "300"
  }
]
}
```

詳細については、AWS 「全般のリファレンス」の [「クォータ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountLimits](#)」の「」を参照してください。

describe-listener-certificates

次の例は、describe-listener-certificates を使用する方法を説明しています。

AWS CLI

セキュアリスナーの証明書を記述するには

この例では、指定されたセキュアリスナーの証明書について説明します。

コマンド:

```
aws elbv2 describe-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

出力:

```
{
  "Certificates": [
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
      "IsDefault": false
    },
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557",
      "IsDefault": false
    },
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/fe59da96-6f58-4a22-8eed-6d0d50477e1d",
      "IsDefault": true
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeListenerCertificates](#)」の「」を参照してください。

describe-listeners

次の例は、describe-listeners を使用する方法を説明しています。

AWS CLI

リスナーを記述するには

この例では、指定されたリスナーについて説明します。

コマンド:

```
aws elbv2 describe-listeners --listener-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

出力:

```
{
  "Listeners": [
    {
      "Port": 80,
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
    }
  ]
}
```

ロードバランサーのリスナーを記述するには

この例では、指定されたロードバランサーのリスナーについて説明します。

コマンド:

```
aws elbv2 describe-listeners --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

出力:

```
{
  "Listeners": [
    {
      "Port": 443,
      "Protocol": "HTTPS",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
      "SslPolicy": "ELBSecurityPolicy-2015-05",
      "Certificates": [
        {
          "CertificateArn": "arn:aws:iam::123456789012:server-certificate/my-server-cert"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65"
    },
    {
      "Port": 80,
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
    }
  ],
}
```



```
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",  
    "ListenerArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeListeners](#)」の「」を参照してください。

describe-load-balancer-attributes

次の例は、describe-load-balancer-attributes を使用する方法を説明しています。

AWS CLI

ロードバランサー属性を記述するには

次のdescribe-load-balancer-attributes例では、指定されたロードバランサーの属性を表示します。

```
aws elbv2 describe-load-balancer-attributes \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

次の出力例は、Application Load Balancer の属性を示しています。

```
{  
  "Attributes": [  
    {  
      "Value": "false",  
      "Key": "access_logs.s3.enabled"  
    },  
    {  
      "Value": "",  
      "Key": "access_logs.s3.bucket"  
    },  
    {  
      "Value": "",  
      "Key": "access_logs.s3.prefix"  
    }  
  ]  
}
```

```
    },
    {
      "Value": "60",
      "Key": "idle_timeout.timeout_seconds"
    },
    {
      "Value": "false",
      "Key": "deletion_protection.enabled"
    },
    {
      "Value": "true",
      "Key": "routing.http2.enabled"
    }
  ]
}
```

次の出力例には、Network Load Balancer の属性が含まれています。

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.bucket"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "false",
      "Key": "deletion_protection.enabled"
    },
    {
      "Value": "false",
      "Key": "load_balancing.cross_zone.enabled"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLoadBalancerAttributes](#)」の「」を参照してください。

describe-load-balancers

次のコード例は、describe-load-balancers を使用する方法を示しています。

AWS CLI

ロードバランサーを記述するには

この例は、指定されたロードバランサーを記述します。

コマンド:

```
aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

出力:

```
{
  "LoadBalancers": [
    {
      "Type": "application",
      "Scheme": "internet-facing",
      "IpAddressType": "ipv4",
      "VpcId": "vpc-3ac0fb5f",
      "AvailabilityZones": [
        {
          "ZoneName": "us-west-2a",
          "SubnetId": "subnet-8360a9e7"
        },
        {
          "ZoneName": "us-west-2b",
          "SubnetId": "subnet-b7d581c0"
        }
      ],
      "CreatedTime": "2016-03-25T21:26:12.920Z",
      "CanonicalHostedZoneId": "Z2P70J7EXAMPLE",
      "DNSName": "my-load-balancer-424835706.us-west-2.elb.amazonaws.com",
      "SecurityGroups": [
```

```
        "sg-5943793c"  
    ],  
    "LoadBalancerName": "my-load-balancer",  
    "State": {  
        "Code": "active"  
    },  
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188"  
    }  
]  
}
```

すべてのロードバランサーを記述するには

この例は、すべてのロードバランサーを記述します。

コマンド:

```
aws elbv2 describe-load-balancers
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLoadBalancers](#)」の「」を参照してください。

describe-rules

次の例は、describe-rules を使用する方法を説明しています。

AWS CLI

例 1: ルールを記述するには

次のdescribe-rules例では、指定されたルールの詳細を表示します。

```
aws elbv2 describe-rules \  
  --rule-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/  
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

例 2: リスナーのルールを記述するには

次のdescribe-rules例では、指定されたリスナーのルールの詳細を表示します。出力には、デフォルトのルールと、追加したその他のルールが含まれます。

```
aws elbv2 describe-rules \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRules](#)」の「」を参照してください。

describe-ssl-policies

次の例は、describe-ssl-policies を使用する方法を説明しています。

AWS CLI

例 1: SSLネゴシエーションに使用されるポリシーをロードバランサータイプ別に一覧表示するには

次のdescribe-ssl-policies例は、Application Load Balancer SSL とのネゴシエーションに使用できるポリシーの名前を示しています。この例では、--queryパラメータを使用してポリシーの名前のみを表示します。

```
aws elbv2 describe-ssl-policies \  
  --load-balancer-type application \  
  --query SslPolicies[*].Name
```

出力:

```
[  
  "ELBSecurityPolicy-2016-08",  
  "ELBSecurityPolicy-TLS13-1-2-2021-06",  
  "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",  
  "ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06",  
  "ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06",  
  "ELBSecurityPolicy-TLS13-1-1-2021-06",  
  "ELBSecurityPolicy-TLS13-1-0-2021-06",  
  "ELBSecurityPolicy-TLS13-1-3-2021-06",  
  "ELBSecurityPolicy-TLS-1-2-2017-01",  
  "ELBSecurityPolicy-TLS-1-1-2017-01",  
  "ELBSecurityPolicy-TLS-1-2-Ext-2018-06",  
  "ELBSecurityPolicy-FS-2018-06",  
  "ELBSecurityPolicy-2015-05",  
  "ELBSecurityPolicy-TLS-1-0-2015-04",
```

```
"ELBSecurityPolicy-FS-1-2-Res-2019-08",
"ELBSecurityPolicy-FS-1-1-2019-08",
"ELBSecurityPolicy-FS-1-2-2019-08",
"ELBSecurityPolicy-FS-1-2-Res-2020-10"
]
```

例 2: 特定のプロトコルをサポートするポリシーを一覧表示するには

次のdescribe-ssl-policies例は、1.3 TLS プロトコルをサポートするポリシーの名前を示しています。この例では、--queryパラメータを使用してポリシーの名前のみを表示します。

```
aws elbv2 describe-ssl-policies \
  --load-balancer-type application \
  --query SslPolicies[?contains(SslProtocols,'TLSv1.3')].Name
```

出力:

```
[
"ELBSecurityPolicy-TLS13-1-2-2021-06",
"ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
"ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06",
"ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06",
"ELBSecurityPolicy-TLS13-1-1-2021-06",
"ELBSecurityPolicy-TLS13-1-0-2021-06",
"ELBSecurityPolicy-TLS13-1-3-2021-06"
]
```

例 3: ポリシーの暗号を表示するには

次のdescribe-ssl-policies例では、指定されたポリシーの暗号の名前を表示します。この例では、--queryパラメータを使用して暗号名のみを表示します。リスト内の最初の暗号の優先度は 1 で、残りの暗号は優先順位が付けられています。

```
aws elbv2 describe-ssl-policies \
  --names ELBSecurityPolicy-TLS13-1-2-2021-06 \
  --query SslPolicies[*].Ciphers[*].Name
```

出力:

```
[
```

```
"TLS_AES_128_GCM_SHA256",  
"TLS_AES_256_GCM_SHA384",  
"TLS_CHACHA20_POLY1305_SHA256",  
"ECDHE-ECDSA-AES128-GCM-SHA256",  
"ECDHE-RSA-AES128-GCM-SHA256",  
"ECDHE-ECDSA-AES128-SHA256",  
"ECDHE-RSA-AES128-SHA256",  
"ECDHE-ECDSA-AES256-GCM-SHA384",  
"ECDHE-RSA-AES256-GCM-SHA384",  
"ECDHE-ECDSA-AES256-SHA384",  
"ECDHE-RSA-AES256-SHA384"
```

```
]
```

詳細については、「Application Load Balancer ユーザーガイド」の「[セキュリティポリシー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSslPolicies](#)」の「」を参照してください。

describe-tags

次のコード例は、describe-tags を使用する方法を示しています。

AWS CLI

ロードバランサーに割り当てられたタグを記述するには

この例では、指定されたロードバランサーに割り当てられたタグについて説明します。

コマンド:

```
aws elbv2 describe-tags --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

出力:

```
{  
  "TagDescriptions": [  
    {  
      "ResourceArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
```

```
    "Tags": [
      {
        "Value": "lima",
        "Key": "project"
      },
      {
        "Value": "digital-media",
        "Key": "department"
      }
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

describe-target-group-attributes

次のコード例は、describe-target-group-attributes を使用する方法を示しています。

AWS CLI

ターゲットグループ属性を記述するには

次のdescribe-target-group-attributes例では、指定されたターゲットグループの属性を表示します。

```
aws elbv2 describe-target-group-attributes \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

プロトコルが HTTP または HTTPS で、ターゲットタイプが instance または の場合、出力には属性が含まれます ip。

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "stickiness.enabled"
    },
  ],
}
```



```
{
  "Value": "300",
  "Key": "deregistration_delay.timeout_seconds"
},
{
  "Value": "lb_cookie",
  "Key": "stickiness.type"
},
{
  "Value": "86400",
  "Key": "stickiness.lb_cookie.duration_seconds"
},
{
  "Value": "0",
  "Key": "slow_start.duration_seconds"
}
]
```

次の出力には、プロトコルが HTTP または HTTPS で、ターゲットタイプが の場合の属性が含まれます lambda。

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "lambda.multi_value_headers.enabled"
    }
  ]
}
```

次の出力には、プロトコルが TCP、TLS、または TCP_ の場合 UDP の属性が含まれます UDP。

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "proxy_protocol_v2.enabled"
    },
    {
      "Value": "300",
      "Key": "deregistration_delay.timeout_seconds"
    }
  ]
}
```

```
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTargetGroupAttributes](#)」の「」を参照してください。

describe-target-groups

次のコード例は、describe-target-groups を使用する方法を示しています。

AWS CLI

例 1: ターゲットグループを記述するには

次の describe-target-groups の例は、指定されたターゲットグループの詳細情報を表示します。

```
aws elbv2 describe-target-groups \
  --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

出力:

```
{
  "TargetGroups": [
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
      "TargetGroupName": "my-targets",
      "Protocol": "HTTP",
      "Port": 80,
      "VpcId": "vpc-3ac0fb5f",
      "HealthCheckProtocol": "HTTP",
      "HealthCheckPort": "traffic-port",
      "HealthCheckEnabled": true,
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 5,
      "HealthyThresholdCount": 5,
      "UnhealthyThresholdCount": 2,
      "HealthCheckPath": "/",
      "Matcher": {
```

```
        "HttpCode": "200"
      },
      "LoadBalancerArns": [
        "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/my-load-balancer/50dc6c495c0c9188"
      ],
      "TargetType": "instance",
      "ProtocolVersion": "HTTP1",
      "IpAddressType": "ipv4"
    }
  ]
}
```

例 2: ロードバランサーのすべてのターゲットグループを記述するには

次の `describe-target-groups` の例は、指定されたロードバランサーのすべてのターゲットグループの詳細情報を表示します。この例では、`--query`パラメータを使用してターゲットグループ名のみを表示します。

```
aws elbv2 describe-target-groups \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
  --query TargetGroups[*].TargetGroupName
```

出力:

```
[
  "my-instance-targets",
  "my-ip-targets",
  "my-lambda-target"
]
```

詳細については、「アプリケーションロードバランサーガイド」の「[ターゲットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTargetGroups](#)」の「」を参照してください。

describe-target-health

次のコード例は、`describe-target-health` を使用する方法を示しています。

AWS CLI

例 1: ターゲットグループのターゲットのヘルスを記述するには

次の `describe-target-health` の例は、指定されたターゲットグループのターゲットのヘルスの詳細情報を表示します。これらのターゲットは正常です。

```
aws elbv2 describe-target-health \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

出力:

```
{  
  "TargetHealthDescriptions": [  
    {  
      "HealthCheckPort": "80",  
      "Target": {  
        "Id": "i-ceddc4d",  
        "Port": 80  
      },  
      "TargetHealth": {  
        "State": "healthy"  
      }  
    },  
    {  
      "HealthCheckPort": "80",  
      "Target": {  
        "Id": "i-0f76fade",  
        "Port": 80  
      },  
      "TargetHealth": {  
        "State": "healthy"  
      }  
    }  
  ]  
}
```

例 2: ターゲットのヘルスを記述するには

次の `describe-target-health` の例は、指定されたターゲットのヘルスの詳細情報を表示します。このターゲットは正常です。

```
aws elbv2 describe-target-health \  
  --targets Id=i-0f76fade,Port=80 \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

出力:

```
{  
  "TargetHealthDescriptions": [  
    {  
      "HealthCheckPort": "80",  
      "Target": {  
        "Id": "i-0f76fade",  
        "Port": 80  
      },  
      "TargetHealth": {  
        "State": "healthy"  
      }  
    }  
  ]  
}
```

次の出力例は、リスナーのアクションでターゲットグループが指定されていないターゲットの出力です。このターゲットはロードバランサーからのトラフィックを受信できません。

```
{  
  "TargetHealthDescriptions": [  
    {  
      "HealthCheckPort": "80",  
      "Target": {  
        "Id": "i-0f76fade",  
        "Port": 80  
      },  
      "TargetHealth": {  
        "State": "unused",  
        "Reason": "Target.NotInUse",  
        "Description": "Target group is not configured to receive traffic  
from the load balancer"  
      }  
    }  
  ]  
}
```

次の例は、リスナーのアクションでターゲットグループが指定されたばかりのターゲットの出力です。このターゲットは現在登録中です。

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "initial",
        "Reason": "Elb.RegistrationInProgress",
        "Description": "Target registration is in progress"
      }
    }
  ]
}
```

次の例は、正常でないターゲットの出力です。

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "unhealthy",
        "Reason": "Target.Timeout",
        "Description": "Connection to target timed out"
      }
    }
  ]
}
```

以下は、ターゲットが Lambda 関数で、ヘルスチェックが無効化されている出力の例です。

```
{
```

```
"TargetHealthDescriptions": [
  {
    "Target": {
      "Id": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
      "AvailabilityZone": "all",
    },
    "TargetHealth": {
      "State": "unavailable",
      "Reason": "Target.HealthCheckDisabled",
      "Description": "Health checks are not enabled for this target"
    }
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTargetHealth](#)」の「」を参照してください。

modify-listener

次のコード例は、modify-listener を使用する方法を示しています。

AWS CLI

例 1: デフォルトのアクションを転送アクションに変更するには

次のmodify-listener例では、指定されたリスナーのデフォルトアクション (フォワードアクション) を変更します。

```
aws elbv2 modify-listener \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f
```

出力:

```
{
  "Listeners": [
    {
      "Protocol": "HTTP",
```

```

    "DefaultActions": [
      {
        "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f",
        "Type": "forward"
      }
    ],
    "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
    "Port": 80,
    "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
  }
]
}

```

例 2: デフォルトのアクションをリダイレクトアクションに変更するには

次のmodify-listener例では、デフォルトのアクションを指定されたリスナーのリダイレクトアクションに変更します。

```

aws elbv2 modify-listener \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
  --default-actions Type=redirect,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f

```

出力:

```

{
  "Listeners": [
    {
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f",
          "Type": "redirect"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",

```



```
    "Port": 80,
    "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2"
  }
]
}
```

例 3: サーバー証明書を変更するには

この例では、指定されたHTTPSリスナーのサーバー証明書を変更します。

```
aws elbv2 modify-listener \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65 \
  --certificates CertificateArn=arn:aws:iam::123456789012:server-certificate/my-
new-server-cert
```

出力:

```
{
  "Listeners": [
    {
      "Protocol": "HTTPS",
      "DefaultActions": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
          "Type": "forward"
        }
      ],
      "SslPolicy": "ELBSecurityPolicy-2015-05",
      "Certificates": [
        {
          "CertificateArn": "arn:aws:iam::123456789012:server-certificate/
my-new-server-cert"
        }
      ],
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
      "Port": 443,
      "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65"
    }
  ]
}
```

```
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyListener](#)」の「」を参照してください。

modify-load-balancer-attributes

次の例は、`modify-load-balancer-attributes` を使用する方法を説明しています。

AWS CLI

削除保護を有効にするには

この例では、指定されたロードバランサーの削除保護を有効にします。

コマンド:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --attributes Key=deletion_protection.enabled,Value=true
```

出力:

```
{
  "Attributes": [
    {
      "Value": "true",
      "Key": "deletion_protection.enabled"
    },
    {
      "Value": "false",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "60",
      "Key": "idle_timeout.timeout_seconds"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.prefix"
    }
  ]
}
```

```
    },
    {
      "Value": "",
      "Key": "access_logs.s3.bucket"
    }
  ]
}
```

アイドルタイムアウトを変更するには

この例では、指定されたロードバランサーのアイドルタイムアウト値を変更します。

コマンド:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188 --attributes Key=idle_timeout.timeout_seconds,Value=30
```

出力:

```
{
  "Attributes": [
    {
      "Value": "30",
      "Key": "idle_timeout.timeout_seconds"
    },
    {
      "Value": "false",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "true",
      "Key": "deletion_protection.enabled"
    },
    {
      "Value": "",
      "Key": "access_logs.s3.bucket"
    }
  ]
}
```

```
}
```

アクセスログを有効にするには

この例では、指定されたロードバランサーのアクセスログを有効にします。S3 バケットはロードバランサーと同じリージョンに存在し、Elastic Load Balancing サービスへのアクセスを許可するポリシーがアタッチされている必要があることに注意してください。

コマンド:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --attributes Key=access_logs.s3.enabled,Value=true Key=access_logs.s3.bucket,Value=my-loadbalancer-logs Key=access_logs.s3.prefix,Value=myapp
```

出力:

```
{
  "Attributes": [
    {
      "Value": "true",
      "Key": "access_logs.s3.enabled"
    },
    {
      "Value": "my-load-balancer-logs",
      "Key": "access_logs.s3.bucket"
    },
    {
      "Value": "myapp",
      "Key": "access_logs.s3.prefix"
    },
    {
      "Value": "60",
      "Key": "idle_timeout.timeout_seconds"
    },
    {
      "Value": "false",
      "Key": "deletion_protection.enabled"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyLoadBalancerAttributes](#)」の「」を参照してください。

modify-rule

次の例は、modify-rule を使用する方法を説明しています。

AWS CLI

ルールを変更するには

次のmodify-rule例では、指定されたルールのアクションと条件を更新します。

```
aws elbv2 modify-rule \  
  --actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \  
  --conditions Field=path-pattern,Values='/images/*' \  
  --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

出力:

```
{  
  "Rules": [  
    {  
      "Priority": "10",  
      "Conditions": [  
        {  
          "Field": "path-pattern",  
          "Values": [  
            "/images/*"  
          ]  
        }  
      ],  
      "RuleArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee",  
      "IsDefault": false,  
      "Actions": [  
        {  
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
```

```
    "Type": "forward"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyRule](#)」の「」を参照してください。

modify-target-group-attributes

次の例は、`modify-target-group-attributes` を使用する方法を説明しています。

AWS CLI

登録解除遅延タイムアウトを変更するには

この例では、登録解除遅延タイムアウトを、指定されたターゲットグループの指定された値に設定します。

コマンド:

```
aws elbv2 modify-target-group-attributes --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 --attributes Key=deregistration_delay.timeout_seconds,Value=600
```

出力:

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "stickiness.enabled"
    },
    {
      "Value": "600",
      "Key": "deregistration_delay.timeout_seconds"
    },
    {
      "Value": "lb_cookie",
```

```
    "Key": "stickiness.type"
  },
  {
    "Value": "86400",
    "Key": "stickiness.lb_cookie.duration_seconds"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyTargetGroupAttributes](#)」の「」を参照してください。

modify-target-group

次の例は、modify-target-group を使用する方法を説明しています。

AWS CLI

ターゲットグループのヘルスチェック設定を変更するには

次のmodify-target-group例では、指定されたターゲットグループのターゲットのヘルスを評価するために使用されるヘルスチェックの設定を変更します。カンマをCLI解析する方法により、二重引用符ではなく一重引用符で--matcherオプションの範囲を囲む必要があることに注意してください。

```
aws elbv2 modify-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f \  
  --health-check-protocol HTTPS \  
  --health-check-port 443 \  
  --matcher HttpCode='200,299'
```

出力:

```
{  
  "TargetGroups": [  
    {  
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-https-targets/2453ed029918f21f",  
      "TargetGroupName": "my-https-targets",  
      "Protocol": "HTTPS",
```

```
    "Port": 443,
    "VpcId": "vpc-3ac0fb5f",
    "HealthCheckProtocol": "HTTPS",
    "HealthCheckPort": "443",
    "HealthCheckEnabled": true,
    "HealthCheckIntervalSeconds": 30,
    "HealthCheckTimeoutSeconds": 5,
    "HealthyThresholdCount": 5,
    "UnhealthyThresholdCount": 2,
    "Matcher": {
      "HttpCode": "200,299"
    },
    "LoadBalancerArns": [
      "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/my-load-balancer/50dc6c495c0c9188"
    ],
    "TargetType": "instance",
    "ProtocolVersion": "HTTP1",
    "IpAddressType": "ipv4"
  }
]
}
```

詳細については、「アプリケーションロードバランサーガイド」の「[ターゲットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyTargetGroup](#)」の「」を参照してください。

register-targets

次の例は、register-targets を使用する方法を説明しています。

AWS CLI

例 1: インスタンス ID でターゲットグループにターゲットを登録するには

次のregister-targets例では、指定されたインスタンスをターゲットグループに登録します。ターゲットグループにはターゲットタイプが必要でinstance。

```
aws elbv2 register-targets \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
```



```
--targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

例 2: ポートオーバーライドを使用してターゲットをターゲットグループに登録するには

次のregister-targets例では、複数のポートを使用して、指定されたインスタンスをターゲットグループに登録します。これにより、ターゲットグループのターゲットと同じインスタンスにコンテナを登録できます。

```
aws elbv2 register-targets \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-internal-targets/3bb63f11dfb0faf9 \  
  --targets Id=i-0598c7d356eba48d7,Port=80 Id=i-0598c7d356eba48d7,Port=766
```

例 3: ターゲットを IP アドレスでターゲットグループに登録するには

次のregister-targets例では、指定された IP アドレスをターゲットグループに登録します。ターゲットグループにはターゲットタイプが必要ですip。

```
aws elbv2 register-targets \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \  
  --targets Id=10.0.1.15 Id=10.0.1.23
```

例 4: Lambda 関数をターゲットとして登録するには

次のregister-targets例では、指定された IP アドレスをターゲットグループに登録します。ターゲットグループにはターゲットタイプが必要ですlambda。Lambda 関数を呼び出すには、Elastic Load Balancing アクセス許可を付与する必要があります。

```
aws elbv2 register-targets \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-ip-targets/8518e899d173178f \  
  --targets Id=arn:aws:lambda:us-west-2:123456789012:function:my-function
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterTargets](#)」の「」を参照してください。

remove-listener-certificates

次のコード例は、remove-listener-certificates を使用する方法を示しています。

AWS CLI

セキュアリスナーから証明書を削除するには

この例では、指定されたセキュアリスナーから指定された証明書を削除します。

コマンド:

```
aws elbv2 remove-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 --certificates CertificateArn=arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveListenerCertificates](#)」の「」を参照してください。

remove-tags

次のコード例は、remove-tags を使用する方法を示しています。

AWS CLI

ロードバランサーからタグを削除するには

次のremove-tags例では、指定されたロードバランサーから project と department タグを削除します。

```
aws elbv2 remove-tags \  
  --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --tag-keys project department
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveTags](#)」の「」を参照してください。

set-ip-address-type

次の例は、set-ip-address-type を使用する方法を説明しています。

AWS CLI

ロードバランサーのアドレスタイプを設定するには

この例では、指定されたロードバランサーのアドレスタイプを `dualstack` に設定します。ロードバランサーサブネットには、関連付けられたIPv6CIDRブロックが必要です。

コマンド:

```
aws elbv2 set-ip-address-type --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --ip-address-type dualstack
```

出力:

```
{
  "IpAddressType": "dualstack"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [SetIpAddressType](#)」の「」を参照してください。

set-rule-priorities

次のコード例は、`set-rule-priorities` を使用する方法を示しています。

AWS CLI

ルールの優先度を設定するには

この例では、指定されたルールの優先度を設定します。

コマンド:

```
aws elbv2 set-rule-priorities --rule-priorities RuleArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3,Priority=5
```

出力:

```
{
```

```
"Rules": [
  {
    "Priority": "5",
    "Conditions": [
      {
        "Field": "path-pattern",
        "Values": [
          "/img/*"
        ]
      }
    ],
    "RuleArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-
rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3",
    "IsDefault": false,
    "Actions": [
      {
        "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
        "Type": "forward"
      }
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetRulePriorities](#)」の「」を参照してください。

set-security-groups

次のコード例は、set-security-groups を使用する方法を示しています。

AWS CLI

セキュリティグループをロードバランサーに関連付けるには

この例では、指定されたセキュリティグループを指定されたロードバランサーに関連付けます。

コマンド:

```
aws elbv2 set-security-groups --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --security-
groups sg-5943793c
```

出力:

```
{
  "SecurityGroupIds": [
    "sg-5943793c"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetSecurityGroups](#)」の「」を参照してください。

set-subnets

次の例は、set-subnets を使用する方法を説明しています。

AWS CLI

ロードバランサーのアベイラビリティゾーンを有効にするには

この例では、指定されたロードバランサーの指定されたサブネットのアベイラビリティゾーンを有効にします。

コマンド:

```
aws elbv2 set-subnets --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 --subnets subnet-8360a9e7 subnet-b7d581c0
```

出力:

```
{
  "AvailabilityZones": [
    {
      "SubnetId": "subnet-8360a9e7",
      "ZoneName": "us-west-2a"
    },
    {
      "SubnetId": "subnet-b7d581c0",
      "ZoneName": "us-west-2b"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetSubnets](#)」の「」を参照してください。

を使用した Elastic Transcoder の例 AWS CLI

次のコード例は、Elastic Transcoder AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

cancel-job

次のコード例は、cancel-job を使用する方法を示しています。

AWS CLI

のジョブをキャンセルするには ElasticTranscoder

これにより、の指定されたジョブがキャンセルされます ElasticTranscoder。

コマンド:

```
aws elastictranscoder cancel-job --id 333333333333-abcde3
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelJob](#)」の「」を参照してください。

create-job

次の例は、create-job を使用する方法を説明しています。

AWS CLI

のジョブを作成するには ElasticTranscoder

次のcreate-job例では、 のジョブを作成します ElasticTranscoder。

```
aws elastictranscoder create-job \  
  --pipeline-id 111111111111-abcde1 \  
  --inputs file://inputs.json \  
  --outputs file://outputs.json \  
  --output-key-prefix "recipes/" \  
  --user-metadata file://user-metadata.json
```

inputs.json の内容:

```
[{  
  "Key": "ETS_example_file.mp4",  
  "FrameRate": "auto",  
  "Resolution": "auto",  
  "AspectRatio": "auto",  
  "Interlaced": "auto",  
  "Container": "mp4"  
}]
```

outputs.json の内容 :

```
[  
  {  
    "Key": "webm/ETS_example_file-kindlefirehd.webm",  
    "Rotate": "0",  
    "PresetId": "1351620000001-100250"  
  }  
]
```

user-metadata.json の内容:

```
{  
  "Food type": "Italian",  
  "Cook book": "recipe notebook"  
}
```

出力:

```
{
  "Job": {
    "Status": "Submitted",
    "Inputs": [
      {
        "Container": "mp4",
        "FrameRate": "auto",
        "Key": "ETS_example_file.mp4",
        "AspectRatio": "auto",
        "Resolution": "auto",
        "Interlaced": "auto"
      }
    ],
    "Playlists": [],
    "Outputs": [
      {
        "Status": "Submitted",
        "Rotate": "0",
        "PresetId": "1351620000001-100250",
        "Watermarks": [],
        "Key": "webm/ETS_example_file-kindlefirehd.webm",
        "Id": "1"
      }
    ],
    "PipelineId": "3333333333333-abcde3",
    "OutputKeyPrefix": "recipes/",
    "UserMetadata": {
      "Cook book": "recipe notebook",
      "Food type": "Italian"
    },
    "Output": {
      "Status": "Submitted",
      "Rotate": "0",
      "PresetId": "1351620000001-100250",
      "Watermarks": [],
      "Key": "webm/ETS_example_file-kindlefirehd.webm",
      "Id": "1"
    },
    "Timing": {
      "SubmitTimeMillis": 1533838012298
    },
    "Input": {
```



```

        "Container": "mp4",
        "FrameRate": "auto",
        "Key": "ETS_example_file.mp4",
        "AspectRatio": "auto",
        "Resolution": "auto",
        "Interlaced": "auto"
    },
    "Id": "1533838012294-example",
    "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
    }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateJob](#)」の「」を参照してください。

create-pipeline

次の例は、create-pipeline を使用する方法を説明しています。

AWS CLI

のパイプラインを作成するには ElasticTranscoder

次のcreate-pipeline例では、のパイプラインを作成します ElasticTranscoder。

```

aws elastictranscoder create-pipeline \
  --name Default \
  --input-bucket salesoffice.example.com-source \
  --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \
  --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-
east-1:111222333444:ETS_Errors \
  --content-config file://content-config.json \
  --thumbnail-config file://thumbnail-config.json

```

content-config.json の内容:

```

{
  "Bucket": "salesoffice.example.com-public-promos",
  "Permissions": [
    {

```

```
        "GranteeType": "Email",
        "Grantee": "marketing-promos@example.com",
        "Access": [
            "FullControl"
        ]
    },
    ],
    "StorageClass": "Standard"
}
```

thumbnail-config.json の内容:

```
{
  "Bucket": "salesoffice.example.com-public-promos-thumbnails",
  "Permissions": [
    {
      "GranteeType": "Email",
      "Grantee": "marketing-promos@example.com",
      "Access": [
        "FullControl"
      ]
    }
  ],
  "StorageClass": "ReducedRedundancy"
}
```

出力:

```
{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "salesoffice.example.com-public-promos",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    }
  }
}
```

```
    },
    "Name": "Default",
    "ThumbnailConfig": {
      "Bucket": "salesoffice.example.com-public-promos-thumbnails",
      "StorageClass": "ReducedRedundancy",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    },
    "Notifications": {
      "Completed": "",
      "Warning": "",
      "Progressing": "",
      "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
    },
    "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
    "InputBucket": "salesoffice.example.com-source",
    "Id": "1533765810590-example",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1533765810590-example"
  },
  "Warnings": [
    {
      "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
      "Code": "6006"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePipeline](#)」の「」を参照してください。

create-preset

次のコード例は、create-preset を使用する方法を示しています。

AWS CLI

のプリセットを作成するには ElasticTranscoder

次のcreate-preset例では、のプリセットを作成します ElasticTranscoder。

```
aws elastictranscoder create-preset \  
  --name DefaultPreset \  
  --description "Use for published videos" \  
  --container mp4 \  
  --video file://video.json \  
  --audio file://audio.json \  
  --thumbnails file://thumbnails.json
```

video.json の内容:

```
{  
  "Codec": "H.264",  
  "CodecOptions": {  
    "Profile": "main",  
    "Level": "2.2",  
    "MaxReferenceFrames": "3",  
    "MaxBitRate": "",  
    "BufferSize": "",  
    "InterlacedMode": "Progressive",  
    "ColorSpaceConversionMode": "None"  
  },  
  "KeyframesMaxDist": "240",  
  "FixedGOP": "false",  
  "BitRate": "1600",  
  "FrameRate": "auto",  
  "MaxFrameRate": "30",  
  "MaxWidth": "auto",  
  "MaxHeight": "auto",  
  "SizingPolicy": "Fit",  
  "PaddingPolicy": "Pad",  
  "DisplayAspectRatio": "auto",  
  "Watermarks": [  
    {  
      "Id": "company logo",
```

```
        "MaxWidth": "20%",
        "MaxHeight": "20%",
        "SizingPolicy": "ShrinkToFit",
        "HorizontalAlign": "Right",
        "HorizontalOffset": "10px",
        "VerticalAlign": "Bottom",
        "VerticalOffset": "10px",
        "Opacity": "55.5",
        "Target": "Content"
    }
]
}
```

audio.json の内容:

```
{
  "Codec": "AAC",
  "CodecOptions": {
    "Profile": "AAC-LC"
  },
  "SampleRate": "44100",
  "BitRate": "96",
  "Channels": "2"
}
```

thumbnails.json の内容:

```
{
  "Format": "png",
  "Interval": "120",
  "MaxWidth": "auto",
  "MaxHeight": "auto",
  "SizingPolicy": "Fit",
  "PaddingPolicy": "Pad"
}
```

出力:

```
{
  "Preset": {
    "Thumbnails": {
      "SizingPolicy": "Fit",
```

```
    "MaxWidth": "auto",
    "Format": "png",
    "PaddingPolicy": "Pad",
    "Interval": "120",
    "MaxHeight": "auto"
  },
  "Container": "mp4",
  "Description": "Use for published videos",
  "Video": {
    "SizingPolicy": "Fit",
    "MaxWidth": "auto",
    "PaddingPolicy": "Pad",
    "MaxFrameRate": "30",
    "FrameRate": "auto",
    "MaxHeight": "auto",
    "KeyframesMaxDist": "240",
    "FixedGOP": "false",
    "Codec": "H.264",
    "Watermarks": [
      {
        "SizingPolicy": "ShrinkToFit",
        "VerticalOffset": "10px",
        "VerticalAlign": "Bottom",
        "Target": "Content",
        "MaxWidth": "20%",
        "MaxHeight": "20%",
        "HorizontalAlign": "Right",
        "HorizontalOffset": "10px",
        "Opacity": "55.5",
        "Id": "company logo"
      }
    ]
  },
  "CodecOptions": {
    "Profile": "main",
    "MaxBitRate": "32",
    "InterlacedMode": "Progressive",
    "Level": "2.2",
    "ColorSpaceConversionMode": "None",
    "MaxReferenceFrames": "3",
    "BufferSize": "5"
  },
  "BitRate": "1600",
  "DisplayAspectRatio": "auto"
},
```

```
"Audio": {
  "Channels": "2",
  "CodecOptions": {
    "Profile": "AAC-LC"
  },
  "SampleRate": "44100",
  "Codec": "AAC",
  "BitRate": "96"
},
"Type": "Custom",
"Id": "1533765290724-example"
"Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1533765290724-example",
"Name": "DefaultPreset"
},
"Warning": ""
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePreset](#)」の「」を参照してください。

delete-pipeline

次の例は、delete-pipeline を使用する方法を説明しています。

AWS CLI

指定された ElasticTranscoder パイプラインを削除するには

これにより、指定された ElasticTranscoder パイプラインが削除されます。

コマンド:

```
aws elastictranscoder delete-pipeline --id 111111111111-abcde1
```

出力:

```
{
  "Success": "true"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePipeline](#)」の「」を参照してください。

delete-preset

次の例は、delete-preset を使用する方法を説明しています。

AWS CLI

指定された ElasticTranscoder プリセットを削除するには

これにより、指定された ElasticTranscoder プリセットが削除されます。

コマンド:

```
aws elastictranscoder delete-preset --id 555555555555-abcde5
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePreset](#)」の「」を参照してください。

list-jobs-by-pipeline

次の例は、list-jobs-by-pipeline を使用する方法を説明しています。

AWS CLI

指定されたパイプライン内の ElasticTranscoder ジョブのリストを取得するには

この例では、指定されたパイプライン内の ElasticTranscoder ジョブのリストを取得します。

コマンド:

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 111111111111-abcde1
```

出力:

```
{  
  "Jobs": []  
}
```


- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobsByPipeline](#)」の「」を参照してください。

list-jobs-by-status

次の例は、list-jobs-by-status を使用する方法を説明しています。

AWS CLI

ステータスが Complete の ElasticTranscoder ジョブのリストを取得するには

この例では、ステータスが Complete の ElasticTranscoder ジョブのリストを取得します。

コマンド:

```
aws elastictranscoder list-jobs-by-status --status Complete
```

出力:

```
{  
  "Jobs": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobsByStatus](#)」の「」を参照してください。

list-pipelines

次の例は、list-pipelines を使用する方法を説明しています。

AWS CLI

ElasticTranscoder パイプラインのリストを取得するには

この例では、ElasticTranscoder パイプラインのリストを取得します。

コマンド:

```
aws elastictranscoder list-pipelines
```

出力:

```
{
  "Pipelines": [
    {
      "Status": "Active",
      "ContentConfig": {
        "Bucket": "ets-example",
        "Permissions": []
      },
      "Name": "example-pipeline",
      "ThumbnailConfig": {
        "Bucket": "ets-example",
        "Permissions": []
      },
      "Notifications": {
        "Completed": "arn:aws:sns:us-west-2:123456789012:ets_example",
        "Warning": "",
        "Progressing": "",
        "Error": ""
      },
      "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
      "InputBucket": "ets-example",
      "OutputBucket": "ets-example",
      "Id": "3333333333333-abcde3",
      "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    {
      "Status": "Paused",
      "ContentConfig": {
        "Bucket": "ets-example",
        "Permissions": []
      },
      "Name": "example-php-test",
      "ThumbnailConfig": {
        "Bucket": "ets-example",
        "Permissions": []
      },
      "Notifications": {
        "Completed": "",
        "Warning": "",
        "Progressing": "",
        "Error": ""
      }
    }
  ]
}
```

```
    },
    "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
    "InputBucket": "ets-example",
    "OutputBucket": "ets-example",
    "Id": "3333333333333-abcde2",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde2"
  },
  {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-west-output",
      "Permissions": []
    },
  },
  "Name": "pipeline-west",
  "ThumbnailConfig": {
    "Bucket": "ets-west-output",
    "Permissions": []
  },
  "Notifications": {
    "Completed": "arn:aws:sns:us-west-2:123456789012:ets-notifications",
    "Warning": "",
    "Progressing": "",
    "Error": ""
  },
  "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
  "InputBucket": "ets-west-input",
  "OutputBucket": "ets-west-output",
  "Id": "3333333333333-abcde1",
  "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde1"
}
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPipelines](#)」の「」を参照してください。

list-presets

次のコード例は、list-presets を使用する方法を示しています。

AWS CLI

ElasticTranscoder プリセットのリストを取得するには

この例では、ElasticTranscoder プリセットのリストを取得します。

コマンド:

```
aws elastictranscoder list-presets --max-items 2
```

出力:

```
{
  "Presets": [
    {
      "Container": "mp4",
      "Name": "KindleFireHD-preset",
      "Video": {
        "Resolution": "1280x720",
        "FrameRate": "30",
        "KeyframesMaxDist": "90",
        "FixedGOP": "false",
        "Codec": "H.264",
        "Watermarks": [],
        "CodecOptions": {
          "Profile": "main",
          "MaxReferenceFrames": "3",
          "ColorSpaceConversionMode": "None",
          "InterlacedMode": "Progressive",
          "Level": "4"
        },
        "AspectRatio": "16:9",
        "BitRate": "2200"
      },
      "Audio": {
        "Channels": "2",
        "CodecOptions": {
          "Profile": "AAC-LC"
        },
        "SampleRate": "48000",
        "Codec": "AAC",
        "BitRate": "160"
      },
      "Type": "Custom",
    }
  ]
}
```

```
    "Id": "333333333333-abcde2",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/333333333333-abcde2",
    "Thumbnails": {
      "AspectRatio": "16:9",
      "Interval": "60",
      "Resolution": "192x108",
      "Format": "png"
    }
  },
  {
    "Thumbnails": {
      "AspectRatio": "16:9",
      "Interval": "60",
      "Resolution": "192x108",
      "Format": "png"
    },
    "Container": "mp4",
    "Description": "Custom preset for transcoding jobs",
    "Video": {
      "Resolution": "1280x720",
      "FrameRate": "30",
      "KeyframesMaxDist": "90",
      "FixedGOP": "false",
      "Codec": "H.264",
      "Watermarks": [],
      "CodecOptions": {
        "Profile": "main",
        "MaxReferenceFrames": "3",
        "ColorSpaceConversionMode": "None",
        "InterlacedMode": "Progressive",
        "Level": "3.1"
      },
      "AspectRatio": "16:9",
      "BitRate": "2200"
    },
    "Audio": {
      "Channels": "2",
      "CodecOptions": {
        "Profile": "AAC-LC"
      },
      "SampleRate": "44100",
      "Codec": "AAC",
      "BitRate": "160"
    }
  }
}
```

```
    },
    "Type": "Custom",
    "Id": "333333333333-abcde3",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/333333333333-abcde3",
    "Name": "Roman's Preset"
  }
],
"NextToken": "eyJQYWdlVG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPresets](#)」の「」を参照してください。

read-job

次の例は、read-job を使用する方法を説明しています。

AWS CLI

ElasticTranscoder ジョブを取得するには

この例では、指定された ElasticTranscoder ジョブを取得します。

コマンド:

```
aws elastictranscoder read-job --id 1533838012294-example
```

出力:

```
{
  "Job": {
    "Status": "Progressing",
    "Inputs": [
      {
        "Container": "mp4",
        "FrameRate": "auto",
        "Key": "ETS_example_file.mp4",
        "AspectRatio": "auto",
        "Resolution": "auto",
        "Interlaced": "auto"
      }
    ]
  },
}
```

```
"Playlists": [],
"Outputs": [
  {
    "Status": "Progressing",
    "Rotate": "0",
    "PresetId": "1351620000001-100250",
    "Watermarks": [],
    "Key": "webm/ETS_example_file-kindlefirehd.webm",
    "Id": "1"
  }
],
"PipelineId": "3333333333333-abcde3",
"OutputKeyPrefix": "recipes/",
"UserMetadata": {
  "Cook book": "recipe notebook",
  "Food type": "Italian"
},
"Output": {
  "Status": "Progressing",
  "Rotate": "0",
  "PresetId": "1351620000001-100250",
  "Watermarks": [],
  "Key": "webm/ETS_example_file-kindlefirehd.webm",
  "Id": "1"
},
"Timing": {
  "SubmitTimeMillis": 1533838012298,
  "StartTimeMillis": 1533838013786
},
"Input": {
  "Container": "mp4",
  "FrameRate": "auto",
  "Key": "ETS_example_file.mp4",
  "AspectRatio": "auto",
  "Resolution": "auto",
  "Interlaced": "auto"
},
"Id": "1533838012294-example",
"Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ReadJob](#)」の「」を参照してください。

read-pipeline

次の例は、read-pipeline を使用する方法を説明しています。

AWS CLI

ElasticTranscoder パイプラインを取得するには

この例では、指定された ElasticTranscoder パイプラインを取得します。

コマンド:

```
aws elastictranscoder read-pipeline --id 333333333333-abcde3
```

出力:

```
{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-example",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    },
    "Name": "Default",
    "ThumbnailConfig": {
      "Bucket": "ets-example",
      "StorageClass": "ReducedRedundancy",
      "Permissions": [
        {
          "Access": [
```



```
        "FullControl"
      ],
      "Grantee": "marketing-promos@example.com",
      "GranteeType": "Email"
    }
  ]
},
"Notifications": {
  "Completed": "",
  "Warning": "",
  "Progressing": "",
  "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
},
"Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
"InputBucket": "ets-example",
"Id": "333333333333-abcde3",
"Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-abcde3"
},
"Warnings": [
  {
    "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
    "Code": "6006"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReadPipeline](#)」の「」を参照してください。

read-preset

次の例は、read-preset を使用する方法を説明しています。

AWS CLI

ElasticTranscoder プリセットを取得するには

この例では、指定された ElasticTranscoder プリセットを取得します。

コマンド:

```
aws elastictranscoder read-preset --id 1351620000001-500020
```

出力:

```
{
  "Preset": {
    "Thumbnails": {
      "SizingPolicy": "ShrinkToFit",
      "MaxWidth": "192",
      "Format": "png",
      "PaddingPolicy": "NoPad",
      "Interval": "300",
      "MaxHeight": "108"
    },
    "Container": "fmp4",
    "Description": "System preset: MPEG-Dash Video - 4.8M",
    "Video": {
      "SizingPolicy": "ShrinkToFit",
      "MaxWidth": "1280",
      "PaddingPolicy": "NoPad",
      "FrameRate": "30",
      "MaxHeight": "720",
      "KeyframesMaxDist": "60",
      "FixedGOP": "true",
      "Codec": "H.264",
      "Watermarks": [
        {
          "SizingPolicy": "ShrinkToFit",
          "VerticalOffset": "10%",
          "VerticalAlign": "Top",
          "Target": "Content",
          "MaxWidth": "10%",
          "MaxHeight": "10%",
          "HorizontalAlign": "Left",
          "HorizontalOffset": "10%",
          "Opacity": "100",
          "Id": "TopLeft"
        },
        {
          "SizingPolicy": "ShrinkToFit",
          "VerticalOffset": "10%",
```

```
        "VerticalAlign": "Top",
        "Target": "Content",
        "MaxWidth": "10%",
        "MaxHeight": "10%",
        "HorizontalAlign": "Right",
        "HorizontalOffset": "10%",
        "Opacity": "100",
        "Id": "TopRight"
    },
    {
        "SizingPolicy": "ShrinkToFit",
        "VerticalOffset": "10%",
        "VerticalAlign": "Bottom",
        "Target": "Content",
        "MaxWidth": "10%",
        "MaxHeight": "10%",
        "HorizontalAlign": "Left",
        "HorizontalOffset": "10%",
        "Opacity": "100",
        "Id": "BottomLeft"
    },
    {
        "SizingPolicy": "ShrinkToFit",
        "VerticalOffset": "10%",
        "VerticalAlign": "Bottom",
        "Target": "Content",
        "MaxWidth": "10%",
        "MaxHeight": "10%",
        "HorizontalAlign": "Right",
        "HorizontalOffset": "10%",
        "Opacity": "100",
        "Id": "BottomRight"
    }
],
"CodecOptions": {
    "Profile": "main",
    "MaxBitRate": "4800",
    "InterlacedMode": "Progressive",
    "Level": "3.1",
    "ColorSpaceConversionMode": "None",
    "MaxReferenceFrames": "3",
    "BufferSize": "9600"
},
"BitRate": "4800",
```

```

        "DisplayAspectRatio": "auto"
    },
    "Type": "System",
    "Id": "1351620000001-500020",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/1351620000001-500020",
    "Name": "System preset: MPEG-Dash Video - 4.8M"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ReadPreset](#)」の「」を参照してください。

update-pipeline-notifications

次の例は、update-pipeline-notifications を使用する方法を説明しています。

AWS CLI

ElasticTranscoder パイプラインの通知を更新するには

この例では、指定された ElasticTranscoder パイプラインの通知を更新します。

コマンド:

```

aws elastictranscoder update-pipeline-notifications --id 111111111111-
abcde1 --notifications Progressing=arn:aws:sns:us-west-2:0123456789012:my-
topic,Completed=arn:aws:sns:us-west-2:0123456789012:my-topic,Warning=arn:aws:sns:us-
west-2:0123456789012:my-topic,Error=arn:aws:sns:us-east-1:111222333444:ETS_Errors

```

出力:

```

{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-example",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],

```

```
        "Grantee": "marketing-promos@example.com",
        "GranteeType": "Email"
      }
    ]
  },
  "Name": "Default",
  "ThumbnailConfig": {
    "Bucket": "ets-example",
    "StorageClass": "ReducedRedundancy",
    "Permissions": [
      {
        "Access": [
          "FullControl"
        ],
        "Grantee": "marketing-promos@example.com",
        "GranteeType": "Email"
      }
    ]
  },
  "Notifications": {
    "Completed": "arn:aws:sns:us-west-2:0123456789012:my-topic",
    "Warning": "arn:aws:sns:us-west-2:0123456789012:my-topic",
    "Progressing": "arn:aws:sns:us-west-2:0123456789012:my-topic",
    "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
  },
  "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
  "InputBucket": "ets-example",
  "Id": "111111111111-abcde1",
  "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:pipeline/111111111111-abcde1"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePipelineNotifications](#)」の「」を参照してください。

update-pipeline-status

次のコード例は、update-pipeline-status を使用する方法を示しています。

AWS CLI

ElasticTranscoder パイプラインのステータスを更新するには

この例では、指定された ElasticTranscoder パイプラインのステータスを更新します。

コマンド:

```
aws elastictranscoder update-pipeline-status --id 111111111111-abcde1 --  
status Paused
```

出力:

```
{  
  "Pipeline": {  
    "Status": "Paused",  
    "ContentConfig": {  
      "Bucket": "ets-example",  
      "StorageClass": "Standard",  
      "Permissions": [  
        {  
          "Access": [  
            "FullControl"  
          ],  
          "Grantee": "marketing-promos@example.com",  
          "GranteeType": "Email"  
        }  
      ]  
    },  
    "Name": "Default",  
    "ThumbnailConfig": {  
      "Bucket": "ets-example",  
      "StorageClass": "ReducedRedundancy",  
      "Permissions": [  
        {  
          "Access": [  
            "FullControl"  
          ],  
          "Grantee": "marketing-promos@example.com",  
          "GranteeType": "Email"  
        }  
      ]  
    },  
    "Notifications": {  
      "Completed": "",  
      "Warning": "",  
      "Progressing": "",  
    }  
  }  
}
```

```

    "Error": "arn:aws:sns:us-east-1:803981987763:ETS_Errors"
  },
  "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
  "InputBucket": "ets-example",
  "Id": "111111111111-abcde1",
  "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/111111111111-abcde1"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePipelineStatus](#)」の「」を参照してください。

update-pipeline

次のコード例は、update-pipeline を使用する方法を示しています。

AWS CLI

ElasticTranscoder パイプラインを更新するには

次のupdate-pipeline例では、指定された ElasticTranscoder パイプラインを更新します。

```

aws elastictranscoder update-pipeline \
  --id 111111111111-abcde1 \
  --name DefaultExample \
  --input-bucket salesoffice.example.com-source \
  --role arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role \
  --notifications Progressing="",Completed="",Warning="",Error=arn:aws:sns:us-
east-1:111222333444:ETS_Errors \
  --content-config file://content-config.json \
  --thumbnail-config file://thumbnail-config.json

```

content-config.json の内容:

```

{
  "Bucket": "salesoffice.example.com-public-promos",
  "Permissions": [
    {
      "GranteeType": "Email",
      "Grantee": "marketing-promos@example.com",

```

```
        "Access":[
            "FullControl"
        ]
    },
    ],
    "StorageClass":"Standard"
}
```

thumbnail-config.json の内容:

```
{
  "Bucket":"salesoffice.example.com-public-promos-thumbnails",
  "Permissions":[
    {
      "GranteeType":"Email",
      "Grantee":"marketing-promos@example.com",
      "Access":[
        "FullControl"
      ]
    }
  ],
  "StorageClass":"ReducedRedundancy"
}
```

出力:

```
{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-example",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    }
  },
  "Name": "DefaultExample",
}
```



```
"ThumbnailConfig": {
  "Bucket": "ets-example",
  "StorageClass": "ReducedRedundancy",
  "Permissions": [
    {
      "Access": [
        "FullControl"
      ],
      "Grantee": "marketing-promos@example.com",
      "GranteeType": "Email"
    }
  ]
},
"Notifications": {
  "Completed": "",
  "Warning": "",
  "Progressing": "",
  "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
},
"Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
"InputBucket": "ets-example",
"Id": "333333333333-abcde3",
"Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-abcde3"
},
"Warnings": [
  {
    "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
    "Code": "6006"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePipeline](#)」の「」を参照してください。

ElastiCache を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています ElastiCache。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags-to-resource

次のコード例は、add-tags-to-resource を使用する方法を示しています。

AWS CLI

リソースにタグを追加するには

次のadd-tags-to-resource例では、クラスターまたはスナップショットリソースに最大 10 個のタグ、キーと値のペアを追加します。

```
aws elasticache add-tags-to-resource \  
  --resource-name "arn:aws:elasticache:us-east-1:1234567890:cluster:my-mem-  
cluster" \  
  --tags '{"20150202":15, "ElastiCache":"Service"}'
```

出力:

```
{  
  "TagList": [  
    {  
      "Value": "20150202",  
      "Key": "APIVersion"    }  
  ]  
}
```

```
    },  
    {  
      "Value": "ElastiCache",  
      "Key": "Service"  
    }  
  ]  
}
```

詳細については、ElastiCache ユーザーガイドの [「コスト配分タグによるコストのモニタリング」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AddTagsToResource](#)」の「」を参照してください。

authorize-cache-security-group-ingress

次の例は、authorize-cache-security-group-ingress を使用する方法を説明しています。

AWS CLI

キャッシュセキュリティグループが進入することを許可するには

次のauthorize-cache-security-group-ingress例では、キャッシュセキュリティグループへのネットワーク進入を許可します。

```
aws elasticache authorize-cache-security-group-ingress \  
  --cache-security-group-name "my-sec-grp" \  
  --ec2-security-group-name "my-ec2-sec-grp" \  
  --ec2-security-group-owner-id "1234567890"
```

コマンドは出力を生成しません。

詳細については、ElastiCache ユーザーガイドの [「Amazon でのセルフサービスの更新 ElastiCache」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AuthorizeCacheSecurityGroupIngress](#)」の「」を参照してください。

batch-apply-update-action

次のコード例は、batch-apply-update-action を使用する方法を示しています。

AWS CLI

サービスの更新を適用するには

次のbatch-apply-update-action例では、Redis クラスターにサービス更新を適用します。

```
aws elasticache batch-apply-update-action \  
  --service-update-name elc-xxxxx406-xxx \  
  --replication-group-ids test-cluster
```

出力:

```
{  
  "ProcessedUpdateActions": [  
    {  
      "ReplicationGroupId": "pat-cluster",  
      "ServiceUpdateName": "elc-xxxxx406-xxx",  
      "UpdateActionStatus": "waiting-to-start"  
    }  
  ],  
  "UnprocessedUpdateActions": []  
}
```

詳細については、Elasticache ユーザーガイドの「[Amazon でのセルフサービスの更新 ElastiCache](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchApplyUpdateAction](#)」の「」を参照してください。

batch-stop-update-action

次の例は、batch-stop-update-action を使用する方法を説明しています。

AWS CLI

サービスの更新を停止するには

次のbatch-stop-update-action例では、Redis クラスターにサービス更新を適用します。

```
aws elasticache batch-stop-update-action \  
  --service-update-name elc-xxxxx406-xxx \  
  --replication-group-ids test-cluster
```

出力:

```
{
  "ProcessedUpdateActions": [
    {
      "ReplicationGroupId": "pat-cluster",
      "ServiceUpdateName": "elc-xxxxx406-xxx",
      "UpdateActionStatus": "stopping"
    }
  ],
  "UnprocessedUpdateActions": []
}
```

詳細については、Elasticache ユーザーガイドの [「Amazon でのセルフサービスの更新 ElastiCache」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchStopUpdateAction](#)」の「」を参照してください。

copy-snapshot

次のコード例は、copy-snapshot を使用する方法を示しています。

AWS CLI

スナップショットをコピーするには

次のcopy-snapshot例では、既存のスナップショットのコピーを作成します。

```
aws elasticache copy-snapshot \  
  --source-snapshot-name "my-snapshot" \  
  --target-snapshot-name "my-snapshot-copy"
```

出力:

```
{
  "Snapshot": {
    "Engine": "redis",
    "CacheParameterGroupName": "default.redis3.2",
    "VpcId": "vpc-3820329f3",
    "CacheClusterId": "my-redis4",
  }
}
```

```
"SnapshotRetentionLimit": 7,
"NumCacheNodes": 1,
"SnapshotName": "my-snapshot-copy",
"CacheClusterCreateTime": "2016-12-21T22:24:04.955Z",
"AutoMinorVersionUpgrade": true,
"PreferredAvailabilityZone": "us-east-1c",
"SnapshotStatus": "creating",
"SnapshotSource": "manual",
"SnapshotWindow": "07:00-08:00",
"EngineVersion": "3.2.4",
"NodeSnapshots": [
  {
    "CacheSize": "3 MB",
    "SnapshotCreateTime": "2016-12-28T07:00:52Z",
    "CacheNodeId": "0001",
    "CacheNodeCreateTime": "2016-12-21T22:24:04.955Z"
  }
],
"CacheSubnetGroupName": "default",
"Port": 6379,
"PreferredMaintenanceWindow": "tue:09:30-tue:10:30",
"CacheNodeType": "cache.m3.large"
}
```

詳細については、ElastiCache ユーザーガイドの「[バックアップのエクスポート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopySnapshot](#)」の「」を参照してください。

create-cache-cluster

次のコード例は、create-cache-cluster を使用する方法を示しています。

AWS CLI

キャッシュクラスターを作成するには

次のcreate-cache-cluster例では、Redis エンジンを使用してキャッシュクラスターを作成します。

```
aws elasticache create-cache-cluster \
```

```
--cache-cluster-id "cluster-test" \  
--engine redis \  
--cache-node-type cache.m5.large \  
--num-cache-nodes 1
```

出力:

```
{  
  "CacheCluster": {  
    "CacheClusterId": "cluster-test",  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",  
    "CacheNodeType": "cache.m5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "CacheClusterStatus": "creating",  
    "NumCacheNodes": 1,  
    "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",  
    "PendingModifiedValues": {},  
    "CacheSecurityGroups": [],  
    "CacheParameterGroup": {  
      "CacheParameterGroupName": "default.redis5.0",  
      "ParameterApplyStatus": "in-sync",  
      "CacheNodeIdsToReboot": []  
    },  
    "CacheSubnetGroupName": "default",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:30-07:30",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

詳細については、Elasticache ユーザーガイドの「[クラスターの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCacheCluster](#)」の「」を参照してください。

create-cache-parameter-group

次のコード例は、create-cache-parameter-group を使用する方法を示しています。

AWS CLI

キャッシュパラメータグループを作成するには

次のcreate-cache-parameter-group例では、新しい Amazon ElastiCache キャッシュパラメータグループを作成します。

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-family "redis5.0" \  
  --cache-parameter-group-name "mygroup" \  
  --description "mygroup"
```

出力:

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "mygroup",  
    "CacheParameterGroupFamily": "redis5.0",  
    "Description": "my group"  
  }  
}
```

詳細については、Elasticache ユーザーガイドの「[パラメータグループの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCacheParameterGroup](#)」の「」を参照してください。

create-cache-subnet-group

次のコード例は、create-cache-subnet-group を使用する方法を示しています。

AWS CLI

キャッシュサブネットグループを作成するには

次のcreate-cache-subnet-group例では、新しいキャッシュサブネットグループを作成します。

```
aws elasticache create-cache-subnet-group \  
  --cache-subnet-group-name "mygroup" \  
  --cache-subnet-group-description "my subnet group" \  
  --vpc-subnet-id "vpc-subnet-id"
```



```
--subnet-ids "subnet-xxxxec4f"
```

出力:

```
{
  "CacheSubnetGroup": {
    "CacheSubnetGroupName": "mygroup",
    "CacheSubnetGroupDescription": "my subnet group",
    "VpcId": "vpc-a3e97cdb",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-xxxxec4f",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        }
      }
    ]
  }
}
```

詳細については、Elasticache ユーザーガイドの [「キャッシュサブネットグループの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateCacheSubnetGroup](#)」の「」を参照してください。

create-global-replication-group

次のコード例は、create-global-replication-group を使用する方法を示しています。

AWS CLI

グローバルレプリケーショングループを作成するには

次のcreate-global-replication-group例では、新しいグローバルレプリケーショングループを作成します。

```
aws elasticache create-global-replication-group \
  --global-replication-group-id-suffix my-global-replication-group \
  --primary-replication-group-id my-primary-cluster
```

出力:

```
{
  "GlobalReplicationGroup": {
    "GlobalReplicationGroupId": "sgau-my-global-replication-group",
    "GlobalReplicationGroupDescription": " ",
    "Status": "creating",
    "CacheNodeType": "cache.r5.large",
    "Engine": "redis",
    "EngineVersion": "5.0.6",
    "Members": [
      {
        "ReplicationGroupId": "my-primary-cluster",
        "ReplicationGroupRegion": "us-west-2",
        "Role": "PRIMARY",
        "AutomaticFailover": "enabled",
        "Status": "associating"
      }
    ],
    "ClusterEnabled": true,
    "GlobalNodeGroups": [
      {
        "GlobalNodeGroupId": "sgau-my-global-replication-group-0001",
        "Slots": "0-16383"
      }
    ],
    "AuthTokenEnabled": false,
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

詳細については、Elasticache ユーザーガイドの [「グローバルデータストアを使用した AWS リージョン間のレプリケーション」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateGlobalReplicationGroup](#)」の「」を参照してください。

create-replication-group

次の例は、create-replication-group を使用する方法を説明しています。

AWS CLI

レプリケーショングループを作成するには

次のcreate-replication-group例では、Redis (クラスターモードが無効) または Redis (クラスターモードが有効) レプリケーショングループを作成します。このオペレーションは Redis でのみ有効です。

```
aws elasticache create-replication-group \  
  --replication-group-id "mygroup" \  
  --replication-group-description "my group" \  
  --engine "redis" \  
  --cache-node-type "cache.m5.large"
```

出力:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "mygroup",  
    "Description": "my group",  
    "Status": "creating",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "mygroup-001"  
    ],  
    "AutomaticFailover": "disabled",  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:00-07:00",  
    "ClusterEnabled": false,  
    "CacheNodeType": "cache.m5.large",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

詳細については、Elasticache ユーザーガイドの「[Redis レプリケーショングループの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateReplicationGroup](#)」の「」を参照してください。

create-snapshot

次のコード例は、create-snapshot を使用する方法を示しています。

AWS CLI

スナップショットを作成するには

次のcreate-snapshot例では、Redis エンジンを使用してスナップショットを作成します。

```
aws elasticache create-snapshot \  
  --snapshot-name mysnapshot \  
  --cache-cluster-id cluster-test
```

出力:

```
{  
  "Snapshot": {  
    "SnapshotName": "mysnapshot",  
    "CacheClusterId": "cluster-test",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "CacheNodeType": "cache.m5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "NumCacheNodes": 1,  
    "PreferredAvailabilityZone": "us-west-2b",  
    "CacheClusterCreateTime": "2020-03-19T03:12:01.483Z",  
    "PreferredMaintenanceWindow": "sat:13:00-sat:14:00",  
    "Port": 6379,  
    "CacheParameterGroupName": "default.redis5.0",  
    "CacheSubnetGroupName": "default",  
    "VpcId": "vpc-a3e97cdb",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:30-07:30",  
    "NodeSnapshots": [  
      {  
        "CacheNodeId": "0001",  
        "CacheSize": "",  
        "CacheNodeCreateTime": "2020-03-19T03:12:01.483Z"  
      }  
    ]  
  }  
}
```

詳細については、Elasticache ユーザーガイドの「[Redis ElastiCache のバックアップと復元](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSnapshot](#)」の「」を参照してください。

create-user-group

次のコード例は、create-user-group を使用する方法を示しています。

AWS CLI

ユーザーグループを作成するには

次のcreate-user-group例では、新しいユーザーグループを作成します。

```
aws elasticache create-user-group \  
  --user-group-id myusergroup \  
  --engine redis \  
  --user-ids default
```

出力:

```
{  
  "UserGroupId": "myusergroup",  
  "Status": "creating",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"  
}
```

詳細については、Elasticache ユーザーガイドの「[ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUserGroup](#)」の「」を参照してください。

create-user

次の例は、create-user を使用する方法を説明しています。

AWS CLI

ユーザーを作成するには

次のcreate-user例では、新しいユーザーを作成します。

```
aws elasticache create-user \  
  --user-id user1 \  
  --user-name myUser \  
  --passwords mYnuUzrpAxXw2rdzx \  
  --engine redis \  
  --access-string "on ~app:* -@all +@read"
```

出力:

```
{  
  "UserId": "user2",  
  "UserName": "myUser",  
  "Status": "active",  
  "Engine": "redis",  
  "AccessString": "on ~app:* -@all +@read +@hash +@bitmap +@geo -setbit -bitfield  
-hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius -  
georadiusbymember",  
  "UserGroupIds": [],  
  "Authentication": {  
    "Type": "password",  
    "PasswordCount": 1  
  },  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx52:user:user2"  
}
```

詳細については、Elasticache ユーザーガイドの「[ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUser](#)」の「」を参照してください。

decrease-node-groups-in-global-replication-group

次の例は、`decrease-node-groups-in-global-replication-group` を使用する方法を説明しています。

AWS CLI

グローバルレプリケーショングループ内のノードグループの数を減らすには

以下は、Redis エンジンを使用してノードグループの数 `decrease-node-groups-in-global-replication-group` を減らします。

```
aws elasticache decrease-node-groups-in-global-replication-group \  
  --global-replication-group-id sgaui-test \  
  --node-group-count 1 \  
  --apply-immediately \  
  --global-node-groups-to-retain sgaui-test-0003
```

出力:

```
{  
  "GlobalReplicationGroup":  
  {  
    "GlobalReplicationGroupId": "sgaui-test",  
    "GlobalReplicationGroupDescription": "test",  
    "Status": "modifying",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "Members": [  
      {  
        "ReplicationGroupId": "test-2",  
        "ReplicationGroupRegion": "us-east-1",  
        "Role": "SECONDARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      },  
      {  
        "ReplicationGroupId": "test-1",  
        "ReplicationGroupRegion": "us-west-2",  
        "Role": "PRIMARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      }  
    ]  
  }  
}
```

```
    }
  ],
  "ClusterEnabled": true,
  "GlobalNodeGroups": [
    {
      "GlobalNodeGroupId": "sgaui-test-0001",
      "Slots": "0-449,1816-5461"
    },
    {
      "GlobalNodeGroupId": "sgaui-test-0002",
      "Slots": "6827-10922"
    },
    {
      "GlobalNodeGroupId": "sgaui-test-0003",
      "Slots": "10923-14052,15418-16383"
    },
    {
      "GlobalNodeGroupId": "sgaui-test-0004",
      "Slots": "450-1815,5462-6826,14053-15417"
    }
  ],
  "AuthTokenEnabled": false,
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false
}
}
```

詳細については、ElastiCache ユーザーガイドの「[グローバルデータストアを使用した AWS リージョン間のレプリケーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DecreaseNodeGroupsInGlobalReplicationGroup](#)」の「」を参照してください。

decrease-replica-count

次の例は、decrease-replica-count を使用する方法を説明しています。

AWS CLI

レプリカ数を減らすには

次のdecrease-replica-count例では、Redis (クラスターモードが無効) レプリケーショングループのレプリカの数、または Redis (クラスターモードが有効) レプリケーショングループの 1

つ以上のノードグループ (シャード) 内のレプリカノードの数を動的に減らします。このオペレーションは、クラスターのダウンタイムなしで実行されます。

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-cluster \  
  --apply-immediately \  
  --new-replica-count 2
```

出力:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "my-cluster",  
    "Description": " ",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "myrepliac",  
      "my-cluster-001",  
      "my-cluster-002",  
      "my-cluster-003"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "my-cluster.xxxxx.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "ReaderEndpoint": {  
          "Address": "my-cluster-  
ro.xxxxx.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "NodeGroupMembers": [  
          {  
            "CacheClusterId": "myrepliac",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Address":  
"myrepliac.xxxxx.0001.usw2.cache.amazonaws.com",  
              "Port": 6379  
            }  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
    },
    "PreferredAvailabilityZone": "us-west-2a",
    "CurrentRole": "replica"
  },
  {
    "CacheClusterId": "my-cluster-001",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Address": "my-
cluster-001.xxxxx.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2a",
    "CurrentRole": "primary"
  },
  {
    "CacheClusterId": "my-cluster-002",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Address": "my-
cluster-002.xxxxx.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2a",
    "CurrentRole": "replica"
  },
  {
    "CacheClusterId": "my-cluster-003",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Address": "my-
cluster-003.xxxxx.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2a",
    "CurrentRole": "replica"
  }
]
}
],
"AutomaticFailover": "disabled",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
```

```
    "CacheNodeType": "cache.r5.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

詳細については、Elasticache ユーザーガイドの「[レプリカ数の変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DecreaseReplicaCount](#)」の「」を参照してください。

delete-cache-cluster

次の例は、delete-cache-cluster を使用する方法を説明しています。

AWS CLI

キャッシュクラスターを削除するには

次のdelete-cache-cluster例では、以前にプロビジョニングされた指定されたクラスターを削除します。コマンドは、関連するすべてのキャッシュノード、ノードエンドポイント、およびクラスター自体を削除します。このオペレーションから正常なレスポンスを受信すると、Amazon はクラスターの削除を ElastiCache すぐに開始します。このオペレーションをキャンセルまたは元に戻すことはできません。

このオペレーションは、次の場合には無効です。

Redis (クラスターモードが有効) clustersAレプリケーションgroupA最後のリードレプリカであるクラスターマルチ AZ モードenabledAであるノードグループ (シャード)Redis (クラスターモードが有効) レプリケーションgroupAからのクラスター利用可能な状態ではないクラスター

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id "my-cluster-002"
```

出力:

```
{  
  "CacheCluster": {  
    "CacheClusterId": "my-cluster-002",  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",
```

```
"CacheNodeType": "cache.r5.xlarge",
"Engine": "redis",
"EngineVersion": "5.0.5",
"CacheClusterStatus": "deleting",
"NumCacheNodes": 1,
"PreferredAvailabilityZone": "us-west-2a",
"CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",
"PreferredMaintenanceWindow": "mon:04:05-mon:05:05",
"PendingModifiedValues": {},
"NotificationConfiguration": {
  "TopicArn": "arn:aws:sns:us-west-x:xxxxxxx4152:My_Topic",
  "TopicStatus": "active"
},
"CacheSecurityGroups": [],
"CacheParameterGroup": {
  "CacheParameterGroupName": "mygroup",
  "ParameterApplyStatus": "in-sync",
  "CacheNodeIdsToReboot": []
},
"CacheSubnetGroupName": "kxkxk",
"AutoMinorVersionUpgrade": true,
"SecurityGroups": [
  {
    "SecurityGroupId": "sg-xxxxxxxxxx9836",
    "Status": "active"
  },
  {
    "SecurityGroupId": "sg-xxxxxxxxxx7b",
    "Status": "active"
  }
],
"ReplicationGroupId": "my-cluster",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}
```

詳細については、ElastiCache ユーザーガイドの「[クラスタの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCacheCluster](#)」の「」を参照してください。

delete-cache-parameter-group

次のコード例は、delete-cache-parameter-group を使用する方法を示しています。

AWS CLI

キャッシュパラメータグループを削除するには

次のdelete-cache-parameter-group例では、指定されたキャッシュパラメータグループを削除します。キャッシュクラスターに関連付けられているキャッシュパラメータグループは削除できません。

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myparamgroup
```

このコマンドでは何も出力されません。

詳細については、Elasticache ユーザーガイドの [「パラメータグループの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteCacheParameterGroup](#)」の「」を参照してください。

delete-cache-subnet-group

次のコード例は、delete-cache-subnet-group を使用する方法を示しています。

AWS CLI

キャッシュサブネットグループを削除するには

次のdelete-cache-subnet-group例では、指定されたキャッシュサブネットグループを削除します。キャッシュサブネットグループがクラスターに関連付けられている場合は、削除することはできません。

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name "mygroup"
```

このコマンドでは何も出力されません。

詳細については、Elasticache ユーザーガイドの [「サブネットグループの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteCacheSubnetGroup](#)」の「」を参照してください。

delete-global-replication-group

次の例は、delete-global-replication-group を使用する方法を説明しています。

AWS CLI

グローバルレプリケーショングループを削除するには

次のdelete-global-replication-group例では、新しいグローバルレプリケーショングループを削除します。

```
aws elasticache delete-global-replication-group \  
  --global-replication-group-id my-global-replication-group \  
  --retain-primary-replication-group
```

出力:

```
{  
  "GlobalReplicationGroup": {  
    "GlobalReplicationGroupId": "sgaui-my-grg",  
    "GlobalReplicationGroupDescription": "my-grg",  
    "Status": "deleting",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "Members": [  
      {  
        "ReplicationGroupId": "my-cluster-grg",  
        "ReplicationGroupRegion": "us-west-2",  
        "Role": "PRIMARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      }  
    ],  
    "ClusterEnabled": false,  
    "AuthTokenEnabled": false,  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

```
}
```

詳細については、Elasticache ユーザーガイドの「[グローバルデータストアを使用した AWS リージョン間のレプリケーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGlobalReplicationGroup](#)」の「」を参照してください。

delete-replication-group

次のコード例は、delete-replication-group を使用する方法を示しています。

AWS CLI

レプリケーショングループを削除するには

次のdelete-replication-group例では、既存のレプリケーショングループを削除します。デフォルトでは、このオペレーションはプライマリ/プライマリとすべてのリードレプリカを含むレプリケーショングループ全体を削除します。レプリケーショングループにプライマリが 1 つしかない場合は、オプションでリードレプリカのみを削除し、プライマリは RetainPrimaryCluster=true を設定して保持できます。

このオペレーションから正常なレスポンスを受信すると、Amazon は選択したリソースの削除をElastiCache すぐに開始します。このオペレーションをキャンセルまたは元に戻すことはできません。Redis にのみ有効です。

```
aws elasticache delete-replication-group \  
  --replication-group-id "mygroup"
```

出力:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "mygroup",  
    "Description": "my group",  
    "Status": "deleting",  
    "PendingModifiedValues": {},  
    "AutomaticFailover": "disabled",  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:00-07:00",
```

```
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteReplicationGroup](#)」の「」を参照してください。

delete-snapshot

次のコード例は、delete-snapshot を使用する方法を示しています。

AWS CLI

スナップショットを削除するには

次のdelete-snapshot例では、Redis エンジンを使用してスナップショットを削除しました。

```
aws elasticache delete-snapshot \  
  --snapshot-name mysnapshot
```

出力:

```
{  
  "Snapshot": {  
    "SnapshotName": "my-cluster-snapshot",  
    "ReplicationGroupId": "mycluster",  
    "ReplicationGroupDescription": "mycluster",  
    "SnapshotStatus": "deleting",  
    "SnapshotSource": "manual",  
    "CacheNodeType": "cache.r5.xlarge",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "PreferredMaintenanceWindow": "thu:12:00-thu:13:00",  
    "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxxxx152:My_Topic",  
    "Port": 6379,  
    "CacheParameterGroupName": "default.redis5.0.cluster.on",  
    "CacheSubnetGroupName": "default",  
    "VpcId": "vpc-a3e97cdb",  
    "AutoMinorVersionUpgrade": true,  
    "SnapshotRetentionLimit": 1,  
    "SnapshotWindow": "13:00-14:00",
```



```
"NumNodeGroups": 4,
"AutomaticFailover": "enabled",
"NodeSnapshots": [
  {
    "CacheClusterId": "mycluster-0002-003",
    "NodeGroupId": "0002",
    "CacheNodeId": "0001",
    "CacheSize": "6 MB",
    "CacheNodeCreateTime": "2020-06-18T00:05:44.719000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  },
  {
    "CacheClusterId": "mycluster-0003-003",
    "NodeGroupId": "0003",
    "CacheNodeId": "0001",
    "CacheSize": "6 MB",
    "CacheNodeCreateTime": "2019-12-05T19:13:15.912000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  },
  {
    "CacheClusterId": "mycluster-0004-002",
    "NodeGroupId": "0004",
    "CacheNodeId": "0001",
    "CacheSize": "6 MB",
    "CacheNodeCreateTime": "2019-12-09T19:44:34.324000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  },
  {
    "CacheClusterId": "mycluster-0005-003",
    "NodeGroupId": "0005",
    "CacheNodeId": "0001",
    "CacheSize": "6 MB",
    "CacheNodeCreateTime": "2020-06-18T00:05:44.775000+00:00",
    "SnapshotCreateTime": "2020-06-25T20:34:30+00:00"
  }
]
}
```

詳細については、Elasticache ユーザーガイドの「[Redis ElastiCache のバックアップと復元](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSnapshot](#)」の「」を参照してください。

delete-user-group

次のコード例は、delete-user-group を使用する方法を示しています。

AWS CLI

ユーザーグループを削除するには

次のdelete-user-group例では、ユーザーグループを削除します。

```
aws elasticache delete-user-group \  
  --user-group-id myusergroup
```

出力:

```
{  
  "UserId": "myusergroup",  
  "Status": "deleting",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"  
}
```

詳細については、Elasticache ユーザーガイドの [「ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteUserGroup](#)」の「」を参照してください。

delete-user

次のコード例は、delete-user を使用する方法を示しています。

AWS CLI

ユーザーを削除するには

次のdelete-user例では、ユーザーを削除します。

```
aws elasticache delete-user \  
  --user-id myuserid
```

```
--user-id user2
```

出力:

```
{
  "UserId": "user1",
  "UserName": "myUser",
  "Status": "deleting",
  "Engine": "redis",
  "AccessString": "on ~* +@all",
  "UserGroupIds": [
    "myusergroup"
  ],
  "Authentication": {
    "Type": "password",
    "PasswordCount": 1
  },
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx52:user:user1"
}
```

詳細については、Elasticache ユーザーガイドの「[ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteUser](#)」の「」を参照してください。

describe-cache-clusters

次の例は、describe-cache-clusters を使用する方法を説明しています。

AWS CLI

キャッシュクラスターを記述するには

次のdescribe-cache-clusters例では、キャッシュクラスターについて説明します。

```
aws elasticache describe-cache-clusters
```

出力:

```
{
  "CacheClusters": [
```

```
{
  "CacheClusterId": "my-cluster-003",
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/
elasticache/home#client-download:",
  "CacheNodeType": "cache.r5.large",
  "Engine": "redis",
  "EngineVersion": "5.0.5",
  "CacheClusterStatus": "available",
  "NumCacheNodes": 1,
  "PreferredAvailabilityZone": "us-west-2a",
  "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
  "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
  "PendingModifiedValues": {},
  "NotificationConfiguration": {
    "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxx152:My_Topic",
    "TopicStatus": "active"
  },
  "CacheSecurityGroups": [],
  "CacheParameterGroup": {
    "CacheParameterGroupName": "default.redis5.0",
    "ParameterApplyStatus": "in-sync",
    "CacheNodeIdsToReboot": []
  },
  "CacheSubnetGroupName": "kxkxk",
  "AutoMinorVersionUpgrade": true,
  "SecurityGroups": [
    {
      "SecurityGroupId": "sg-xxxxxd7b",
      "Status": "active"
    }
  ],
  "ReplicationGroupId": "my-cluster",
  "SnapshotRetentionLimit": 0,
  "SnapshotWindow": "06:30-07:30",
  "AuthTokenEnabled": false,
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false,
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx152:cluster:my-cache-
cluster",
  "ReplicationGroupLogDeliveryEnabled": false,
  "LogDeliveryConfigurations": [
    {
      "LogType": "slow-log",
      "DestinationType": "cloudwatch-logs",

```

```

        "DestinationDetails": {
            "CloudWatchLogsDetails": {
                "LogGroup": "test-log"
            }
        },
        "LogFormat": "text",
        "Status": "active"
    }
]
}

```

詳細については、Elasticache ユーザーガイドの「[クラスターの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCacheClusters](#)」の「」を参照してください。

describe-cache-engine-versions

次の例は、describe-cache-engine-versions を使用する方法を説明しています。

AWS CLI

キャッシュエンジンのバージョンを記述するには

次のdescribe-cache-engine-versions例では、使用可能なキャッシュエンジンとそのバージョンのリストを返します。

```
aws elasticache describe-cache-engine-versions \
  --engine Redis
```

出力:

```

{
  "CacheEngineVersions": [
    {
      "Engine": "redis",
      "EngineVersion": "2.6.13",
      "CacheParameterGroupFamily": "redis2.6",
      "CacheEngineDescription": "Redis",
      "CacheEngineVersionDescription": "redis version 2.6.13"
    },
  ],
}

```

```
{
  "Engine": "redis",
  "EngineVersion": "2.8.19",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.19"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.21",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.21"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.22",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.22"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.23",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.23"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.24",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.24"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.6",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.6"
},
{
  "Engine": "redis",
```

```
    "EngineVersion": "3.2.10",
    "CacheParameterGroupFamily": "redis3.2",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 3.2.10"
  },
  {
    "Engine": "redis",
    "EngineVersion": "3.2.4",
    "CacheParameterGroupFamily": "redis3.2",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 3.2.4"
  },
  {
    "Engine": "redis",
    "EngineVersion": "3.2.6",
    "CacheParameterGroupFamily": "redis3.2",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 3.2.6"
  },
  {
    "Engine": "redis",
    "EngineVersion": "4.0.10",
    "CacheParameterGroupFamily": "redis4.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 4.0.10"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.0",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.0"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.3",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.3"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.4",
    "CacheParameterGroupFamily": "redis5.0",
```

```
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.4"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.5",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.5"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCacheEngineVersions](#)」の「」を参照してください。

describe-cache-parameter-groups

次のコード例は、describe-cache-parameter-groups を使用する方法を示しています。

AWS CLI

キャッシュパラメータグループを記述するには

次のdescribe-cache-parameter-groups例では、キャッシュパラメータグループの説明のリストを返します。

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name "mygroup"
```

出力:

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "mygroup",
      "CacheParameterGroupFamily": "redis5.0",
      "Description": " "
    }
  ]
}
```


詳細については、ElastiCache ユーザーガイドの「[パラメータグループを使用したエンジンパラメータの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCacheParameterGroups](#)」の「」を参照してください。

describe-cache-parameters

次のコード例は、describe-cache-parameters を使用する方法を示しています。

AWS CLI

キャッシュパラメータを記述するには

次のdescribe-cache-parameters 「」の例では、指定されたキャッシュパラメータグループの詳細なパラメータリストを返します。

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name "myparamgroup"
```

出力:

```
{  
  "Parameters": [  
    {  
      "ParameterName": "activedefrag",  
      "ParameterValue": "yes",  
      "Description": "Enabled active memory defragmentation",  
      "Source": "user",  
      "DataType": "string",  
      "AllowedValues": "yes,no",  
      "IsModifiable": true,  
      "MinimumEngineVersion": "5.0.0",  
      "ChangeType": "immediate"  
    },  
    {  
      "ParameterName": "active-defrag-cycle-max",  
      "ParameterValue": "75",  
      "Description": "Maximal effort for defrag in CPU percentage",  
      "Source": "user",  
      "DataType": "integer",  
      "AllowedValues": "1-75",  
      "IsModifiable": true,  
    }  
  ]  
}
```

```
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-cycle-min",
    "ParameterValue": "5",
    "Description": "Minimal effort for defrag in CPU percentage",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-75",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-ignore-bytes",
    "ParameterValue": "104857600",
    "Description": "Minimum amount of fragmentation waste to start active
defrag",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1048576-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-max-scan-fields",
    "ParameterValue": "1000",
    "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-1000000",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-threshold-lower",
    "ParameterValue": "10",
    "Description": "Minimum percentage of fragmentation to start active
defrag",
    "Source": "user",
```

```
    "DataType": "integer",
    "AllowedValues": "1-100",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "active-defrag-threshold-upper",
    "ParameterValue": "100",
    "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-100",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "activeresharding",
    "ParameterValue": "yes",
    "Description": "Apply rehashing or not.",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "appendfsync",
    "ParameterValue": "everysec",
    "Description": "fsync policy for AOF persistence",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "always,everysec,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "appendonly",
    "ParameterValue": "no",
    "Description": "Enable Redis persistence.",
```

```
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-hard-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer hard limit in bytes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in bytes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in seconds.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
    "ParameterValue": "33554432",
    "Description": "Pubsub client output buffer hard limit in bytes.",
```

```
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
    "ParameterValue": "8388608",
    "Description": "Pubsub client output buffer soft limit in bytes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
    "ParameterValue": "60",
    "Description": "Pubsub client output buffer soft limit in seconds.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
    "ParameterValue": "60",
    "Description": "Replica client output buffer soft limit in seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-query-buffer-limit",
    "ParameterValue": "1073741824",
    "Description": "Max size of a single client query buffer",
```

```
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1048576-1073741824",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "close-on-replica-write",
    "ParameterValue": "yes",
    "Description": "If enabled, clients who attempt to write to a read-only
replica will be disconnected. Applicable to 2.8.23 and higher.",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "cluster-enabled",
    "ParameterValue": "no",
    "Description": "Enable cluster mode",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "cluster-require-full-coverage",
    "ParameterValue": "no",
    "Description": "Whether cluster becomes unavailable if one or more slots
are not covered",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "databases",
```

```
    "ParameterValue": "16",
    "Description": "Set the number of databases.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-1200000",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "hash-max-ziplist-entries",
    "ParameterValue": "512",
    "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "hash-max-ziplist-value",
    "ParameterValue": "64",
    "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "hll-sparse-max-bytes",
    "ParameterValue": "3000",
    "Description": "HyperLogLog sparse representation bytes limit",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-16000",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
}
```

```
{
  "ParameterName": "lazyfree-lazy-eviction",
  "ParameterValue": "no",
  "Description": "Perform an asynchronous delete on evictions",
  "Source": "user",
  "DataType": "string",
  "AllowedValues": "yes,no",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "lazyfree-lazy-expire",
  "ParameterValue": "no",
  "Description": "Perform an asynchronous delete on expired keys",
  "Source": "user",
  "DataType": "string",
  "AllowedValues": "yes,no",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "lazyfree-lazy-server-del",
  "ParameterValue": "no",
  "Description": "Perform an asynchronous delete on key updates",
  "Source": "user",
  "DataType": "string",
  "AllowedValues": "yes,no",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "lfu-decay-time",
  "ParameterValue": "1",
  "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
  "Source": "user",
  "DataType": "integer",
  "AllowedValues": "0-",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
}
```



```
    },
    {
      "ParameterName": "lfu-log-factor",
      "ParameterValue": "10",
      "Description": "The log factor for incrementing key counter for LFU
eviction policy",
      "Source": "user",
      "DataType": "integer",
      "AllowedValues": "1-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "list-compress-depth",
      "ParameterValue": "0",
      "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
      "Source": "user",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "list-max-ziplist-size",
      "ParameterValue": "-2",
      "Description": "The number of entries allowed per internal list node can
be specified as a fixed maximum size or a maximum number of elements",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "-5,-4,-3,-2,-1,1-",
      "IsModifiable": false,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "lua-replicate-commands",
      "ParameterValue": "yes",
      "Description": "Always enable Lua effect replication or not",
      "Source": "user",
      "DataType": "string",
```

```
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lua-time-limit",
    "ParameterValue": "5000",
    "Description": "Max execution time of a Lua script in milliseconds. 0
for unlimited execution without warnings.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "5000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "maxclients",
    "ParameterValue": "65000",
    "Description": "The maximum number of Redis clients.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-65000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "maxmemory-policy",
    "ParameterValue": "volatile-lru",
    "Description": "Max memory policy.",
    "Source": "user",
    "DataType": "string",
    "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "maxmemory-samples",
    "ParameterValue": "3",
    "Description": "Max memory samples.",
```

```
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "min-replicas-max-lag",
    "ParameterValue": "10",
    "Description": "The maximum amount of replica lag in seconds beyond
which the master would stop taking writes. A value of 0 means the master always
takes writes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "min-replicas-to-write",
    "ParameterValue": "0",
    "Description": "The minimum number of replicas that must be present with
lag no greater than min-replicas-max-lag for master to take writes. Setting this to
0 means the master always takes writes.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "notify-keyspace-events",
    "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
    "Source": "user",
    "DataType": "string",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
```

```

    "ParameterName": "proto-max-bulk-len",
    "ParameterValue": "536870912",
    "Description": "Max size of a single element request",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "1048576-536870912",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "rename-commands",
    "ParameterValue": "",
    "Description": "Redis commands that can be dynamically renamed by the
customer",
    "Source": "user",
    "DataType": "string",
    "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.3",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "repl-backlog-size",
    "ParameterValue": "1048576",
    "Description": "The replication backlog size in bytes for PSYNC. This is
the size of the buffer which accumulates slave data when slave is disconnected for
some time, so that when slave reconnects again, only transfer the portion of data
which the slave missed. Minimum value is 16K.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "16384-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "repl-backlog-ttl",
    "ParameterValue": "3600",
    "Description": "The amount of time in seconds after the master no longer
have any slaves connected for the master to free the replication backlog. A value
of 0 means to never release the backlog.",
    "Source": "user",

```

```
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-allow-chaining",
    "ParameterValue": "no",
    "Description": "Configures if chaining of replicas is allowed",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-ignore-maxmemory",
    "ParameterValue": "yes",
    "Description": "Determines if replica ignores maxmemory setting by not
evicting items independent from the master",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-lazy-flush",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous flushDB during replica sync",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "reserved-memory-percent",
    "ParameterValue": "25",
```

```
    "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-100",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "set-max-intset-entries",
    "ParameterValue": "512",
    "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "slowlog-log-slower-than",
    "ParameterValue": "10000",
    "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "slowlog-max-len",
    "ParameterValue": "128",
    "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
```

```
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "stream-node-max-bytes",
    "ParameterValue": "4096",
    "Description": "The maximum size of a single node in a stream in bytes",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "stream-node-max-entries",
    "ParameterValue": "100",
    "Description": "The maximum number of items a single node in a stream
can contain",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "tcp-keepalive",
    "ParameterValue": "300",
    "Description": "If non-zero, send ACKs every given number of seconds.",
    "Source": "user",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "timeout",
    "ParameterValue": "0",
    "Description": "Close connection if client is idle for a given number of
seconds, or never if 0.",
    "Source": "user",
```

```

        "DataType": "integer",
        "AllowedValues": "0,20-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "zset-max-ziplist-entries",
        "ParameterValue": "128",
        "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed.",
        "Source": "user",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "zset-max-ziplist-value",
        "ParameterValue": "64",
        "Description": "The threshold of biggest sorted set entries in order for
the dataset to be compressed.",
        "Source": "user",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    }
]
}

```

詳細については、ElastiCache ユーザーガイドの「[パラメータ管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCacheParameters](#)」の「」を参照してください。

describe-cache-subnet-groups

次の例は、describe-cache-subnet-groups を使用する方法を説明しています。

AWS CLI

キャッシュサブネットグループを記述するには

次のdescribe-cache-subnet-groups例では、サブネットグループのリストを返します。

```
aws elasticache describe-cache-subnet-groups
```

出力:

```
{
  "CacheSubnetGroups": [
    {
      "CacheSubnetGroupName": "default",
      "CacheSubnetGroupDescription": "Default CacheSubnetGroup",
      "VpcId": "vpc-a3e97cdb",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-8d4bacf5",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2b"
          }
        },
        {
          "SubnetIdentifier": "subnet-dde21380",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2c"
          }
        },
        {
          "SubnetIdentifier": "subnet-6485ec4f",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2d"
          }
        },
        {
          "SubnetIdentifier": "subnet-b4ebebff",
          "SubnetAvailabilityZone": {
            "Name": "us-west-2a"
          }
        }
      ]
    }
  ],
}
```

```
{
  "CacheSubnetGroupName": "kxxkk",
  "CacheSubnetGroupDescription": "mygroup",
  "VpcId": "vpc-a3e97cdb",
  "Subnets": [
    {
      "SubnetIdentifier": "subnet-b4ebebff",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      }
    }
  ]
},
{
  "CacheSubnetGroupName": "test",
  "CacheSubnetGroupDescription": "test",
  "VpcId": "vpc-a3e97cdb",
  "Subnets": [
    {
      "SubnetIdentifier": "subnet-b4ebebff",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      }
    }
  ]
}
]
```

詳細については、[「Elasticache ユーザーガイド」の「サブネットとサブネットグループ」](#)または「for ElastiCache Memcached ユーザーガイド」の[「サブネットとサブネットグループ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCacheSubnetGroups](#)」の「」を参照してください。

describe-engine-default-parameters

次の例は、describe-engine-default-parameters を使用する方法を説明しています。

AWS CLI

エンジンのデフォルトパラメータを記述するには

次のdescribe-engine-default-parameters例では、指定されたキャッシュエンジンのデフォルトのエンジンおよびシステムパラメータ情報を返します。

```
aws elasticache describe-engine-default-parameters \  
--cache-parameter-group-family "redis5.0"
```

出力:

```
{  
  "EngineDefaults": {  
    "Parameters": [  
      {  
        "ParameterName": "activedefrag",  
        "ParameterValue": "no",  
        "Description": "Enabled active memory defragmentation",  
        "Source": "system",  
        "DataType": "string",  
        "AllowedValues": "yes,no",  
        "IsModifiable": true,  
        "MinimumEngineVersion": "5.0.0",  
        "ChangeType": "immediate"  
      },  
      {  
        "ParameterName": "active-defrag-cycle-max",  
        "ParameterValue": "75",  
        "Description": "Maximal effort for defrag in CPU percentage",  
        "Source": "system",  
        "DataType": "integer",  
        "AllowedValues": "1-75",  
        "IsModifiable": true,  
        "MinimumEngineVersion": "5.0.0",  
        "ChangeType": "immediate"  
      },  
      {  
        "ParameterName": "active-defrag-cycle-min",  
        "ParameterValue": "5",  
        "Description": "Minimal effort for defrag in CPU percentage",  
        "Source": "system",  
        "DataType": "integer",  
        "AllowedValues": "1-75",  
        "IsModifiable": true,  
        "MinimumEngineVersion": "5.0.0",  
        "ChangeType": "immediate"  
      }  
    ]  
  }  
}
```

```
    },
    {
      "ParameterName": "active-defrag-ignore-bytes",
      "ParameterValue": "104857600",
      "Description": "Minimum amount of fragmentation waste to start
active defrag",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "1048576-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "active-defrag-max-scan-fields",
      "ParameterValue": "1000",
      "Description": "Maximum number of set/hash/zset/list fields that
will be processed from the main dictionary scan",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "1-1000000",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "active-defrag-threshold-lower",
      "ParameterValue": "10",
      "Description": "Minimum percentage of fragmentation to start active
defrag",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "1-100",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "active-defrag-threshold-upper",
      "ParameterValue": "100",
      "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
      "Source": "system",
      "DataType": "integer",
```

```
    "AllowedValues": "1-100",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "activeresharding",
    "ParameterValue": "yes",
    "Description": "Apply rehashing or not.",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "appendfsync",
    "ParameterValue": "everysec",
    "Description": "fsync policy for AOF persistence",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "always,everysec,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "appendonly",
    "ParameterValue": "no",
    "Description": "Enable Redis persistence.",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-hard-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer hard limit in bytes.",
    "Source": "system",
    "DataType": "integer",
```

```
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-limit",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in bytes.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
    "ParameterValue": "0",
    "Description": "Normal client output buffer soft limit in seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
    "ParameterValue": "33554432",
    "Description": "Pubsub client output buffer hard limit in bytes.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
    "ParameterValue": "8388608",
    "Description": "Pubsub client output buffer soft limit in bytes.",
    "Source": "system",
    "DataType": "integer",
```

```
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
    "ParameterValue": "60",
    "Description": "Pubsub client output buffer soft limit in seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
    "ParameterValue": "60",
    "Description": "Replica client output buffer soft limit in
seconds.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "client-query-buffer-limit",
    "ParameterValue": "1073741824",
    "Description": "Max size of a single client query buffer",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1048576-1073741824",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "close-on-replica-write",
    "ParameterValue": "yes",
    "Description": "If enabled, clients who attempt to write to a read-
only replica will be disconnected. Applicable to 2.8.23 and higher.",
```

```
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "cluster-enabled",
    "ParameterValue": "no",
    "Description": "Enable cluster mode",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "cluster-require-full-coverage",
    "ParameterValue": "no",
    "Description": "Whether cluster becomes unavailable if one or more
slots are not covered",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "databases",
    "ParameterValue": "16",
    "Description": "Set the number of databases.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-1200000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "hash-max-ziplist-entries",
    "ParameterValue": "512",
```



```
        "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "hash-max-ziplist-value",
        "ParameterValue": "64",
        "Description": "The threshold of biggest hash entries in order for
the dataset to be compressed.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "hll-sparse-max-bytes",
        "ParameterValue": "3000",
        "Description": "HyperLogLog sparse representation bytes limit",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "1-16000",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "lazyfree-lazy-eviction",
        "ParameterValue": "no",
        "Description": "Perform an asynchronous delete on evictions",
        "Source": "system",
        "DataType": "string",
        "AllowedValues": "yes,no",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
```

```
    "ParameterName": "lazyfree-lazy-expire",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous delete on expired keys",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lazyfree-lazy-server-del",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous delete on key updates",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lfu-decay-time",
    "ParameterValue": "1",
    "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "lfu-log-factor",
    "ParameterValue": "10",
    "Description": "The log factor for incrementing key counter for LFU
eviction policy",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  }
}
```

```
    },
    {
      "ParameterName": "list-compress-depth",
      "ParameterValue": "0",
      "Description": "Number of quicklist ziplist nodes from each side
of the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "list-max-ziplist-size",
      "ParameterValue": "-2",
      "Description": "The number of entries allowed per internal list node
can be specified as a fixed maximum size or a maximum number of elements",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "-5,-4,-3,-2,-1,1-",
      "IsModifiable": false,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "lua-replicate-commands",
      "ParameterValue": "yes",
      "Description": "Always enable Lua effect replication or not",
      "Source": "system",
      "DataType": "string",
      "AllowedValues": "yes,no",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "lua-time-limit",
      "ParameterValue": "5000",
      "Description": "Max execution time of a Lua script in milliseconds.
0 for unlimited execution without warnings.",
      "Source": "system",
      "DataType": "integer",
```

```
    "AllowedValues": "5000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "maxclients",
    "ParameterValue": "65000",
    "Description": "The maximum number of Redis clients.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-65000",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "requires-reboot"
  },
  {
    "ParameterName": "maxmemory-policy",
    "ParameterValue": "volatile-lru",
    "Description": "Max memory policy.",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "maxmemory-samples",
    "ParameterValue": "3",
    "Description": "Max memory samples.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "1-",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "min-replicas-max-lag",
    "ParameterValue": "10",
```

```

        "Description": "The maximum amount of replica lag in seconds beyond
which the master would stop taking writes. A value of 0 means the master always
takes writes.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "min-replicas-to-write",
        "ParameterValue": "0",
        "Description": "The minimum number of replicas that must be present
with lag no greater than min-replicas-max-lag for master to take writes. Setting
this to 0 means the master always takes writes.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "notify-keyspace-events",
        "Description": "The keyspace events for Redis to notify Pub/Sub
clients about. By default all notifications are disabled",
        "Source": "system",
        "DataType": "string",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "proto-max-bulk-len",
        "ParameterValue": "536870912",
        "Description": "Max size of a single element request",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "1048576-536870912",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
},

```

```

    {
      "ParameterName": "rename-commands",
      "ParameterValue": "",
      "Description": "Redis commands that can be dynamically renamed by
the customer",
      "Source": "system",
      "DataType": "string",
      "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.3",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "repl-backlog-size",
      "ParameterValue": "1048576",
      "Description": "The replication backlog size in bytes for PSYNC.
This is the size of the buffer which accumulates slave data when slave is
disconnected for some time, so that when slave reconnects again, only transfer the
portion of data which the slave missed. Minimum value is 16K.",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "16384-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "repl-backlog-ttl",
      "ParameterValue": "3600",
      "Description": "The amount of time in seconds after the master no
longer have any slaves connected for the master to free the replication backlog. A
value of 0 means to never release the backlog.",
      "Source": "system",
      "DataType": "integer",
      "AllowedValues": "0-",
      "IsModifiable": true,
      "MinimumEngineVersion": "5.0.0",
      "ChangeType": "immediate"
    },
    {
      "ParameterName": "replica-allow-chaining",
      "ParameterValue": "no",
      "Description": "Configures if chaining of replicas is allowed",

```

```
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-ignore-maxmemory",
    "ParameterValue": "yes",
    "Description": "Determines if replica ignores maxmemory setting by
not evicting items independent from the master",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "replica-lazy-flush",
    "ParameterValue": "no",
    "Description": "Perform an asynchronous flushDB during replica
sync",
    "Source": "system",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "IsModifiable": false,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
  {
    "ParameterName": "reserved-memory-percent",
    "ParameterValue": "25",
    "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
    "Source": "system",
    "DataType": "integer",
    "AllowedValues": "0-100",
    "IsModifiable": true,
    "MinimumEngineVersion": "5.0.0",
    "ChangeType": "immediate"
  },
},
```

```
{
  "ParameterName": "set-max-intset-entries",
  "ParameterValue": "512",
  "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
  "Source": "system",
  "DataType": "integer",
  "AllowedValues": "0-",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "slowlog-log-slower-than",
  "ParameterValue": "10000",
  "Description": "The execution time, in microseconds, to exceed in
order for the command to get logged. Note that a negative number disables the slow
log, while a value of zero forces the logging of every command.",
  "Source": "system",
  "DataType": "integer",
  "AllowedValues": "-",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "slowlog-max-len",
  "ParameterValue": "128",
  "Description": "The length of the slow log. There is no limit to
this length. Just be aware that it will consume memory. You can reclaim memory used
by the slow log with SLOWLOG RESET.",
  "Source": "system",
  "DataType": "integer",
  "AllowedValues": "0-",
  "IsModifiable": true,
  "MinimumEngineVersion": "5.0.0",
  "ChangeType": "immediate"
},
{
  "ParameterName": "stream-node-max-bytes",
  "ParameterValue": "4096",
  "Description": "The maximum size of a single node in a stream in
bytes",
  "Source": "system",
```



```
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "stream-node-max-entries",
        "ParameterValue": "100",
        "Description": "The maximum number of items a single node in a
stream can contain",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "tcp-keepalive",
        "ParameterValue": "300",
        "Description": "If non-zero, send ACKs every given number of
seconds.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "timeout",
        "ParameterValue": "0",
        "Description": "Close connection if client is idle for a given
number of seconds, or never if 0.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0,20-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "zset-max-ziplist-entries",
```

```

        "ParameterValue": "128",
        "Description": "The maximum number of sorted set entries in order
for the dataset to be compressed.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    },
    {
        "ParameterName": "zset-max-ziplist-value",
        "ParameterValue": "64",
        "Description": "The threshold of biggest sorted set entries in order
for the dataset to be compressed.",
        "Source": "system",
        "DataType": "integer",
        "AllowedValues": "0-",
        "IsModifiable": true,
        "MinimumEngineVersion": "5.0.0",
        "ChangeType": "immediate"
    }
]
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEngineDefaultParameters](#)」の「」を参照してください。

describe-events

次の例は、describe-events を使用する方法を説明しています。

AWS CLI

レプリケーショングループのイベントを記述するには

次のdescribe-events例では、レプリケーショングループのイベントのリストを返します。

```

aws elasticache describe-events \
  --source-identifier test-cluster \
  --source-type replication-group

```

出力:

```
{
  "Events": [
    {
      "SourceIdentifier": "test-cluster",
      "SourceType": "replication-group",
      "Message": "Automatic failover has been turned on for replication group
test-cluster",
      "Date": "2020-03-18T23:51:34.457Z"
    },
    {
      "SourceIdentifier": "test-cluster",
      "SourceType": "replication-group",
      "Message": "Replication group test-cluster created",
      "Date": "2020-03-18T23:50:31.378Z"
    }
  ]
}
```

詳細については、Elasticache ユーザーガイドの「[イベントのモニタリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

describe-global-replication-groups

次の例は、describe-global-replication-groups を使用する方法を説明しています。

AWS CLI

グローバルレプリケーショングループを記述するには

次のdescribe-global-replication-groups例では、グローバルデータストアの詳細を返します。

```
aws elasticache describe-global-replication-groups \
  --global-replication-group-id my-grg
```

出力:

```
{
  "GlobalReplicationGroups": [
    {
      "GlobalReplicationGroupId": "my-grg",
      "GlobalReplicationGroupDescription": "my-grg",
      "Status": "creating",
      "CacheNodeType": "cache.r5.large",
      "Engine": "redis",
      "EngineVersion": "5.0.6",
      "ClusterEnabled": false,
      "AuthTokenEnabled": false,
      "TransitEncryptionEnabled": false,
      "AtRestEncryptionEnabled": false
    }
  ]
}
```

詳細については、Elasticache ユーザーガイドの「[グローバルデータストアを使用した AWS リージョン間のレプリケーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGlobalReplicationGroups](#)」の「」を参照してください。

describe-replication-groups

次の例は、describe-replication-groups を使用する方法を説明しています。

AWS CLI

レプリケーショングループの詳細のリストを返すには

次のdescribe-replication-groups例では、レプリケーショングループを返します。

```
aws elasticache describe-replication-groups
```

出力:

```
{
  "ReplicationGroups": [
    {
      "ReplicationGroupId": "my-cluster",
      "Description": "mycluster",

```

```
"Status": "available",
"PendingModifiedValues": {},
"MemberClusters": [
  "pat-cluster-001",
  "pat-cluster-002",
  "pat-cluster-003",
  "pat-cluster-004"
],
"NodeGroups": [
  {
    "NodeGroupId": "0001",
    "Status": "available",
    "PrimaryEndpoint": {
      "Address": "my-
cluster.xxxxih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "ReaderEndpoint": {
      "Address": "my-cluster-
ro.xxxxih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "NodeGroupMembers": [
      {
        "CacheClusterId": "my-cluster-001",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "pat-
cluster-001.xxxxih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "primary"
      },
      {
        "CacheClusterId": "my-cluster-002",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "pat-
cluster-002.xxxxih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "replica"
      }
    ]
  }
]
```

```

        },
        {
            "CacheClusterId": "my-cluster-003",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "pat-
cluster-003.xxxxih.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
        },
        {
            "CacheClusterId": "my-cluster-004",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
                "Address": "pat-
cluster-004.xxxih.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2a",
            "CurrentRole": "replica"
        }
    ]
}
],
"AutomaticFailover": "disabled",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.xlarge",
"AuthTokenEnabled": false,
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxxx152:replicationgroup:my-cluster",
"LogDeliveryConfigurations": [
    {
        "LogType": "slow-log",
        "DestinationType": "cloudwatch-logs",
        "DestinationDetails": {
            "CloudWatchLogsDetails": {
                "LogGroup": "test-log"
            }
        }
    }
]

```

```
    },
    "LogFormat": "json",
    "Status": "active"
  }
]
}
```

詳細については、Elasticache ユーザーガイドの「[クラスターの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReplicationGroups](#)」の「」を参照してください。

describe-reserved-cache-nodes-offerings

次の例は、describe-reserved-cache-nodes-offerings を使用する方法を説明しています。

AWS CLI

説明するには reserved-cache-nodes-offerings

次のdescribe-reserved-cache-nodes-offerings例では、reserved-cache-nodeオプションの詳細を返します。

```
aws elasticache describe-reserved-cache-nodes-offerings
```

出力:

```
{
  "ReservedCacheNodesOfferings": [
    {
      "ReservedCacheNodesOfferingId": "01ce0a19-a476-41cb-8aee-48eachbc8e5",
      "CacheNodeType": "cache.t3.small",
      "Duration": 31536000,
      "FixedPrice": 97.0,
      "UsagePrice": 0.0,
      "ProductDescription": "memcached",
      "OfferingType": "Partial Upfront",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.011,
          "RecurringChargeFrequency": "Hourly"
        }
      ]
    }
  ]
}
```

```
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0443a27b-4da5-4b90-b92d-929fbd7abed2",
  "CacheNodeType": "cache.m3.2xlarge",
  "Duration": 31536000,
  "FixedPrice": 1772.0,
  "UsagePrice": 0.0,
  "ProductDescription": "redis",
  "OfferingType": "Heavy Utilization",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": 0.25,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
...
]
```

詳細については、「[Elasticache Redis ユーザーガイド](#)」の「[リザーブドノードオフリングに関する情報の取得](#)」または「[Elasticache Memcached ユーザーガイド](#)」の「[リザーブドノードオフリングに関する情報の取得](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeReservedCacheNodesOfferings](#)」の「」を参照してください。

describe-reserved-cache-nodes

次のコード例は、describe-reserved-cache-nodes を使用する方法を示しています。

AWS CLI

予約済みキャッシュノードを記述するには

次のdescribe-reserved-cache-nodes例では、このアカウントのリザーブドキャッシュノードに関する情報、または指定されたリザーブドキャッシュノードに関する情報を返します。

AWS Elasticache describe-reserved-cache-nodes

出力:

```
{
  "ReservedCacheNodes": [
    {
      "ReservedCacheNodeId": "mynode",
      "ReservedCacheNodesOfferingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxx71",
      "CacheNodeType": "cache.t3.small",
      "StartTime": "2019-12-06T02:50:44.003Z",
      "Duration": 31536000,
      "FixedPrice": 0.0,
      "UsagePrice": 0.0,
      "CacheNodeCount": 1,
      "ProductDescription": "redis",
      "OfferingType": "No Upfront",
      "State": "payment-pending",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.023,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "ReservationARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxxx52:reserved-instance:mynode"
    }
  ]
}
```

詳細については、Elasticache ユーザーガイドの「[リザーブドノードによるコストの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReservedCacheNodes](#)」の「」を参照してください。

describe-service-updates

次のコード例は、describe-service-updates を使用する方法を示しています。

AWS CLI

サービスの更新を記述するには

次のdescribe-service-updates例では、サービスの更新に関する詳細を返します。

aws elasticache describe-service-updates

出力:

```
{
  "ServiceUpdates": [
    {
      "ServiceUpdateName": "elc-xxxxxxx7-001",
      "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
      "ServiceUpdateEndDate": "2020-02-09T15:59:59Z",
      "ServiceUpdateSeverity": "important",
      "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
      "ServiceUpdateStatus": "available",
      "ServiceUpdateDescription": "Upgrades to improve the security,
reliability, and operational performance of your ElastiCache nodes",
      "ServiceUpdateType": "security-update",
      "Engine": "redis, memcached",
      "EngineVersion": "redis 2.6.13 and onwards, memcached 1.4.5 and
onwards",
      "AutoUpdateAfterRecommendedApplyByDate": false,
      "EstimatedUpdateTime": "30 minutes per node"
    },
    {
      "ServiceUpdateName": "elc-xxxxxxx4-001",
      "ServiceUpdateReleaseDate": "2019-06-11T15:00:00Z",
      "ServiceUpdateEndDate": "2019-10-01T09:24:00Z",
      "ServiceUpdateSeverity": "important",
      "ServiceUpdateRecommendedApplyByDate": "2019-07-11T14:59:59Z",
      "ServiceUpdateStatus": "expired",
      "ServiceUpdateDescription": "Upgrades to improve the security,
reliability, and operational performance of your ElastiCache nodes",
      "ServiceUpdateType": "security-update",
      "Engine": "redis",
      "EngineVersion": "redis 3.2.6, redis 4.0 and onwards",
      "AutoUpdateAfterRecommendedApplyByDate": false,
      "EstimatedUpdateTime": "30 minutes per node"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeServiceUpdates](#)」の「」を参照してください。

describe-snapshots

次の例は、describe-snapshots を使用する方法を説明しています。

AWS CLI

スナップショットを記述するには

次の「describe-snapshots」の例は、クラスターまたはレプリケーショングループのスナップショットに関する情報を返します。

```
aws elasticache describe-snapshots
```

出力:

```
{
  "Snapshots": [
    {
      "SnapshotName": "automatic.my-cluster2-002-2019-12-05-06-38",
      "CacheClusterId": "my-cluster2-002",
      "SnapshotStatus": "available",
      "SnapshotSource": "automated",
      "CacheNodeType": "cache.r5.large",
      "Engine": "redis",
      "EngineVersion": "5.0.5",
      "NumCacheNodes": 1,
      "PreferredAvailabilityZone": "us-west-2a",
      "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
      "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
      "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxx52:My_Topic",
      "Port": 6379,
      "CacheParameterGroupName": "default.redis5.0",
      "CacheSubnetGroupName": "kxkxk",
      "VpcId": "vpc-a3e97cdb",
      "AutoMinorVersionUpgrade": true,
      "SnapshotRetentionLimit": 1,
      "SnapshotWindow": "06:30-07:30",
      "NodeSnapshots": [
        {
          "CacheNodeId": "0001",
          "CacheSize": "5 MB",
          "CacheNodeCreateTime": "2019-11-26T01:22:52.396Z",
          "SnapshotCreateTime": "2019-12-05T06:38:23Z"
        }
      ]
    }
  ]
}
```

```
    }
  ]
},
{
  "SnapshotName": "myreplica-backup",
  "CacheClusterId": "myreplica",
  "SnapshotStatus": "available",
  "SnapshotSource": "manual",
  "CacheNodeType": "cache.r5.large",
  "Engine": "redis",
  "EngineVersion": "5.0.5",
  "NumCacheNodes": 1,
  "PreferredAvailabilityZone": "us-west-2a",
  "CacheClusterCreateTime": "2019-11-26T00:14:52.439Z",
  "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
  "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
  "Port": 6379,
  "CacheParameterGroupName": "default.redis5.0",
  "CacheSubnetGroupName": "kxkxk",
  "VpcId": "vpc-a3e97cdb",
  "AutoMinorVersionUpgrade": true,
  "SnapshotRetentionLimit": 0,
  "SnapshotWindow": "09:00-10:00",
  "NodeSnapshots": [
    {
      "CacheNodeId": "0001",
      "CacheSize": "5 MB",
      "CacheNodeCreateTime": "2019-11-26T00:14:52.439Z",
      "SnapshotCreateTime": "2019-11-26T00:25:01Z"
    }
  ]
},
{
  "SnapshotName": "my-cluster",
  "CacheClusterId": "my-cluster-003",
  "SnapshotStatus": "available",
  "SnapshotSource": "manual",
  "CacheNodeType": "cache.r5.large",
  "Engine": "redis",
  "EngineVersion": "5.0.5",
  "NumCacheNodes": 1,
  "PreferredAvailabilityZone": "us-west-2a",
  "CacheClusterCreateTime": "2019-11-25T23:56:17.186Z",
  "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
```

```

    "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",
    "Port": 6379,
    "CacheParameterGroupName": "default.redis5.0",
    "CacheSubnetGroupName": "kxxkxk",
    "VpcId": "vpc-a3e97cdb",
    "AutoMinorVersionUpgrade": true,
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "09:00-10:00",
    "NodeSnapshots": [
      {
        "CacheNodeId": "0001",
        "CacheSize": "5 MB",
        "CacheNodeCreateTime": "2019-11-25T23:56:17.186Z",
        "SnapshotCreateTime": "2019-11-26T03:08:33Z"
      }
    ]
  }
]
}

```

詳細については、Elasticache ユーザーガイドの「[Redis ElastiCache のバックアップと復元](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSnapshots](#)」の「」を参照してください。

describe-update-actions

次の例は、describe-update-actions を使用する方法を説明しています。

AWS CLI

更新アクションを記述するには

次のdescribe-update-actions例では、更新アクションの詳細を返します。

```
aws elasticache describe-update-actions
```

出力:

```

{
  "UpdateActions": [
    {

```

```
"ReplicationGroupId": "mycluster",
"ServiceUpdateName": "elc-20191007-001",
"ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
"ServiceUpdateSeverity": "important",
"ServiceUpdateStatus": "available",
"ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
"ServiceUpdateType": "security-update",
"UpdateActionAvailableDate": "2019-12-05T19:15:19.995Z",
"UpdateActionStatus": "complete",
"NodesUpdated": "9/9",
"UpdateActionStatusModifiedDate": "2019-12-05T19:15:20.461Z",
"SlaMet": "n/a",
"Engine": "redis"
},
{
  "CacheClusterId": "my-memcached-cluster",
  "ServiceUpdateName": "elc-20191007-001",
  "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
  "ServiceUpdateSeverity": "important",
  "ServiceUpdateStatus": "available",
  "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
  "ServiceUpdateType": "security-update",
  "UpdateActionAvailableDate": "2019-12-04T18:26:05.349Z",
  "UpdateActionStatus": "complete",
  "NodesUpdated": "1/1",
  "UpdateActionStatusModifiedDate": "2019-12-04T18:26:05.352Z",
  "SlaMet": "n/a",
  "Engine": "redis"
},
{
  "ReplicationGroupId": "my-cluster",
  "ServiceUpdateName": "elc-20191007-001",
  "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
  "ServiceUpdateSeverity": "important",
  "ServiceUpdateStatus": "available",
  "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
  "ServiceUpdateType": "security-update",
  "UpdateActionAvailableDate": "2019-11-26T03:36:26.320Z",
  "UpdateActionStatus": "complete",
  "NodesUpdated": "4/4",
  "UpdateActionStatusModifiedDate": "2019-12-04T22:11:12.664Z",
  "SlaMet": "n/a",
  "Engine": "redis"
},
```

```
{
  "ReplicationGroupId": "my-cluster2",
  "ServiceUpdateName": "elc-20191007-001",
  "ServiceUpdateReleaseDate": "2019-10-09T16:00:00Z",
  "ServiceUpdateSeverity": "important",
  "ServiceUpdateStatus": "available",
  "ServiceUpdateRecommendedApplyByDate": "2019-11-08T15:59:59Z",
  "ServiceUpdateType": "security-update",
  "UpdateActionAvailableDate": "2019-11-26T01:26:01.617Z",
  "UpdateActionStatus": "complete",
  "NodesUpdated": "3/3",
  "UpdateActionStatusModifiedDate": "2019-11-26T01:26:01.753Z",
  "SlaMet": "n/a",
  "Engine": "redis"
}
]
```

詳細については、ElastiCache ユーザーガイドの「[Amazon でのセルフサービスの更新 ElastiCache](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUpdateActions](#)」の「」を参照してください。

describe-user-groups

次の例は、describe-user-groups を使用する方法を説明しています。

AWS CLI

ユーザーグループを記述するには

次のdescribe-user-groups例では、ユーザーグループのリストを返します。

```
aws elasticache describe-user-groups
```

出力:

```
{
  "UserGroups": [
    {
```

```
    "UserGroupId": "myusergroup",
    "Status": "active",
    "Engine": "redis",
    "UserIds": [
      "default"
    ],
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxx52:usergroup:myusergroup"
  }
]
}
```

詳細については、Elasticache ユーザーガイドの「[ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUserGroups](#)」の「」を参照してください。

describe-users

次の例は、describe-users を使用する方法を説明しています。

AWS CLI

ユーザーを記述するには

次のdescribe-users例では、ユーザーのリストを返します。

```
aws elasticache describe-users
```

出力:

```
{
  "Users": [
    {
      "UserId": "default",
      "UserName": "default",
      "Status": "active",
      "Engine": "redis",
      "AccessString": "on ~* +@all",
```



```

    "UserGroupIds": [
      "myusergroup"
    ],
    "Authentication": {
      "Type": "no-password"
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:default"
  },
  {
    "UserId": "user1",
    "UserName": "myUser",
    "Status": "active",
    "Engine": "redis",
    "AccessString": "on ~* +@all",
    "UserGroupIds": [],
    "Authentication": {
      "Type": "password",
      "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user1"
  },
  {
    "UserId": "user2",
    "UserName": "myUser",
    "Status": "active",
    "Engine": "redis",
    "AccessString": "on ~app:* -@all +@read +@hash +@bitmap +@geo -setbit -
bitfield -hset -hsetnx -hmset -hincrby -hincrbyfloat -hdel -bitop -geoadd -georadius
-georadiusbymember",
    "UserGroupIds": [],
    "Authentication": {
      "Type": "password",
      "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
  }
]
}

```

詳細については、Elasticache ユーザーガイドの「[ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeUsers](#)」の「」を参照してください。

disassociate-global-replication-group

次の例は、disassociate-global-replication-group を使用する方法を説明しています。

AWS CLI

グローバルレプリケーショングループからセカンダリクラスターの関連付けを解除するには

次のdisassociate-global-replication-group例では、グローバルデータストアからセカンダリクラスターを削除します。

```
aws elasticache disassociate-global-replication-group \  
  --global-replication-group-id my-grg \  
  --replication-group-id my-cluster-grg-secondary \  
  --replication-group-region us-east-1
```

出力:

```
{  
  "GlobalReplicationGroup": {  
    "GlobalReplicationGroupId": "my-grg",  
    "GlobalReplicationGroupDescription": "my-grg",  
    "Status": "modifying",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "Members": [  
      {  
        "ReplicationGroupId": "my-cluster-grg-secondary",  
        "ReplicationGroupRegion": "us-east-1",  
        "Role": "SECONDARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      },  
      {  
        "ReplicationGroupId": "my-cluster-grg",  
        "ReplicationGroupRegion": "us-west-2",  
        "Role": "PRIMARY",  
        "AutomaticFailover": "enabled",  
        "Status": "associated"  
      }  
    ],  
    "ClusterEnabled": false,  
  }  
}
```

```
    "AuthTokenEnabled": false,  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

詳細については、Elasticache ユーザーガイドの「[グローバルデータストアを使用した AWS リージョン間のレプリケーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateGlobalReplicationGroup](#)」の「」を参照してください。

increase-node-groups-in-global-replication-group

次のコード例は、`increase-node-groups-in-global-replication-group` を使用する方法を示しています。

AWS CLI

グローバルレプリケーショングループ内のノードグループの数を増やすには

以下は、Redis エンジンを使用してノードグループ数 `increase-node-groups-in-global-replication-group` を増やします。

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --global-replication-group-id sgaui-pat-test-4 \  
  --node-group-count 6 \  
  --apply-immediately
```

出力:

```
{  
  "GlobalReplicationGroup": {  
    "GlobalReplicationGroupId": "sgaui-test-4",  
    "GlobalReplicationGroupDescription": "test-4",  
    "Status": "modifying",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "Members": [  
      {  
        "ReplicationGroupId": "my-cluster-b",
```

```
        "ReplicationGroupRegion": "us-east-1",
        "Role": "SECONDARY",
        "AutomaticFailover": "enabled",
        "Status": "associated"
    },
    {
        "ReplicationGroupId": "my-cluster-a",
        "ReplicationGroupRegion": "us-west-2",
        "Role": "PRIMARY",
        "AutomaticFailover": "enabled",
        "Status": "associated"
    }
],
"ClusterEnabled": true,
"GlobalNodeGroups": [
    {
        "GlobalNodeId": "sgaui-test-4-0001",
        "Slots": "0-234,2420-5461"
    },
    {
        "GlobalNodeId": "sgaui-test-4-0002",
        "Slots": "5462-5904,6997-9830"
    },
    {
        "GlobalNodeId": "sgaui-test-4-0003",
        "Slots": "10923-11190,13375-16383"
    },
    {
        "GlobalNodeId": "sgaui-test-4-0004",
        "Slots": "235-2419,5905-6996"
    },
    {
        "GlobalNodeId": "sgaui-test-4-0005",
        "Slots": "9831-10922,11191-13374"
    }
],
"AuthTokenEnabled": false,
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
```

詳細については、Elasticache ユーザーガイドの「[グローバルデータストアを使用した AWS リージョン間のレプリケーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [IncreaseNodeGroupsInGlobalReplicationGroup](#)」の「」を参照してください。

increase-replica-count

次のコード例は、increase-replica-count を使用する方法を示しています。

AWS CLI

レプリカ数を増やすには

次のincrease-replica-count例では、2 つのことのいずれかを実行します。Redis (クラスターモードが無効) レプリケーショングループのレプリカの数を実動的に増やすことができます。または、Redis (クラスターモードが有効) レプリケーショングループの 1 つ以上のノードグループ (シャード) 内のレプリカノードの数を実動的に増やすこともできます。このオペレーションは、クラスターのダウンタイムなしで実行されます。

```
aws elasticache increase-replica-count \  
  --replication-group-id "my-cluster" \  
  --apply-immediately \  
  --new-replica-count 3
```

出力:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "my-cluster",  
    "Description": " ",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "my-cluster-001",  
      "my-cluster-002",  
      "my-cluster-003",  
      "my-cluster-004"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",
```

```
    "Status": "modifying",
    "PrimaryEndpoint": {
      "Address": "my-
cluster.xxxxxih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "ReaderEndpoint": {
      "Address": "my-cluster-
ro.xxxxxxih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "NodeGroupMembers": [
      {
        "CacheClusterId": "my-cluster-001",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "my-
cluster-001.xxxxxih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "primary"
      },
      {
        "CacheClusterId": "my-cluster-003",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "my-
cluster-003.xxxxxih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "replica"
      }
    ]
  }
},
"AutomaticFailover": "disabled",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.xlarge",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
```

```
}  
}
```

詳細については、Elasticache ユーザーガイドの「[シャードのレプリカ数を増やす](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[IncreaseReplicaCount](#)」の「」を参照してください。

list-allowed-node-type-modifications

次の例は、list-allowed-node-type-modifications を使用する方法を説明しています。

AWS CLI

許可されたノードの変更を一覧表示するには

次のlist-allowed-node-type-modifications例では、Redis クラスターまたはレプリケーショングループの現在のノードタイプをスケールできるすべての使用可能なノードタイプを一覧表示します。

```
aws elasticache list-allowed-node-type-modifications \  
--replication-group-id "my-replication-group"
```

出力:

```
{  
  "ScaleUpModifications": [  
    "cache.m5.12xlarge",  
    "cache.m5.24xlarge",  
    "cache.m5.4xlarge",  
    "cache.r5.12xlarge",  
    "cache.r5.24xlarge",  
    "cache.r5.2xlarge",  
    "cache.r5.4xlarge"  
  ],  
  "ScaleDownModifications": [  
    "cache.m3.large",  
    "cache.m3.medium",  
    "cache.m3.xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",
```

```
    "cache.m5.2xlarge",
    "cache.m5.large",
    "cache.m5.xlarge",
    "cache.r3.large",
    "cache.r4.large",
    "cache.r4.xlarge",
    "cache.r5.large",
    "cache.t2.medium",
    "cache.t2.micro",
    "cache.t2.small",
    "cache.t3.medium",
    "cache.t3.micro",
    "cache.t3.small"
  ]
}
```

詳細については、Elasticache ユーザーガイドの「[Redis クラスター ElastiCache のスケーリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAllowedNodeTypeModifications](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、リソースのタグを一覧表示します。

```
aws elasticache list-tags-for-resource \
  --resource-name "arn:aws:elasticache:us-east-1:123456789012:cluster:my-cluster"
```

出力:

```
{
  "TagList": [
    {
      "Key": "Project",
      "Value": "querySpeedUp"
    },
  ],
}
```



```
{
  "Key": "Environment",
  "Value": "PROD"
}
]
```

詳細については、Elasticache ユーザーガイドの「[を使用したタグの一覧表示 AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

modify-cache-cluster

次の例は、modify-cache-cluster を使用する方法を説明しています。

AWS CLI

キャッシュクラスターを変更するには

次のmodify-cache-cluster例では、指定されたクラスターの設定を変更します。

```
aws elasticache modify-cache-cluster \
  --cache-cluster-id "my-cluster" \
  --num-cache-nodes 1
```

出力:

```
{
  "CacheCluster": {
    "CacheClusterId": "my-cluster",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheNodeType": "cache.m5.large",
    "Engine": "redis",
    "EngineVersion": "5.0.5",
    "CacheClusterStatus": "available",
    "NumCacheNodes": 1,
    "PreferredAvailabilityZone": "us-west-2c",
    "CacheClusterCreateTime": "2019-12-04T18:24:56.652Z",
    "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
```

```

    "PendingModifiedValues": {},
    "CacheSecurityGroups": [],
    "CacheParameterGroup": {
        "CacheParameterGroupName": "default.redis5.0",
        "ParameterApplyStatus": "in-sync",
        "CacheNodeIdsToReboot": []
    },
    "CacheSubnetGroupName": "default",
    "AutoMinorVersionUpgrade": true,
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "07:00-08:00",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
}
}

```

詳細については、Elasticache ユーザーガイドの [ElastiCache 「クラスターの変更」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyCacheCluster](#)」の「」を参照してください。

modify-cache-parameter-group

次のコード例は、modify-cache-parameter-group を使用する方法を示しています。

AWS CLI

キャッシュパラメータグループを変更するには

次のmodify-cache-parameter-group例では、指定されたキャッシュパラメータグループのパラメータを変更します。

```

aws elasticache modify-cache-parameter-group \
  --cache-parameter-group-name "mygroup" \
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"

```

出力:

```

{
  "CacheParameterGroupName": "mygroup"
}

```

詳細については、Elasticache ユーザーガイドの [「パラメータグループの変更」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyCacheParameterGroup](#)」の「」を参照してください。

modify-cache-subnet-group

次のコード例は、modify-cache-subnet-group を使用する方法を示しています。

AWS CLI

キャッシュサブネットグループを変更するには

次のmodify-cache-subnet-group例では、指定されたキャッシュサブネットグループを変更します。

```
aws elasticache modify-cache-subnet-group \  
  --cache-subnet-group-name kxkxk \  
  --cache-subnet-group-description "mygroup"
```

出力:

```
{  
  "CacheSubnetGroup": {  
    "CacheSubnetGroupName": "kxkxk",  
    "CacheSubnetGroupDescription": "mygroup",  
    "VpcId": "vpc-xxxxcdb",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-xxxxbff",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      }  
    ]  
  }  
}
```

詳細については、Elasticache ユーザーガイドの [「サブネットグループの変更」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyCacheSubnetGroup](#)」の「」を参照してください。

modify-global-replication-group

次のコード例は、modify-global-replication-group を使用する方法を示しています。

AWS CLI

グローバルレプリケーショングループを変更するには

以下は、Redis エンジンを使用して、グローバルレプリケーショングループのプロパティ modify-global-replication-group を変更します。この場合、自動フェイルオーバーを無効にします。

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id sgai-pat-group \  
  --apply-immediately \  
  --no-automatic-failover-enabled
```

出力

```
{  
  "GlobalReplicationGroup": {  
    "GlobalReplicationGroupId": "sgai-test-group",  
    "GlobalReplicationGroupDescription": " ",  
    "Status": "modifying",  
    "CacheNodeType": "cache.r5.large",  
    "Engine": "redis",  
    "EngineVersion": "5.0.6",  
    "ClusterEnabled": false,  
    "AuthTokenEnabled": false,  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

詳細については、ElastiCache ユーザーガイドの「[グローバルデータストアを使用した AWS リージョン間のレプリケーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyGlobalReplicationGroup](#)」の「」を参照してください。

modify-replication-group-shard-configuration

次のコード例は、`modify-replication-group-shard-configuration` を使用する方法を示しています。

AWS CLI

レプリケーショングループのシャード設定を変更するには

以下は、Redis エンジンを使用してノードグループの数 `modify-replication-group-shard-configuration` を減らします。

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id mycluster \  
  --node-group-count 3 \  
  --apply-immediately \  
  --node-groups-to-remove 0002
```

出力

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "mycluster",  
    "Description": "mycluster",  
    "GlobalReplicationGroupInfo": {},  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "mycluster-0002-001",  
      "mycluster-0002-002",  
      "mycluster-0002-003",  
      "mycluster-0003-001",  
      "mycluster-0003-002",  
      "mycluster-0003-003",  
      "mycluster-0003-004",  
      "mycluster-0004-001",  
      "mycluster-0004-002",  
      "mycluster-0004-003",  
      "mycluster-0005-001",  
      "mycluster-0005-002",  
      "mycluster-0005-003"  
    ],  
    "NodeGroups": [  

```

```
{
  "NodeGroupId": "0002",
  "Status": "modifying",
  "Slots": "894-1767,3134-4443,5149-5461,6827-7332,12570-13662",
  "NodeGroupMembers": [
    {
      "CacheClusterId": "mycluster-0002-001",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2c"
    },
    {
      "CacheClusterId": "mycluster-0002-002",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2a"
    },
    {
      "CacheClusterId": "mycluster-0002-003",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2b"
    }
  ]
},
{
  "NodeGroupId": "0003",
  "Status": "modifying",
  "Slots":
"0-324,5462-5692,6784-6826,7698-8191,10923-11075,12441-12569,13663-16383",
  "NodeGroupMembers": [
    {
      "CacheClusterId": "mycluster-0003-001",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2c"
    },
    {
      "CacheClusterId": "mycluster-0003-002",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2b"
    },
    {
      "CacheClusterId": "mycluster-0003-003",
      "CacheNodeId": "0001",
      "PreferredAvailabilityZone": "us-west-2a"
    }
  ]
}
```

```
        "CacheClusterId": "mycluster-0003-004",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2c"
    }
]
},
{
    "NodeGroupId": "0004",
    "Status": "modifying",
    "Slots": "325-336,4706-5148,7333-7697,9012-10922,11076-12440",
    "NodeGroupMembers": [
        {
            "CacheClusterId": "mycluster-0004-001",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2b"
        },
        {
            "CacheClusterId": "mycluster-0004-002",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2a"
        },
        {
            "CacheClusterId": "mycluster-0004-003",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2c"
        }
    ]
},
{
    "NodeGroupId": "0005",
    "Status": "modifying",
    "Slots": "337-893,1768-3133,4444-4705,5693-6783,8192-9011",
    "NodeGroupMembers": [
        {
            "CacheClusterId": "mycluster-0005-001",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2a"
        },
        {
            "CacheClusterId": "mycluster-0005-002",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2c"
        }
    ]
}
```

```
        "CacheClusterId": "mycluster-0005-003",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2b"
    }
]
},
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"ConfigurationEndpoint": {
    "Address": "mycluster.g2xbih.clustercfg.usw2.cache.amazonaws.com",
    "Port": 6379
},
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "13:00-14:00",
"ClusterEnabled": true,
"CacheNodeType": "cache.r5.xlarge",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}
```

詳細については、ElastiCache ユーザーガイドの「[Redis クラスター ElastiCache のスケーリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyReplicationGroupShardConfiguration](#)」の「」を参照してください。

modify-replication-group

次のコード例は、modify-replication-group を使用する方法を示しています。

AWS CLI

レプリケーショングループを変更するには

以下は、Redis エンジンを使用してマルチ AZ をmodify-replication-group無効にします。

```
aws elasticache modify-replication-group \  
  --replication-group-id test-cluster \  
  --no-multi-az-enabled \  
  --apply-immediately
```


出力

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "test-cluster",
    "Description": "test-cluster",
    "GlobalReplicationGroupInfo": {
      "GlobalReplicationGroupId": "sgaui-pat-group",
      "GlobalReplicationGroupMemberRole": "PRIMARY"
    },
    "Status": "available",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "test-cluster-001",
      "test-cluster-002",
      "test-cluster-003"
    ],
    "NodeGroups": [
      {
        "NodeGroupId": "0001",
        "Status": "available",
        "PrimaryEndpoint": {
          "Address": "test-
cluster.g2xbih.ng.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "ReaderEndpoint": {
          "Address": "test-cluster-
ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "NodeGroupMembers": [
          {
            "CacheClusterId": "test-cluster-001",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
              "Address": "test-
cluster-001.g2xbih.0001.usw2.cache.amazonaws.com",
              "Port": 6379
            },
            "PreferredAvailabilityZone": "us-west-2c",
            "CurrentRole": "primary"
          },
          {
```

```

        "CacheClusterId": "test-cluster-002",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "test-
cluster-002.g2xbih.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    },
    {
        "CacheClusterId": "test-cluster-003",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "test-
cluster-003.g2xbih.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "replica"
    }
]
}
],
"SnapshottingClusterId": "test-cluster-002",
"AutomaticFailover": "enabled",
"MultiAZ": "disabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "08:00-09:00",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}

```

詳細については、Elasticache ユーザーガイドの「[レプリケーショングループの変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyReplicationGroup](#)」の「」を参照してください。

modify-user-group

次の例は、modify-user-group を使用する方法を説明しています。

AWS CLI

ユーザーグループを変更するには

次のmodify-user-group例では、ユーザーをユーザーグループに追加します。

```
aws elasticache modify-user-group \  
  --user-group-id myusergroup \  
  --user-ids-to-add user1
```

出力:

```
{  
  "UserGroupId": "myusergroup",  
  "Status": "modifying",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "PendingChanges": {  
    "UserIdsToAdd": [  
      "user1"  
    ]  
  },  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"  
}
```

詳細については、Elasticache ユーザーガイドの [「ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyUserGroup](#)」の「」を参照してください。

modify-user

次の例は、modify-user を使用する方法を説明しています。

AWS CLI

ユーザーを変更するには

次のmodify-user例では、ユーザーのアクセス文字列を変更します。

```
aws elasticache modify-user \  
  --user-id user2 \  
  --append-access-string "on ~* +@all"
```

出力:

```
{  
  "UserId": "user2",  
  "UserName": "myUser",  
  "Status": "modifying",  
  "Engine": "redis",  
  "AccessString": "on ~* +@all",  
  "UserGroupIds": [],  
  "Authentication": {  
    "Type": "password",  
    "PasswordCount": 1  
  },  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx52:user:user2"  
}
```

詳細については、Elasticache ユーザーガイドの [「ロールベースのアクセスコントロール \(RBAC\) によるユーザーの認証」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyUser](#)」の「」を参照してください。

purchase-reserved-cache-nodes-offering

次のコード例は、purchase-reserved-cache-nodes-offering を使用する方法を示しています。

AWS CLI

を購入するには reserved-cache-node-offering

次のpurchase-reserved-cache-nodes-offering例では、リザーブドキャッシュノードサービスを購入できます。

```
aws elasticache purchase-reserved-cache-nodes-offering \  
  --reserved-cache-nodes-offering-id xxxxxxxx-4da5-4b90-b92d-929fbd7abed2
```

出力

```
{  
  "ReservedCacheNode": {  
    "ReservedCacheNodeId": "ri-2020-06-30-17-59-40-474",  
    "ReservedCacheNodesOfferingId": "xxxxxxx-4da5-4b90-b92d-929fbd7abed2",  
    "CacheNodeType": "cache.m3.2xlarge",  
    "StartTime": "2020-06-30T17:59:40.474000+00:00",  
    "Duration": 31536000,  
    "FixedPrice": 1772.0,  
    "UsagePrice": 0.0,  
    "CacheNodeCount": 1,  
    "ProductDescription": "redis",  
    "OfferingType": "Heavy Utilization",  
    "State": "payment-pending",  
    "RecurringCharges": [  
      {  
        "RecurringChargeAmount": 0.25,  
        "RecurringChargeFrequency": "Hourly"  
      }  
    ]  
  }  
}
```

詳細については、「Elasticache Redis [ユーザーガイド](#)」の「[リザーブドノードオフリングに関する情報の取得](#)」または「Elasticache Memcached [ユーザーガイド](#)」の「[リザーブドノードオフリングに関する情報の取得](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスPurchaseReservedCacheNodesOffering](#)」の「」を参照してください。

reboot-cache-cluster

次の例は、reboot-cache-cluster を使用する方法を説明しています。

AWS CLI

キャッシュクラスターを再起動するには

次のreboot-cache-cluster例では、プロビジョニングされたクラスター内のキャッシュノードの一部またはすべてを再起動します。このオペレーションは、変更されたキャッシュパラメータグループをクラスターに適用します。再起動オペレーションはできるだけ早く行われ、クラスターが一時的に停止します。再起動中、クラスターのステータスはに設定されま
すREBOOTING。

```
aws elasticache reboot-cache-cluster \  
  --cache-cluster-id "my-cluster-001" \  
  --cache-node-ids-to-reboot "0001"
```

出力:

```
{  
  "CacheCluster": {  
    "CacheClusterId": "my-cluster-001",  
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/  
home#client-download:",  
    "CacheNodeType": "cache.r5.xlarge",  
    "Engine": "redis",  
    "EngineVersion": "5.0.5",  
    "CacheClusterStatus": "rebooting cache cluster nodes",  
    "NumCacheNodes": 1,  
    "PreferredAvailabilityZone": "us-west-2a",  
    "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",  
    "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",  
    "PendingModifiedValues": {},  
    "NotificationConfiguration": {  
      "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxx152:My_Topic",  
      "TopicStatus": "active"  
    },  
    "CacheSecurityGroups": [],  
    "CacheParameterGroup": {  
      "CacheParameterGroupName": "mygroup",  
      "ParameterApplyStatus": "in-sync",  
      "CacheNodeIdsToReboot": []  
    },  
    "CacheSubnetGroupName": "kxkxk",  
    "AutoMinorVersionUpgrade": true,  
  },  
}
```

```
    "SecurityGroups": [  
      {  
        "SecurityGroupId": "sg-xxxxxxxxxxxx836",  
        "Status": "active"  
      },  
      {  
        "SecurityGroupId": "sg-xxxxxxx7b",  
        "Status": "active"  
      }  
    ],  
    "ReplicationGroupId": "my-cluster",  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "07:30-08:30",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

詳細については、Elasticache ユーザーガイドの「クラスターの再起動」 < <https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Rebooting.html> を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RebootCacheCluster](#)」の「」を参照してください。

reset-cache-parameter-group

次の例は、reset-cache-parameter-group を使用する方法を説明しています。

AWS CLI

キャッシュパラメータグループをリセットするには

次のreset-cache-parameter-group例では、キャッシュパラメータグループのパラメータをエンジンまたはシステムのデフォルト値に変更します。パラメータ名のリストを送信することで、特定のパラメータをリセットできます。キャッシュパラメータグループ全体をリセットするには、--reset-all-parameters および --cache-parameter-group-nameパラメータを指定します。

```
aws elasticache reset-cache-parameter-group \  
  --cache-parameter-group-name "mygroup" \  
  --reset-all-parameters
```

出力:

```
{
  "CacheParameterGroupName": "mygroup"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetCacheParameterGroup](#)」の「」を参照してください。

start-migration

次の例は、start-migration を使用する方法を説明しています。

AWS CLI

移行を開始するには

以下は、Redis エンジンを使用して ElastiCache、Amazon のセルフホスト Redis から Amazon EC2 にデータを start-migration 移行します。

```
aws elasticache start-migration \
  --replication-group-id test \
  --customer-node-endpoint-
  list "Address='test.g2xbih.ng.0001.usw2.cache.amazonaws.com',Port=6379"
```

出力

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "test",
    "Description": "test",
    "GlobalReplicationGroupInfo": {},
    "Status": "modifying",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "test-001",
      "test-002",
      "test-003"
    ],
    "NodeGroups": [
      {
        "NodeGroupId": "0001",
```



```
"Status": "available",
"PrimaryEndpoint": {
  "Address": "test.g2xbih.ng.0001.usw2.cache.amazonaws.com",
  "Port": 6379
},
"ReaderEndpoint": {
  "Address": "test-ro.g2xbih.ng.0001.usw2.cache.amazonaws.com",
  "Port": 6379
},
"NodeGroupMembers": [
  {
    "CacheClusterId": "test-001",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Address":
"test-001.g2xbih.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2a",
    "CurrentRole": "primary"
  },
  {
    "CacheClusterId": "test-002",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Address":
"test-002.g2xbih.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2c",
    "CurrentRole": "replica"
  },
  {
    "CacheClusterId": "test-003",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
      "Address":
"test-003.g2xbih.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2b",
    "CurrentRole": "replica"
  }
]
```

```
    }
  ],
  "SnapshottingClusterId": "test-002",
  "AutomaticFailover": "enabled",
  "MultiAZ": "enabled",
  "SnapshotRetentionLimit": 1,
  "SnapshotWindow": "07:30-08:30",
  "ClusterEnabled": false,
  "CacheNodeType": "cache.r5.large",
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false
}
}
```

詳細については、Elasticache ユーザーガイドの「[へのオンライン移行 ElastiCache](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartMigration](#)」の「」を参照してください。

test-failover

次のコード例は、test-failover を使用する方法を示しています。

AWS CLI

ノードグループのフェイルオーバーをテストするには

次のtest-failover例では、レプリケーショングループ (コンソールでクラスターと呼ばれる) 内の指定されたノードグループ (コンソールでシャードと呼ばれる) で自動フェイルオーバーをテストします。

```
aws elasticache test-failover /
  --replication-group-id "mycluster" /
  --node-group-id "0001"
```

出力:

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "mycluster",
    "Description": "My Cluster",
```

```
"Status": "available",
"PendingModifiedValues": {},
"MemberClusters": [
  "mycluster-0001-001",
  "mycluster-0001-002",
  "mycluster-0001-003",
  "mycluster-0002-001",
  "mycluster-0002-002",
  "mycluster-0002-003",
  "mycluster-0003-001",
  "mycluster-0003-002",
  "mycluster-0003-003"
],
"NodeGroups": [
  {
    "NodeGroupId": "0001",
    "Status": "available",
    "Slots": "0-5461",
    "NodeGroupMembers": [
      {
        "CacheClusterId": "mycluster-0001-001",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2b"
      },
      {
        "CacheClusterId": "mycluster-0001-002",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2a"
      },
      {
        "CacheClusterId": "mycluster-0001-003",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2c"
      }
    ]
  },
  {
    "NodeGroupId": "0002",
    "Status": "available",
    "Slots": "5462-10922",
    "NodeGroupMembers": [
      {
        "CacheClusterId": "mycluster-0002-001",
        "CacheNodeId": "0001",
```

```

        "PreferredAvailabilityZone": "us-west-2a"
    },
    {
        "CacheClusterId": "mycluster-0002-002",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2b"
    },
    {
        "CacheClusterId": "mycluster-0002-003",
        "CacheNodeId": "0001",
        "PreferredAvailabilityZone": "us-west-2c"
    }
]
},
{
    "NodeGroupId": "0003",
    "Status": "available",
    "Slots": "10923-16383",
    "NodeGroupMembers": [
        {
            "CacheClusterId": "mycluster-0003-001",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2c"
        },
        {
            "CacheClusterId": "mycluster-0003-002",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2b"
        },
        {
            "CacheClusterId": "mycluster-0003-003",
            "CacheNodeId": "0001",
            "PreferredAvailabilityZone": "us-west-2a"
        }
    ]
}
],
"AutomaticFailover": "enabled",
"ConfigurationEndpoint": {
    "Address": "mycluster.xxxxih.clustercfg.usw2.cache.amazonaws.com",
    "Port": 6379
},
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "13:00-14:00",

```

```
    "ClusterEnabled": true,  
    "CacheNodeType": "cache.r5.large",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[TestFailover](#)」の「」を参照してください。

MediaStore を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています MediaStore。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-container

次のコード例は、create-container を使用する方法を示しています。

AWS CLI

コンテナを作成するには

次のcreate-container例では、新しい空のコンテナを作成します。

```
aws mediastore create-container --container-name ExampleContainer
```

出力:

```
{
  "Container": {
    "AccessLoggingEnabled": false,
    "CreationTime": 1563557265,
    "Name": "ExampleContainer",
    "Status": "CREATING",
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
  }
}
```

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「コンテナの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateContainer](#)」の「」を参照してください。

delete-container-policy

次の例は、delete-container-policy を使用する方法を説明しています。

AWS CLI

コンテナポリシーを削除するには

次のdelete-container-policy例では、指定されたコンテナに割り当てられたポリシーを削除します。ポリシーが削除されると、AWS Elemental MediaStore は自動的にデフォルトのポリシーをコンテナに割り当てます。

```
aws mediastore delete-container-policy \
  --container-name LiveEvents
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore API リファレンス[DeleteContainerPolicy](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteContainerPolicy](#)」の「」を参照してください。

delete-container

次のコード例は、delete-container を使用する方法を示しています。

AWS CLI

コンテナを削除するには

次のdelete-container例では、指定されたコンテナを削除します。コンテナにオブジェクトが含まれていない場合に限り、コンテナを削除できます。

```
aws mediastore delete-container \  
  --container-name=ExampleLiveDemo
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「コンテナの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteContainer](#)」の「」を参照してください。

delete-cors-policy

次のコード例は、delete-cors-policy を使用する方法を示しています。

AWS CLI

CORSポリシーを削除するには

次のdelete-cors-policy例では、指定されたコンテナに割り当てられたクロスオリジンリソース共有 (CORS) ポリシーを削除します。

```
aws mediastore delete-cors-policy \  
  --container-name ExampleContainer
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[CORS 「ポリシーの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCorsPolicy](#)」の「」を参照してください。

delete-lifecycle-policy

次の例は、delete-lifecycle-policy を使用する方法を説明しています。

AWS CLI

オブジェクトライフサイクルポリシーを削除するには

次のdelete-lifecycle-policy例では、指定されたコンテナにアタッチされたオブジェクトライフサイクルポリシーを削除します。この変更が有効になるまでに最大 20 分かかる場合があります。

```
aws mediastore delete-lifecycle-policy \  
  --container-name LiveEvents
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の「[オブジェクトライフサイクルポリシーの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLifecyclePolicy](#)」の「」を参照してください。

describe-container

次の例は、describe-container を使用する方法を説明しています。

AWS CLI

コンテナの詳細を表示するには

次のdescribe-container例では、指定されたコンテナの詳細を表示します。

```
aws mediastore describe-container \  
  --container-name ExampleContainer
```

出力:


```
{
  "Container": {
    "CreationTime": 1563558086,
    "AccessLoggingEnabled": false,
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
    "Status": "ACTIVE",
    "Name": "ExampleContainer",
    "Endpoint": "https://aaabbbcccddee.data.mediastore.us-west-2.amazonaws.com"
  }
}
```

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の「[コンテナの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeContainer](#)」の「」を参照してください。

describe-object

次の例は、describe-object を使用する方法を説明しています。

AWS CLI

特定のコンテナ内のオブジェクトとフォルダのリストを表示するには

次のdescribe-object例では、特定のコンテナに保存されている項目 (オブジェクトとフォルダ) を表示します。

```
aws mediastore-data describe-object \
  --endpoint https://aaabbbcccddee.data.mediastore.us-west-2.amazonaws.com \
  --path /folder_name/file1234.jpg
```

出力:

```
{
  "ContentType": "image/jpeg",
  "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
  "ContentLength": "2307346",
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e9999994dd89ff7f55555555555555da6d3"
}
```

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の「[オブジェクトの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeObject](#)」の「」を参照してください。

get-container-policy

次の例は、get-container-policy を使用する方法を説明しています。

AWS CLI

コンテナポリシーを表示するには

次のget-container-policy例では、指定されたコンテナのリソースベースのポリシーを表示します。

```
aws mediastore get-container-policy \  
  --container-name ExampleLiveDemo
```

出力:

```
{  
  "Policy": {  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Sid": "PublicReadOverHttps",  
        "Effect": "Allow",  
        "Principal": {  
          "AWS": "arn:aws:iam::111122223333:root"  
        },  
        "Action": [  
          "mediastore:GetObject",  
          "mediastore:DescribeObject"  
        ],  
        "Resource": "arn:aws:mediastore:us-west-2:111122223333:container/  
ExampleLiveDemo/",  
        "Condition": {  
          "Bool": {  
            "aws:SecureTransport": "true"  
          }  
        }  
      }  
    ]  
  }  
}
```

```
}
  }
]
}
```

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の「[コンテナポリシーの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetContainerPolicy](#)」の「」を参照してください。

get-cors-policy

次の例は、get-cors-policy を使用する方法を説明しています。

AWS CLI

CORSポリシーを表示するには

次のget-cors-policy例では、指定されたコンテナに割り当てられたクロスオリジンリソース共有 (CORS) ポリシーを表示します。

```
aws mediastore get-cors-policy \
  --container-name ExampleContainer \
  --region us-west-2
```

出力:

```
{
  "CorsPolicy": [
    {
      "AllowedMethods": [
        "GET",
        "HEAD"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedOrigins": [
        ""
      ],
      "AllowedHeaders": [
```

```

        ""
    ]
}
}
}

```

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の[CORS「ポリシーの表示」](#)を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetCorsPolicy](#)」の「」を参照してください。

get-lifecycle-policy

次の例は、get-lifecycle-policy を使用する方法を説明しています。

AWS CLI

オブジェクトライフサイクルポリシーを表示するには

次のget-lifecycle-policy例は、指定されたコンテナにアタッチされたオブジェクトライフサイクルポリシーを示しています。

```
aws mediastore get-lifecycle-policy \
  --container-name LiveEvents
```

出力:

```
{
  "LifecyclePolicy": {
    "rules": [
      {
        "definition": {
          "path": [
            {
              "prefix": "Football/"
            },
            {
              "prefix": "Baseball/"
            }
          ]
        }
      }
    ]
  }
}
```

```

    "days_since_create": [
      {
        "numeric": [
          ">",
          28
        ]
      }
    ],
    "action": "EXPIRE"
  }
]
}

```

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の「[オブジェクトライフサイクルポリシーの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLifecyclePolicy](#)」の「」を参照してください。

get-object

次の例は、get-object を使用する方法を説明しています。

AWS CLI

オブジェクトをダウンロードするには

次のget-object例では、指定されたエンドポイントにオブジェクトをダウンロードします。

```

aws mediastore-data get-object \
  --endpoint https://aaabbbcccddee.data.mediastore.us-west-2.amazonaws.com \
  --path=/folder_name/README.md README.md

```

出力:

```

{
  "ContentLength": "2307346",
  "ContentType": "image/jpeg",
  "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f5555555555555555da6d3",

```

```
"statusCode": 200
}
```

オブジェクトの一部をダウンロードするには

次のget-object例では、指定されたエンドポイントにオブジェクトの一部をダウンロードします。

```
aws mediastore-data get-object \
  --endpoint https://aaabbbcccddee.data.mediastore.us-west-2.amazonaws.com \
  --path /folder_name/README.md \
  --range="bytes=0-100" README2.md
```

出力:

```
{
  "statusCode": 206,
  "ContentRange": "bytes 0-100/2307346",
  "ContentLength": "101",
  "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
  "ContentType": "image/jpeg",
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eEEEEEE4dd89ff7f5555555555555555da6d3"
}
```

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「オブジェクトのダウンロード」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetObject](#)」の「」を参照してください。

list-containers

次のコード例は、list-containers を使用する方法を示しています。

AWS CLI

コンテナのリストを表示するには

次のlist-containers例では、アカウントに関連付けられているすべてのコンテナのリストを表示します。

aws mediastore list-containers

出力:

```
{
  "Containers": [
    {
      "CreationTime": 1505317931,
      "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-
west-2.amazonaws.com",
      "Status": "ACTIVE",
      "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleLiveDemo",
      "AccessLoggingEnabled": false,
      "Name": "ExampleLiveDemo"
    },
    {
      "CreationTime": 1506528818,
      "Endpoint": "https://ffffggghhhiiijj.data.mediastore.us-
west-2.amazonaws.com",
      "Status": "ACTIVE",
      "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
      "AccessLoggingEnabled": false,
      "Name": "ExampleContainer"
    }
  ]
}
```

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の「[コンテナのリストの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListContainers](#)」の「」を参照してください。

list-items

次の例は、list-items を使用する方法を説明しています。

AWS CLI

例 1: 特定のコンテナ内のオブジェクトとフォルダのリストを表示するには

次のlist-items例では、指定されたコンテナに保存されている項目 (オブジェクトとフォルダ) を表示します。

```
aws mediastore-data list-items \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com
```

出力:

```
{  
  "Items": [  
    {  
      "ContentType": "image/jpeg",  
      "LastModified": 1563571859.379,  
      "Name": "filename.jpg",  
      "Type": "OBJECT",  
      "ETag":  
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",  
      "ContentLength": 3784  
    },  
    {  
      "Type": "FOLDER",  
      "Name": "ExampleLiveDemo"  
    }  
  ]  
}
```

例 2: 特定のフォルダ内のオブジェクトとフォルダのリストを表示するには

次のlist-items例では、特定のフォルダに保存されている項目 (オブジェクトとフォルダ) を表示します。

```
aws mediastore-data list-items \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com
```

出力:

```
{  
  "Items": [  
    {  
      "ContentType": "image/jpeg",  
      "LastModified": 1563571859.379,  
      "Name": "filename.jpg",
```



```
    "Type": "OBJECT",
    "ETag":
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
    "ContentLength": 3784
  },
  {
    "Type": "FOLDER",
    "Name": "ExampleLiveDemo"
  }
]
```

詳細については、「AWS Elemental MediaStore ユーザーガイド」の [「オブジェクトのリストの表示」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListItems](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

コンテナのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたコンテナに割り当てられたタグキーと値を表示します。

```
aws mediastore list-tags-for-resource \  
  --resource arn:aws:mediastore:us-west-2:1213456789012:container/ExampleContainer
```

出力:

```
{  
  "Tags": [  
    {  
      "Value": "Test",  
      "Key": "Environment"  
    },  
    {  
      "Value": "West",
```

```
        "Key": "Region"
      }
    ]
  }
```

詳細については、「AWS Elemental MediaStore API Reference [ListTagsForResource](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-container-policy

次の例は、put-container-policy を使用する方法を説明しています。

AWS CLI

コンテナポリシーを編集するには

次のput-container-policy例では、指定されたコンテナに別のポリシーを割り当てます。この例では、更新されたポリシーは `LiveEventsContainerPolicy.json` という名前のファイルで定義されます。

```
aws mediastore put-container-policy \
  --container-name LiveEvents \
  --policy file://LiveEventsContainerPolicy.json
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の「[コンテナポリシーの編集](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutContainerPolicy](#)」の「」を参照してください。

put-cors-policy

次の例は、put-cors-policy を使用する方法を説明しています。

AWS CLI

例 1: CORSポリシーを追加するには

次のput-cors-policy例では、指定したコンテナにクロスオリジンリソース共有 (CORS) ポリシーを追加します。CORS ポリシーの内容は、という名前のファイルにありますcorsPolicy.json。

```
aws mediastore put-cors-policy \  
  --container-name ExampleContainer \  
  --cors-policy file://corsPolicy.json
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「コンテナへのCORSポリシーの追加」](#)を参照してください。

例 2: CORSポリシーを編集するには

次のput-cors-policy例では、指定されたコンテナに割り当てられたクロスオリジンリソース共有 (CORS) ポリシーを更新します。更新されたCORSポリシーの内容は、という名前のファイルにありますcorsPolicy2.json。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[CORS「ポリシーの編集」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutCorsPolicy](#)」の「」を参照してください。

put-lifecycle-policy

次の例は、put-lifecycle-policy を使用する方法を説明しています。

AWS CLI

オブジェクトライフサイクルポリシーを作成するには

次のput-lifecycle-policy例では、指定されたコンテナにオブジェクトライフサイクルポリシーをアタッチします。これにより、サービスがコンテナにオブジェクトを保存する期間を指定できます。は、という名前のファイルにあるポリシーに示すように、有効期限に達するとコンテナ内のオブジェクト MediaStore を削除しますLiveEventsLifecyclePolicy.json。

```
aws mediastore put-lifecycle-policy \  
  --container-name ExampleContainer \  
  --lifecycle-policy file://ExampleLifecyclePolicy.json
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「コンテナへのオブジェクトライフサイクルポリシーの追加」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLifecyclePolicy](#)」の「」を参照してください。

put-object

次の例は、put-object を使用する方法を説明しています。

AWS CLI

オブジェクトをアップロードするには

次のput-object例では、指定されたコンテナにオブジェクトをアップロードします。オブジェクトがコンテナ内に保存されるフォルダパスを指定できます。フォルダがすでに存在する場合、AWS Elemental はオブジェクトをフォルダに MediaStore 保存します。フォルダが存在しない場合、サービスはフォルダを作成し、オブジェクトをフォルダに保存します。

```
aws mediastore-data put-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --body README.md \  
  --path /folder_name/README.md \  
  --cache-control "max-age=6, public" \  
  --content-type binary/octet-stream
```

出力:

```
{  
  "ContentSHA256":  
    "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",  
  "StorageClass": "TEMPORAL",  
  "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"  
}
```

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「オブジェクトのアップロード」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutObject](#)」の「」を参照してください。

start-access-logging

次の例は、start-access-logging を使用する方法を説明しています。

AWS CLI

コンテナでアクセスログ記録を有効にするには

次のstart-access-logging例では、指定されたコンテナでアクセスログ記録を有効にします。

```
aws mediastore start-access-logging \  
  --container-name LiveEvents
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「コンテナのアクセスログ記録の有効化」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartAccessLogging](#)」の「」を参照してください。

stop-access-logging

次のコード例は、stop-access-logging を使用する方法を示しています。

AWS CLI

コンテナのアクセスログ記録を無効にするには

次のstop-access-logging例では、指定されたコンテナのアクセスログ記録を無効にします。

```
aws mediastore stop-access-logging \  
  --container-name LiveEvents
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「コンテナのアクセスログ記録の無効化」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopAccessLogging](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

コンテナにタグを追加するには

次のtag-resource例では、指定されたコンテナにタグキーと値を追加します。

```
aws mediastore tag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tags '[{"Key": "Region", "Value": "West"}, {"Key": "Environment", "Value": "Test"}]'
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaStore API Reference [TagResource](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

コンテナからタグを削除するには

次のuntag-resource例では、指定されたタグキーとそれに関連付けられた値をコンテナから削除します。

```
aws mediastore untag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tag-keys Region
```

このコマンドでは何も出力されません。

詳細については、「Elemental Reference [UntagResource](#)」の「」を参照してください。AWS MediaStore API

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

を使用した Amazon EMR の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますEMR。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-instance-fleet

次の例は、add-instance-fleet を使用する方法を説明しています。

AWS CLI

タスクインスタンスフリートをクラスターに追加するには

この例では、指定されたクラスターに新しいタスクインスタンスフリートを追加します。

コマンド:

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet InstanceFleetType=TASK,TargetSpotCapacity=1,LaunchSpecifications={SpotSpecification={Timeo
```

出力:

```
{
  "ClusterId": "j-12ABCDEFGH134JK",
  "InstanceFleetId": "if-23ABCDEFGH145JJ"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddInstanceFleet](#)」の「」を参照してください。

add-steps

次のコード例は、add-steps を使用する方法を示しています。

AWS CLI

1. クラスターにカスタムJARステップを追加するには

コマンド:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps
  Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://mybucket/
mytest.jar,Args=arg1,arg2,arg3
  Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://mybucket/
mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3
```

必要なパラメータ :

Jar

任意指定のパラメータ:

Type, Name, ActionOnFailure, Args

出力:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```


2. ストリーミングステップをクラスターに追加するには

コマンド:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=STREAMING,Name='Streaming Program',ActionOnFailure=CONTINUE,Args=[-files,s3://elasticmapreduce/samples/wordcount/wordSplitter.py,-mapper,wordSplitter.py,-reducer,aggregate,-input,s3://elasticmapreduce/samples/wordcount/input,-output,s3://mybucket/wordcount/output]
```

必要なパラメータ :

Type, Args

任意指定のパラメータ:

Name, ActionOnFailure

JSON 同等 (Step.json の内容):

```
[
  {
    "Name": "JSON Streaming Step",
    "Args": ["-files","s3://elasticmapreduce/samples/wordcount/wordSplitter.py",-mapper,"wordSplitter.py",-reducer,"aggregate",-input,"s3://elasticmapreduce/samples/wordcount/input",-output,"s3://mybucket/wordcount/output"],
    "ActionOnFailure": "CONTINUE",
    "Type": "STREAMING"
  }
]
```

NOTE: JSON引数には、リスト内の独自の項目としてオプションと値を含める必要があります。

コマンド (step.json を使用):

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file://./step.json
```

出力:

```
{
  "StepIds": [
```

```
        "s-XXXXXXXX",  
        "s-YYYYYYYY"  
    ]  
}
```

3. 複数のファイルを含むストリーミングステップをクラスターに追加するには (JSON のみ)

JSON (multiplefiles.json):

```
[  
  {  
    "Name": "JSON Streaming Step",  
    "Type": "STREAMING",  
    "ActionOnFailure": "CONTINUE",  
    "Args": [  
      "-files",  
      "s3://mybucket/mapper.py,s3://mybucket/reducer.py",  
      "-mapper",  
      "mapper.py",  
      "-reducer",  
      "reducer.py",  
      "-input",  
      "s3://mybucket/input",  
      "-output",  
      "s3://mybucket/output"]  
  }  
]
```

コマンド:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file://./multiplefiles.json
```

必要なパラメータ :

Type, Args

任意指定のパラメータ:

Name, ActionOnFailure

出力:

```
{
  "StepIds":[
    "s-XXXXXXXX",
  ]
}
```

4. クラスターに Hive ステップを追加するには

コマンド:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=HIVE,Name='Hive
program',ActionOnFailure=CONTINUE,Args=[-f,s3://mybucket/myhivescript.q,-
d,INPUT=s3://mybucket/myhiveinput,-d,OUTPUT=s3://mybucket/myhiveoutput,arg1,arg2]
Type=HIVE,Name='Hive steps',ActionOnFailure=TERMINATE_CLUSTER,Args=[-
f,s3://elasticmapreduce/samples/hive-ads/libs/model-build.q,-d,INPUT=s3://
elasticmapreduce/samples/hive-ads/tables,-d,OUTPUT=s3://mybucket/hive-ads/
output/2014-04-18/11-07-32,-d,LIBS=s3://elasticmapreduce/samples/hive-ads/libs]
```

必要なパラメータ:

Type, Args

任意指定のパラメータ:

Name, ActionOnFailure

出力:

```
{
  "StepIds":[
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

5. クラスターに Pig ステップを追加するには

コマンド:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=PIG,Name='Pig
program',ActionOnFailure=CONTINUE,Args=[-f,s3://mybucket/mypigscript.pig,-
```

```
p, INPUT=s3://mybucket/mypiginput, -p, OUTPUT=s3://mybucket/mypigoutput, arg1, arg2]
Type=PIG, Name='Pig program', Args=[-f, s3://elasticmapreduce/samples/pig-apache/do-
reports2.pig, -p, INPUT=s3://elasticmapreduce/samples/pig-apache/input, -p, OUTPUT=s3://
mybucket/pig-apache/output, arg1, arg2]
```

必要なパラメータ :

Type, Args

任意指定のパラメータ:

Name, ActionOnFailure

出力:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

6. Impala ステップをクラスターに追加するには

コマンド:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=IMPALA, Name='Impala
program', ActionOnFailure=CONTINUE, Args=--impala-script, s3://myimpala/input, --
console-output-path, s3://myimpala/output
```

必要なパラメータ :

Type, Args

任意指定のパラメータ:

Name, ActionOnFailure

出力:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AddSteps](#)」の「」を参照してください。

add-tags

次の例は、add-tags を使用する方法を説明しています。

AWS CLI

1. クラスターにタグを追加するには

コマンド:

```
aws emr add-tags --resource-id j-xxxxxxx --tags name="John Doe" age=29 sex=male
address="123 East NW Seattle"
```

出力:

```
None
```

2. クラスターのタグを一覧表示するには

-- コマンド :

```
aws emr describe-cluster --cluster-id j-XXXXXXYY --query Cluster.Tags
```

出力:

```
[
  {
    "Value": "male",
    "Key": "sex"
  },
  {
```

```
[
  {
    "Value": "123 East NW Seattle",
    "Key": "address"
  },
  {
    "Value": "John Doe",
    "Key": "name"
  },
  {
    "Value": "29",
    "Key": "age"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AddTags](#)」の「」を参照してください。

create-cluster-examples

次のコード例は、create-cluster-examples を使用する方法を示しています。

AWS CLI

以下の例のほとんどは、Amazon EMRサービスロールと Amazon EC2インスタンスプロファイルを指定していることを前提としています。これを行っていない場合は、必要な各IAMロールを指定するか、クラスターの作成時に --use-default-rolesパラメータを使用する必要があります。IAM ロールの指定の詳細については、[「Amazon 管理ガイド」の AWS 「サービスに対する Amazon EMR アクセス許可のIAMロールの設定」](#)を参照してください。 EMR

例 1: クラスターを作成するには

次のcreate-cluster例では、シンプルなEMRクラスターを作成します。

```
aws emr create-cluster \
  --release-label emr-5.14.0 \
  --instance-type m4.large \
  --instance-count 2
```

このコマンドでは何も出力されません。

例 2: デフォルト ServiceRole と InstanceProfile ロールを使用して Amazon EMRクラスターを作成するには

次のcreate-cluster例では、--instance-groups設定を使用する Amazon EMRクラスターを作成します。

```
aws emr create-cluster \  
  --release-label emr-5.14.0 \  
  --service-role EMR_DefaultRole \  
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

例 3: インスタンスフリートを使用する Amazon EMRクラスターを作成するには

次のcreate-cluster例では、--instance-fleets設定を使用する Amazon EMRクラスターを作成し、フリートごとに2つのインスタンスタイプと2つのEC2サブネットを指定します。

```
aws emr create-cluster \  
  --release-label emr-5.14.0 \  
  --service-role EMR_DefaultRole \  
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-ab12345c','subnet-de67890f'] \  
  --instance-fleets  
InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m4.large,TargetCapacity=1}  
InstanceFleetType=CORE,TargetSpotCapacity=11,InstanceTypeConfigs=['{InstanceType=m4.large,TargetCapacity=11}
```

例 4: デフォルトロールを使用してクラスターを作成するには

次のcreate-cluster例では、--use-default-rolesパラメータを使用して、デフォルトのサービスロールとインスタンスプロファイルを指定します。

```
aws emr create-cluster \  
  --release-label emr-5.9.0 \  
  --use-default-roles \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE  
 \  
  --auto-terminate
```

例 5: クラスターを作成し、インストールするアプリケーションを指定するには

次のcreate-cluster例では、--applicationsパラメータを使用して、Amazon がEMRインストールするアプリケーションを指定します。この例では、Hadoop、Hive、Pig をインストールします。

```
aws emr create-cluster \  
  --applications Name=Hadoop Name=Hive Name=Pig \  
  --release-label emr-5.9.0 \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE  
 \  
  --auto-terminate
```

例 6: Spark を含むクラスターを作成するには

次の例では、Spark をインストールします。

```
aws emr create-cluster \  
  --release-label emr-5.9.0 \  
  --applications Name=Spark \  
  --ec2-attributes KeyName=myKey \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE  
 \  
  --auto-terminate
```

例 7: クラスターインスタンスAMIに使用するカスタムを指定するには

次のcreate-cluster例では、ID AMIを持つ Amazon Linux に基づいてクラスターインスタンスを作成しますami-a518e6df。

```
aws emr create-cluster \  
  --name "Cluster with My Custom AMI" \  
  --custom-ami-id ami-a518e6df \  
  --ebs-root-volume-size 20 \  
  --release-label emr-5.9.0 \  
  --use-default-roles \  
  --instance-count 2 \  
  --instance-type m4.Large
```

例 8: アプリケーション設定をカスタマイズするには

次の例では、--configurationsパラメータを使用して、Hadoop のアプリケーションのカスタマイズを含むJSON設定ファイルを指定します。詳細については、「Amazon リリースガイド」の「[アプリケーションの設定](#)」を参照してください。 EMR

configurations.json の内容:

```
[
  {
    "Classification": "mapred-site",
    "Properties": {
      "mapred.tasktracker.map.tasks.maximum": 2
    }
  },
  {
    "Classification": "hadoop-env",
    "Properties": {},
    "Configurations": [
      {
        "Classification": "export",
        "Properties": {
          "HADOOP_DATANODE_HEAPSIZE": 2048,
          "HADOOP_NAMENODE_OPTS": "-XX:GCTimeRatio=19"
        }
      }
    ]
  }
]
```

次の例では、をローカルファイルconfigurations.jsonとして参照しています。

```
aws emr create-cluster \
  --configurations file://configurations.json \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
  \
  --auto-terminate
```

次の例では、Amazon S3 でをファイルconfigurations.jsonとして参照しています。

```
aws emr create-cluster \
  --configurations https://s3.amazonaws.com/myBucket/configurations.json \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
  \
```

--auto-terminate

例 9: マスターインスタンスグループ、コアインスタンスグループ、タスクインスタンスグループを使用してクラスターを作成するには

次のcreate-cluster例では、--instance-groupsを使用して、マスターEC2、コア、タスクインスタンスグループに使用するインスタンスのタイプと数を指定します。

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --instance-
groups Name=Master,InstanceGroupType=MASTER,InstanceType=m4.Large,InstanceCount=1 Name=Core,
```

例 10: すべてのステップを完了した後にクラスターを終了するように指定するには

次のcreate-cluster例では--auto-terminate、を使用して、すべてのステップを完了した後にクラスターを自動的にシャットダウンするように指定します。

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large
InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.Large \
  --auto-terminate
```

例 11: Amazon EC2キーペア、ネットワーク設定、セキュリティグループなどのクラスター設定の詳細を指定するには

次のcreate-cluster例では、[という名前の Amazon EC2キーペアmyKeyと という名前のカスタマイズされたインスタンスプロファイルを持つクラスターを作成しますmyProfile](#)。キーペアは、クラスターノード、ほとんどの場合マスターノードSSHへの接続を承認するために使用します。詳細については、[「Amazon 管理ガイド」のSSH「認証情報に Amazon EC2キーペアを使用する」](#)を参照してください。 EMR

```
aws emr create-cluster \
  --ec2-attributes KeyName=myKey,InstanceProfile=myProfile \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE
\
  --auto-terminate
```

次の例では、Amazon VPCサブネットにクラスターを作成します。

```
aws emr create-cluster \  
  --ec2-attributes SubnetId=subnet-xxxxx \  
  --release-label emr-5.9.0 \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE  
 \  
  --auto-terminate
```

次の例では、us-east-1bアベイラビリティーゾーンにクラスターを作成します。

```
aws emr create-cluster \  
  --ec2-attributes AvailabilityZone=us-east-1b \  
  --release-label emr-5.9.0 \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

次の例では、クラスターを作成し、Amazon EMRが管理するセキュリティグループのみを指定します。

```
aws emr create-cluster \  
  --release-label emr-5.9.0 \  
  --service-role myServiceRole \  
  --ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-  
master1,EmrManagedSlaveSecurityGroup=sg-slave1 \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

次の例では、クラスターを作成し、追加の Amazon EC2 セキュリティグループのみを指定します。

```
aws emr create-cluster \  
  --release-label emr-5.9.0 \  
  --service-role myServiceRole \  
  --ec2-attributes InstanceProfile=myRole,AdditionalMasterSecurityGroups=[sg-  
addMaster1,sg-addMaster2,sg-addMaster3,sg-  
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-  
addSlave3,sg-addSlave4] \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

次の例では、クラスターを作成し、EMRマネージドセキュリティグループと追加のセキュリティグループを指定します。

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --service-role myServiceRole \
  --ec2-attributes InstanceProfile=myRole,EmrManagedMasterSecurityGroup=sg-
master1,EmrManagedSlaveSecurityGroup=sg-slave1,AdditionalMasterSecurityGroups=[sg-
addMaster1,sg-addMaster2,sg-addMaster3,sg-
addMaster4],AdditionalSlaveSecurityGroups=[sg-addSlave1,sg-addSlave2,sg-
addSlave3,sg-addSlave4] \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

次の例では、VPCプライベートサブネットにクラスターを作成し、特定の Amazon EC2 セキュリティグループを使用して Amazon EMR サービスアクセスを有効にします。これは、プライベートサブネットのクラスターに必要です。

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --service-role myServiceRole \
  --ec2-attributes InstanceProfile=myRole,ServiceAccessSecurityGroup=sg-service-
access,EmrManagedMasterSecurityGroup=sg-master,EmrManagedSlaveSecurityGroup=sg-slave
\
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

次の例では、ローカルに保存 `ec2_attributes.json` されている という名前の JSON ファイルを使用して、セキュリティグループ設定パラメータを指定します。NOTE: JSON 引数には、リスト内の独自の項目としてオプションと値を含める必要があります。

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --service-role myServiceRole \
  --ec2-attributes file://ec2_attributes.json \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

`ec2_attributes.json` の内容:

```
[
```

```
{
  "SubnetId": "subnet-xxxxx",
  "KeyName": "myKey",
  "InstanceProfile": "myRole",
  "EmrManagedMasterSecurityGroup": "sg-master1",
  "EmrManagedSlaveSecurityGroup": "sg-slave1",
  "ServiceAccessSecurityGroup": "sg-service-access",
  "AdditionalMasterSecurityGroups": ["sg-addMaster1", "sg-addMaster2", "sg-addMaster3", "sg-addMaster4"],
  "AdditionalSlaveSecurityGroups": ["sg-addSlave1", "sg-addSlave2", "sg-addSlave3", "sg-addSlave4"]
}
```

例 12: デバッグを有効にしてログを指定するには URI

次の `create-cluster` 例では、`--enable-debugging` パラメータを使用しており、Amazon EMR コンソールのデバッグツールを使用してログファイルをより簡単に表示できます。`--log-uri` パラメータは必要です `--enable-debugging`。

```
aws emr create-cluster \
  --enable-debugging \
  --log-uri s3://myBucket/myLog \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate
```

例 13: クラスターの作成時にタグを追加するには

タグは、クラスターの特定と管理に役立つキーと値のペアです。次の `create-cluster` 例では、`--tags` パラメータを使用してクラスターに 3 つのタグを作成します。1 つはキー名 `name` と値 `Shirley Rodriguez`、もう 1 つはキー名 `age` と値 `29`、3 つ目のタグはキー名 `department` と値 `Analytics` です。

```
aws emr create-cluster \
  --tags name="Shirley Rodriguez" age=29 department="Analytics" \
  --release-label emr-5.32.0 \
  --instance-type m5.xlarge \
  --instance-count 3 \
```

```
--use-default-roles
```

次の例では、クラスターに適用されるタグを一覧表示します。

```
aws emr describe-cluster \
  --cluster-id j-XXXXXXYY \
  --query Cluster.Tags
```

例 14: 暗号化やその他のセキュリティ機能を有効にするセキュリティ設定を使用するには

次のcreate-cluster例では、--security-configurationパラメータを使用してEMRクラスターのセキュリティ設定を指定します。Amazon EMRバージョン 4.8.0 以降では、セキュリティ設定を使用できます。

```
aws emr create-cluster \
  --instance-type m4.large \
  --release-label emr-5.9.0 \
  --security-configuration mySecurityConfiguration
```

例 15: インスタンスグループ用に追加のEBSストレージボリュームが設定されたクラスターを作成するには

追加のEBSボリュームを指定する場合は、次の引数が必要です。

VolumeTypeEbsBlockDeviceConfigsが指定されSizeInGBている場合。

次のcreate-cluster例では、コアインスタンスグループのEC2インスタンスに複数のEBSボリュームがアタッチされたクラスターを作成します。

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --use-default-roles \
  --instance-
groups InstanceGroupType=MASTER, InstanceCount=1, InstanceType=d2.xlarge
'InstanceGroupType=CORE, InstanceCount=2, InstanceType=d2.xlarge, EbsConfiguration={EbsOptimiz
{VolumeSpecification={VolumeType=io1, SizeInGB=100, Iops=100}, VolumesPerInstance=4}}]'
\
  --auto-terminate
```

次の例では、マスターインスタンスグループのEC2インスタンスに複数のEBSボリュームがアタッチされたクラスターを作成します。

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --use-default-roles \
  --instance-groups 'InstanceGroupType=MASTER, InstanceCount=1,
InstanceType=d2.xlarge, EbsConfiguration={EbsOptimized=true,
EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=io1, SizeInGB=100,
Iops=100}}],
{VolumeSpecification={VolumeType=standard, SizeInGB=50}, VolumesPerInstance=3}]' InstanceGroup
\
  --auto-terminate
```

例 16: 自動スケーリングポリシーを使用してクラスターを作成するには

Amazon EMRバージョン 4.0 以降を使用して、コアインスタンスグループとタスクインスタンスグループに自動スケーリングポリシーをアタッチできます。自動スケーリングポリシーは、Amazon CloudWatch メトリクスに応答してEC2インスタンスを動的に追加および削除します。詳細については、「Amazon EMR管理ガイド」の「Amazon EMR <<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-automatic-scaling.html>>`_ で自動スケーリングを使用する」を参照してください。

自動スケーリングポリシーをアタッチする場合は、を使用して自動スケーリングのデフォルトロールを指定する必要があります--auto-scaling-role EMR_AutoScaling_DefaultRole。

次のcreate-cluster例では、スケーリングポリシー設定を指定する埋め込みJSON構造を持つ AutoScalingPolicy 引数を使用して、COREインスタンスグループの自動スケーリングポリシーを指定します。JSON 構造が埋め込まれたインスタンスグループには、引数のコレクション全体が一重引用符で囲まれている必要があります。単一引用符の使用は、JSON構造が埋め込まれていないインスタンスグループではオプションです。

```
aws emr create-cluster
  --release-label emr-5.9.0 \
  --use-default-roles --auto-scaling-role EMR_AutoScaling_DefaultRole \
  --instance-
groups 'InstanceGroupType=MASTER, InstanceType=d2.xlarge, InstanceCount=1
InstanceGroupType=CORE, InstanceType=d2.xlarge, InstanceCount=2, AutoScalingPolicy={Constrain
```

次の例では、JSON ファイルを使用してinstancegroupconfig.json、クラスター内のすべてのインスタンスグループの設定を指定します。JSON ファイルは、コアインスタンスグループの自動スケーリングポリシー設定を指定します。

```
aws emr create-cluster \  
  --release-label emr-5.9.0 \  
  --service-role EMR_DefaultRole \  
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \  
  --instance-groups file://myfolder/instancegroupconfig.json \  
  --auto-scaling-role EMR_AutoScaling_DefaultRole
```

instancegroupconfig.json の内容:

```
[  
  {  
    "InstanceCount": 1,  
    "Name": "MyMasterIG",  
    "InstanceGroupType": "MASTER",  
    "InstanceType": "m4.large"  
  },  
  {  
    "InstanceCount": 2,  
    "Name": "MyCoreIG",  
    "InstanceGroupType": "CORE",  
    "InstanceType": "m4.large",  
    "AutoScalingPolicy": {  
      "Constraints": {  
        "MinCapacity": 2,  
        "MaxCapacity": 10  
      },  
      "Rules": [  
        {  
          "Name": "Default-scale-out",  
          "Description": "Replicates the default scale-out rule in the  
console for YARN memory.",  
          "Action": {  
            "SimpleScalingPolicyConfiguration": {  
              "AdjustmentType": "CHANGE_IN_CAPACITY",  
              "ScalingAdjustment": 1,  
              "CoolDown": 300  
            }  
          },  
          "Trigger": {  
            "CloudWatchAlarmDefinition": {  
              "ComparisonOperator": "LESS_THAN",  
              "EvaluationPeriods": 1,  
              "MetricName": "YARNMemoryAvailablePercentage",
```



```

    "Namespace": "AWS/ElasticMapReduce",
    "Period": 300,
    "Threshold": 15,
    "Statistic": "AVERAGE",
    "Unit": "PERCENT",
    "Dimensions": [
      {
        "Key": "JobFlowId",
        "Value": "${emr.clusterId}"
      }
    ]
  }
}
]

```

例 17: クラスターを作成するときにカスタムJARステップを追加する

次のcreate-cluster例では、Amazon S3 に保存されているJARファイルを指定してステップを追加します。ステップは、クラスターに作業を送信します。JAR ファイルで定義されたメイン関数は、EC2インスタンスがプロビジョニングされ、ブートストラップアクションが実行され、アプリケーションがインストールされた後に実行されます。ステップはを使用して指定しますType=CUSTOM_JAR。

カスタムJARステップには、Jar=パラメータが必要です。パラメータは、のパスとファイル名を指定しますJAR。オプションのパラメータは、Type、Name、Args、ActionOnFailure、およびですMainClass。メインクラスが指定されていない場合、JARファイルはマニフェストファイルMain-Classでを指定する必要があります。

```

aws emr create-cluster \
  --steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://  

myBucket/  

mytest.jar,Args=arg1,arg2,arg3 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://  

myBucket/mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3 \  

  --release-label emr-5.3.1 \  

  --instance-  

groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE  

  \
  --auto-terminate

```

例 18: クラスターの作成時にストリーミングステップを追加するには

次のcreate-cluster例では、すべてのステップの実行後に終了するストリーミングステップをクラスターに追加します。ストリーミングステップにはパラメータ Type と Args。ストリーミングステップのオプションパラメータは Name および ActionOnFailure。

次の例では、ステップをインラインで指定します。

```
aws emr create-cluster \
  --steps Type=STREAMING,Name='Streaming Program',ActionOnFailure=CONTINUE,Args=[-
files,s3://elasticmapreduce/samples/wordcount/wordSplitter.py,-
mapper,wordSplitter.py,-reducer,aggregate,-input,s3://elasticmapreduce/samples/
wordcount/input,-output,s3://mybucket/wordcount/output] \
  --release-label emr-5.3.1 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE
\
  --auto-terminate
```

次の例では、という名前のローカルに保存されたJSON設定ファイルを使用しますmultiplefiles.json。JSON 設定では、複数のファイルを指定します。ステップ内で複数のファイルを指定するには、JSON設定ファイルを使用してステップを指定する必要があります。JSON 引数には、リスト内の独自の項目としてオプションと値を含める必要があります。

```
aws emr create-cluster \
  --steps file:///./multiplefiles.json \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE
\
  --auto-terminate
```

multiplefiles.json の内容:

```
[
  {
    "Name": "JSON Streaming Step",
    "Args": [
      "-files",
      "s3://elasticmapreduce/samples/wordcount/wordSplitter.py",
      "-mapper",
```

```

        "wordSplitter.py",
        "-reducer",
        "aggregate",
        "-input",
        "s3://elasticmapreduce/samples/wordcount/input",
        "-output",
        "s3://mybucket/wordcount/output"
    ],
    "ActionOnFailure": "CONTINUE",
    "Type": "STREAMING"
}
]

```

例 19: クラスターの作成時に Hive ステップを追加するには

次の例では、クラスターを作成するときに Hive ステップを追加します。Hive ステップには、パラメータ `Type` と `Args` が必要です。Hive ステップのオプションパラメータは `Name` および `ActionOnFailure` です。

```

aws emr create-cluster \
  --steps Type=HIVE,Name='Hive
  program',ActionOnFailure=CONTINUE,ActionOnFailure=TERMINATE_CLUSTER,Args=[-
  f,s3://elasticmapreduce/samples/hive-ads/libs/model-build.q,-d,INPUT=s3://
  elasticmapreduce/samples/hive-ads/tables,-d,OUTPUT=s3://mybucket/hive-ads/
  output/2014-04-18/11-07-32,-d,LIBS=s3://elasticmapreduce/samples/hive-ads/libs] \
  --applications Name=Hive \
  --release-label emr-5.3.1 \
  --instance-
  groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE

```

例 20: クラスターの作成時に Pig ステップを追加するには

次の例では、クラスターを作成するときに Pig ステップを追加します。豚ステップに必要なパラメータは `Type` と `Args` です。豚ステップのオプションパラメータは `Name` および `ActionOnFailure` です。

```

aws emr create-cluster \
  --steps Type=PIG,Name='Pig program',ActionOnFailure=CONTINUE,Args=[-f,s3://
  elasticmapreduce/samples/pig-apache/do-reports2.pig,-p,INPUT=s3://elasticmapreduce/
  samples/pig-apache/input,-p,OUTPUT=s3://mybucket/pig-apache/output] \
  --applications Name=Pig \
  --release-label emr-5.3.1 \

```

```
--instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

例 21: ブートストラップアクションを追加するには

次のcreate-cluster例では、Amazon S3 に保存されているスクリプトとして定義された 2 つのブートストラップアクションを実行します。

```
aws emr create-cluster \  
  --bootstrap-actions Path=s3://mybucket/  
myscript1,Name=BootstrapAction1,Args=[arg1,arg2] Path=s3://mybucket/  
myscript2,Name=BootstrapAction2,Args=[arg1,arg2] \  
  --release-label emr-5.3.1 \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE  
 \  
  --auto-terminate
```

例 22: EMRFS一貫した表示を有効にし、RetryCount と RetryPeriod の設定をカスタマイズするには

次のcreate-cluster例では、EMRFS一貫したビューの再試行回数と再試行期間を指定します。Consistent=true 引数が必要です。

```
aws emr create-cluster \  
  --instance-type m4.large \  
  --release-label emr-5.9.0 \  
  --emrfs Consistent=true,RetryCount=6,RetryPeriod=30
```

次の例では、という名前のローカルに保存されたEMRFS設定ファイルを使用して、前の例と同じJSON設定を指定しますemrfsconfig.json。

```
aws emr create-cluster \  
  --instance-type m4.large \  
  --release-label emr-5.9.0 \  
  --emrfs file://emrfsconfig.json
```

emrfsconfig.json の内容:

```
{  
  "Consistent": true,
```

```

    "RetryCount": 6,
    "RetryPeriod": 30
  }

```

例 23: Kerberos が設定されたクラスターを作成するには

次のcreate-cluster例では、Kerberos を有効にしたセキュリティ設定を使用してクラスターを作成し、を使用してクラスターの Kerberos パラメータを確立します--kerberos-attributes。

次のコマンドは、クラスターの Kerberos 属性をインラインで指定します。

```

aws emr create-cluster \
  --instance-type m3.xlarge \
  --release-label emr-5.10.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --security-configuration mySecurityConfiguration \
  --kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=123,CrossRealmTrustPrincipalPassword=123

```

次のコマンドは同じ属性を指定しますが、という名前のローカルに保存されたJSONファイルを参照しますkerberos_attributes.json。この例では、ファイルは コマンドを実行するのと同じディレクトリに保存されます。Amazon S3 に保存されている設定ファイルを参照することもできます。

```

aws emr create-cluster \
  --instance-type m3.xlarge \
  --release-label emr-5.10.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --security-configuration mySecurityConfiguration \
  --kerberos-attributes file://kerberos_attributes.json

```

kerberos_attributes.json の内容:

```

{
  "Realm": "EC2.INTERNAL",
  "KdcAdminPassword": "123",
  "CrossRealmTrustPrincipalPassword": "123",
}

```

次のcreate-cluster例では、--instance-groups設定を使用し、マネージドスケーリングポリシーを持つ Amazon EMRクラスターを作成します。

```
aws emr create-cluster \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE,
  --managed-scaling-policy
ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

次のcreate-cluster例では、「-log-encryption-kms-key-id」を使用してログ暗号化に使用されるKMSキー ID を定義する Amazon EMRクラスターを作成します。

```
aws emr create-cluster \
  --release-label emr-5.30.0 \
  --log-uri s3://myBucket/myLog \
  --log-encryption-kms-key-id arn:aws:kms:us-east-1:110302272565:key/
dd559181-283e-45d7-99d1-66da348c4d33 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.Large InstanceGroupType=CORE,
```

次のcreate-cluster例では、「-placement-group-configs」設定を使用して、SPREADプレイスメント戦略を使用してEC2プレイスメントグループ内の高可用性 (HA) クラスターにマスターノードを配置EMRする Amazon クラスターを作成します。

```
aws emr create-cluster \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m4.LargeInstanceGroupType=CORE,
  \
  --placement-group-configs InstanceRole=MASTER
```

次のcreate-cluster例では、「-auto-termination-policy」設定を使用してEMRクラスターの自動アイドル終了しきい値を配置する Amazon クラスターを作成します。

```
aws emr create-cluster \
  --release-label emr-5.34.0 \
```

```

--service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
--auto-termination-policy IdleTimeout=100

```

次のcreate-cluster例では、「--os-release-label」を使用してEMRクラスター起動用のAmazon Linux リリースを定義する Amazon クラスターを作成します。

```

aws emr create-cluster \
--release-label emr-6.6.0 \
--os-release-label 2.0.20220406.1 \
--service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE

```

例 24: EMRリリース 6.15.0 以降で作成されたクラスターインスタンスのサイズ、iops、スループットのEBSルートボリューム属性を指定するには

次のcreate-cluster例では、ルートボリューム属性を使用してEC2インスタンスのルートボリューム仕様を設定する Amazon EMRクラスターを作成します。

```

aws emr create-cluster \
--name "Cluster with My Custom AMI" \
--custom-ami-id ami-a518e6df \
--ebs-root-volume-size 20 \
--ebs-root-volume-iops 3000 \
--ebs-root-volume-throughput 125 \
--release-label emr-6.15.0 \
--use-default-roles \
--instance-count 2 \
--instance-type m4.large

```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateClusterExamples](#)」の「」を参照してください。

create-default-roles

次のコード例は、create-default-roles を使用する方法を示しています。

AWS CLI

1. のデフォルトIAMロールを作成するには EC2

コマンド:

```
aws emr create-default-roles
```

出力:

If the role already exists then the command returns nothing.

If the role does not exist then the output will be:

```
[
  {
    "RolePolicy": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": [
            "cloudwatch:*",
            "dynamodb:*",
            "ec2:Describe*",
            "elasticmapreduce:Describe*",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListInstances",
            "elasticmapreduce:ListSteps",
            "kinesis:CreateStream",
            "kinesis>DeleteStream",
            "kinesis:DescribeStream",
            "kinesis:GetRecords",
            "kinesis:GetShardIterator",
            "kinesis:MergeShards",
            "kinesis:PutRecord",
            "kinesis:SplitShard",
            "rds:Describe*",
            "s3:*",
            "sdb:*",
            "sns:*",
            "sqs:*"
          ]
        }
      ]
    }
  }
]
```



```
        ],
        "Resource": "*",
        "Effect": "Allow"
    }
]
},
"Role": {
    "AssumeRolePolicyDocument": {
        "Version": "2008-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AR0AIQ5SIUGL5KMYBJX6",
    "CreateDate": "2015-06-09T17:09:04.602Z",
    "RoleName": "EMR_EC2_DefaultRole",
    "Path": "/",
    "Arn": "arn:aws:iam::176430881729:role/EMR_EC2_DefaultRole"
}
},
{
    "RolePolicy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "ec2:AuthorizeSecurityGroupIngress",
                    "ec2:CancelSpotInstanceRequests",
                    "ec2:CreateSecurityGroup",
                    "ec2:CreateTags",
                    "ec2>DeleteTags",
                    "ec2:DescribeAvailabilityZones",
                    "ec2:DescribeAccountAttributes",
                    "ec2:DescribeInstances",
                    "ec2:DescribeInstanceStatus",
                    "ec2:DescribeKeyPairs",
                    "ec2:DescribePrefixLists",
```

```

        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeSpotPriceHistory",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcs",
        "ec2:ModifyImageAttribute",
        "ec2:ModifyInstanceAttribute",
        "ec2:RequestSpotInstances",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListInstanceProfiles",
        "iam:ListRolePolicies",
        "iam:PassRole",
        "s3:CreateBucket",
        "s3:Get*",
        "s3:List*",
        "sdb:BatchPutAttributes",
        "sdb:Select",
        "sqs:CreateQueue",
        "sqs>Delete*",
        "sqs:GetQueue*",
        "sqs:ReceiveMessage"
    ],
    "Resource": "*",
    "Effect": "Allow"
}

]
},
"Role": {
    "AssumeRolePolicyDocument": {
        "Version": "2008-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "elasticmapreduce.amazonaws.com"
                }
            }
        ]
    }
}

```

```

    }
  }
]
},
"RoleId": "AROAI3SRVPPVSRDLARBPY",
"CreateDate": "2015-06-09T17:09:10.401Z",
"RoleName": "EMR_DefaultRole",
"Path": "/",
"Arn": "arn:aws:iam::176430881729:role/EMR_DefaultRole"
}
}
]

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDefaultRoles](#)」の「」を参照してください。

create-security-configuration

次のコード例は、create-security-configuration を使用する方法を示しています。

AWS CLI

1. 証明書プロバイダー PEM の場合は で転送時の暗号化を有効にし、ローカルディスクキープロバイダー AWS の場合は SSE-S3 で保管時の S3 暗号化を有効にして KMS セキュリティ設定を作成するには

コマンド:

```

aws emr create-security-configuration --name MySecurityConfig --security-
configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption" : true,
    "EnableAtRestEncryption" : true,
    "InTransitEncryptionConfiguration" : {
      "TLSCertificateConfiguration" : {
        "CertificateProviderType" : "PEM",
        "S3Object" : "s3://mycertstore/artifacts/
MyCerts.zip"
      }
    },
    "AtRestEncryptionConfiguration" : {
      "S3EncryptionConfiguration" : {
        "EncryptionMode" : "SSE-S3"
      }
    }
  }
}

```

```

        },
        "LocalDiskEncryptionConfiguration" : {
            "EncryptionKeyProviderType" : "AwsKms",
            "AwsKmsKey" : "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
        }
    }
}'

```

出力:

```

{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}

```

JSON 同等 (security_configuration.json の内容):

```

{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "EnableAtRestEncryption": true,
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "PEM",
        "S3Object": "s3://mycertstore/artifacts/MyCerts.zip"
      }
    },
    "AtRestEncryptionConfiguration": {
      "S3EncryptionConfiguration": {
        "EncryptionMode": "SSE-S3"
      },
      "LocalDiskEncryptionConfiguration": {
        "EncryptionKeyProviderType": "AwsKms",
        "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
      }
    }
  }
}

```

コマンド (security_configuration.json を使用):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-configuration file:///./security_configuration.json
```

出力:

```
{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}
```

2. クラスター専用KDCおよびクロスレルム信頼を使用して Kerberos を有効にしてセキュリティ設定を作成するには

コマンド:

```
aws emr create-security-configuration --name MySecurityConfig --security-configuration '{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24,
        "CrossRealmTrustConfiguration": {
          "Realm": "AD.DOMAIN.COM",
          "Domain": "ad.domain.com",
          "AdminServer": "ad.domain.com",
          "KdcServer": "ad.domain.com"
        }
      }
    }
  }
}'
```

出力:

```
{
  "CreationDateTime": 1490225558.982,
  "Name": "MySecurityConfig"
}
```

JSON 同等 (security_configuration.json の内容):

```
{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24,
        "CrossRealmTrustConfiguration": {
          "Realm": "AD.DOMAIN.COM",
          "Domain": "ad.domain.com",
          "AdminServer": "ad.domain.com",
          "KdcServer": "ad.domain.com"
        }
      }
    }
  }
}
```

コマンド (security_configuration.json を使用):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-configuration file:///./security_configuration.json
```

出力:

```
{
  "CreationDateTime": 1490225558.982,
  "Name": "MySecurityConfig"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSecurityConfiguration](#)」の「」を参照してください。

delete-security-configuration

次の例は、delete-security-configuration を使用する方法を説明しています。

AWS CLI

現在のリージョンのセキュリティ設定を削除するには

コマンド:

```
aws emr delete-security-configuration --name MySecurityConfig
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteSecurityConfiguration](#)」の「」を参照してください。

describe-cluster

次のコード例は、describe-cluster を使用する方法を示しています。

AWS CLI

コマンド:

```
aws emr describe-cluster --cluster-id j-XXXXXXXX
```

出力:

```
For release-label based uniform instance groups cluster:
```

```
{
  "Cluster": {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1436475075.199,
        "CreationDateTime": 1436474656.563,
      },
      "State": "WAITING",
      "StateChangeReason": {
        "Message": "Waiting for steps to run"
      }
    },
    "Ec2InstanceAttributes": {
      "ServiceAccessSecurityGroup": "sg-xxxxxxx",
      "EmrManagedMasterSecurityGroup": "sg-xxxxxxx",
      "IamInstanceProfile": "EMR_EC2_DefaultRole",
      "Ec2KeyName": "myKey",
      "Ec2AvailabilityZone": "us-east-1c",
    }
  }
}
```

```
    "EmrManagedSlaveSecurityGroup": "sg-yyyyyyyyyy"
  },
  "Name": "My Cluster",
  "ServiceRole": "EMR_DefaultRole",
  "Tags": [],
  "TerminationProtected": true,
  "UnhealthyNodeReplacement": true,
  "ReleaseLabel": "emr-4.0.0",
  "NormalizedInstanceHours": 96,
  "InstanceGroups": [
    {
      "RequestedInstanceCount": 2,
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1436475074.245,
          "CreationDateTime": 1436474656.564,
          "EndDateTime": 1436638158.387
        },
        "State": "RUNNING",
        "StateChangeReason": {
          "Message": ""
        }
      },
      "Name": "CORE",
      "InstanceGroupType": "CORE",
      "Id": "ig-YYYYYYYY",
      "Configurations": [],
      "InstanceType": "m3.large",
      "Market": "ON_DEMAND",
      "RunningInstanceCount": 2
    },
    {
      "RequestedInstanceCount": 1,
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1436475074.245,
          "CreationDateTime": 1436474656.564,
          "EndDateTime": 1436638158.387
        },
        "State": "RUNNING",
        "StateChangeReason": {
          "Message": ""
        }
      }
    }
  ],
```



```

        "Name": "MASTER",
        "InstanceGroupType": "MASTER",
        "Id": "ig-XXXXXXXX",
        "Configurations": [],
        "InstanceType": "m3.large",
        "Market": "ON_DEMAND",
        "RunningInstanceCount": 1
    }
],
"Applications": [
    {
        "Name": "Hadoop"
    }
],
"VisibleToAllUsers": true,
"BootstrapActions": [],
"MasterPublicDnsName": "ec2-54-147-144-78.compute-1.amazonaws.com",
"AutoTerminate": false,
"Id": "j-XXXXXXXX",
"Configurations": [
    {
        "Properties": {
            "fs.s3.consistent.retryPeriodSeconds": "20",
            "fs.s3.enableServerSideEncryption": "true",
            "fs.s3.consistent": "false",
            "fs.s3.consistent.retryCount": "2"
        },
        "Classification": "emrfs-site"
    }
]
}
}
}

```

For release-label based instance fleet cluster:

```

{
  "Cluster": {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1487897289.705,
        "CreationDateTime": 1487896933.942
      },
      "State": "WAITING",
      "StateChangeReason": {

```

```
        "Message": "Waiting for steps to run"
      }
    },
    "Ec2InstanceAttributes": {
      "EmrManagedMasterSecurityGroup": "sg-xxxxx",
      "RequestedEc2AvailabilityZones": [],
      "RequestedEc2SubnetIds": [],
      "IamInstanceProfile": "EMR_EC2_DefaultRole",
      "Ec2AvailabilityZone": "us-east-1a",
      "EmrManagedSlaveSecurityGroup": "sg-xxxxx"
    },
    "Name": "My Cluster",
    "ServiceRole": "EMR_DefaultRole",
    "Tags": [],
    "TerminationProtected": false,
    "UnhealthyNodeReplacement": false,
    "ReleaseLabel": "emr-5.2.0",
    "NormalizedInstanceHours": 472,
    "InstanceCollectionType": "INSTANCE_FLEET",
    "InstanceFleets": [
      {
        "Status": {
          "Timeline": {
            "ReadyDateTime": 1487897212.74,
            "CreationDateTime": 1487896933.948
          },
          "State": "RUNNING",
          "StateChangeReason": {
            "Message": ""
          }
        }
      },
      {
        "ProvisionedSpotCapacity": 1,
        "Name": "MASTER",
        "InstanceFleetType": "MASTER",
        "LaunchSpecifications": {
          "SpotSpecification": {
            "TimeoutDurationMinutes": 60,
            "TimeoutAction": "TERMINATE_CLUSTER"
          }
        }
      },
      {
        "TargetSpotCapacity": 1,
        "ProvisionedOnDemandCapacity": 0,
        "InstanceTypeSpecifications": [
          {
```

```

        "BidPrice": "0.5",
        "InstanceType": "m3.xlarge",
        "WeightedCapacity": 1
    }
],
    "Id": "if-xxxxxxx",
    "TargetOnDemandCapacity": 0
}
],
"Applications": [
    {
        "Version": "2.7.3",
        "Name": "Hadoop"
    }
],
"ScaleDownBehavior": "TERMINATE_AT_INSTANCE_HOUR",
"VisibleToAllUsers": true,
"BootstrapActions": [],
"MasterPublicDnsName": "ec2-xxx-xx-xxx-xx.compute-1.amazonaws.com",
"AutoTerminate": false,
"Id": "j-xxxxx",
"Configurations": []
}
}

```

For ami based uniform instance group cluster:

```

{
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1399400564.432,
                "CreationDateTime": 1399400268.62
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting for steps to run"
            }
        },
        "Ec2InstanceAttributes": {
            "IamInstanceProfile": "EMR_EC2_DefaultRole",
            "Ec2AvailabilityZone": "us-east-1c"
        },
        "Name": "My Cluster",
    }
}

```

```
"Tags": [],
"TerminationProtected": true,
"UnhealthyNodeReplacement": true,
"RunningAmiVersion": "2.5.4",
"InstanceGroups": [
  {
    "RequestedInstanceCount": 1,
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1399400558.848,
        "CreationDateTime": 1399400268.621
      },
      "State": "RUNNING",
      "StateChangeReason": {
        "Message": ""
      }
    },
    "Name": "Master instance group",
    "InstanceGroupType": "MASTER",
    "InstanceType": "m1.small",
    "Id": "ig-ABCD",
    "Market": "ON_DEMAND",
    "RunningInstanceCount": 1
  },
  {
    "RequestedInstanceCount": 2,
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1399400564.439,
        "CreationDateTime": 1399400268.621
      },
      "State": "RUNNING",
      "StateChangeReason": {
        "Message": ""
      }
    },
    "Name": "Core instance group",
    "InstanceGroupType": "CORE",
    "InstanceType": "m1.small",
    "Id": "ig-DEF",
    "Market": "ON_DEMAND",
    "RunningInstanceCount": 2
  }
],
```

```
    "Applications": [  
      {  
        "Version": "1.0.3",  
        "Name": "hadoop"  
      }  
    ],  
    "BootstrapActions": [],  
    "VisibleToAllUsers": false,  
    "RequestedAmiVersion": "2.4.2",  
    "LogUri": "s3://myLogUri/",  
    "AutoTerminate": false,  
    "Id": "j-XXXXXXXX"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeCluster](#)」の「」を参照してください。

describe-step

次の例は、describe-step を使用する方法を説明しています。

AWS CLI

次のコマンドは、クラスター ID `j-3SD91U2E1L2QX` を持つクラスターでステップ ID `s-3LZC0QUT43AM` を持つステップを記述します。

```
aws emr describe-step --cluster-id j-3SD91U2E1L2QX --step-id s-3LZC0QUT43AM
```

出力:

```
{  
  "Step": {  
    "Status": {  
      "Timeline": {  
        "EndDateTime": 1433200470.481,  
        "CreationDateTime": 1433199926.597,  
        "StartDateTime": 1433200404.959  
      },  
      "State": "COMPLETED",  
      "StateChangeReason": {}  
    }  
  }  
}
```

```
    },
    "Config": {
      "Args": [
        "s3://us-west-2.elasticmapreduce/libs/hive/hive-script",
        "--base-path",
        "s3://us-west-2.elasticmapreduce/libs/hive/",
        "--install-hive",
        "--hive-versions",
        "0.13.1"
      ],
      "Jar": "s3://us-west-2.elasticmapreduce/libs/script-runner/script-
runner.jar",
      "Properties": {}
    },
    "Id": "s-3LZC0QUT43AM",
    "ActionOnFailure": "TERMINATE_CLUSTER",
    "Name": "Setup hive"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStep](#)」の「」を参照してください。

get

次の例は、get を使用する方法を説明しています。

AWS CLI

次の例は、クラスター ID を持つクラスター内のマスターインスタンスから `hadoop-examples.jar` アーカイブをダウンロードします `j-3SD91U2E1L2QX`。

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /home/hadoop-examples.jar --dest ~
```

- API 詳細については、「コマンドリファレンスの [取得](#)」を参照してください。AWS CLI

list-clusters

次のコード例は、list-clusters を使用する方法を示しています。

AWS CLI

次のコマンドは、現在のリージョン内のすべてのアクティブなEMRクラスターを一覧表示します。

```
aws emr list-clusters --active
```

出力:

```
{
  "Clusters": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1433200405.353,
          "CreationDateTime": 1433199926.596
        },
        "State": "WAITING",
        "StateChangeReason": {
          "Message": "Waiting after step completed"
        }
      },
      "NormalizedInstanceHours": 6,
      "Id": "j-3SD91U2E1L2QX",
      "Name": "my-cluster"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListClusters](#)」の「」を参照してください。

list-instance-fleets

次のコード例は、list-instance-fleets を使用する方法を示しています。

AWS CLI

クラスター内のインスタンスフリートの設定の詳細を取得するには

この例では、指定されたクラスター内のインスタンスフリートの詳細を一覧表示します。

コマンド:

```
list-instance-fleets --cluster-id 'j-12ABCDEFGH134JK'
```

出力:

```
{
  "InstanceFleets": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1488759094.637,
          "CreationDateTime": 1488758719.817
        },
        "State": "RUNNING",
        "StateChangeReason": {
          "Message": ""
        }
      },
      "ProvisionedSpotCapacity": 6,
      "Name": "CORE",
      "InstanceFleetType": "CORE",
      "LaunchSpecifications": {
        "SpotSpecification": {
          "TimeoutDurationMinutes": 60,
          "TimeoutAction": "TERMINATE_CLUSTER"
        }
      },
      "ProvisionedOnDemandCapacity": 2,
      "InstanceTypeSpecifications": [
        {
          "BidPrice": "0.5",
          "InstanceType": "m3.xlarge",
          "WeightedCapacity": 2
        }
      ],
      "Id": "if-1ABC2DEFGHIJ3"
    },
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1488759058.598,
          "CreationDateTime": 1488758719.811
        }
      }
    }
  ]
}
```



```
    },
    "State": "RUNNING",
    "StateChangeReason": {
      "Message": ""
    }
  },
  "ProvisionedSpotCapacity": 0,
  "Name": "MASTER",
  "InstanceFleetType": "MASTER",
  "ProvisionedOnDemandCapacity": 1,
  "InstanceTypeSpecifications": [
    {
      "BidPriceAsPercentageOfOnDemandPrice": 100.0,
      "InstanceType": "m3.xlarge",
      "WeightedCapacity": 1
    }
  ],
  "Id": "if-2ABC4DEFGHIJ4"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInstanceFleets](#)」の「」を参照してください。

list-instances

次のコード例は、list-instances を使用する方法を示しています。

AWS CLI

次のコマンドは、クラスター ID を持つクラスター内のすべてのインスタンスを一覧表示します `j-3C6XNQ39VR9WL`。

```
aws emr list-instances --cluster-id j-3C6XNQ39VR9WL
```

出力:

```
For a uniform instance group based cluster
{
  "Instances": [
    {
```

```
"Status": {
  "Timeline": {
    "ReadyDateTime": 1433200400.03,
    "CreationDateTime": 1433199960.152
  },
  "State": "RUNNING",
  "StateChangeReason": {}
},
"Ec2InstanceId": "i-f19ecfee",
"PublicDnsName": "ec2-52-52-41-150.us-west-2.compute.amazonaws.com",
"PrivateDnsName": "ip-172-21-11-216.us-west-2.compute.internal",
"PublicIpAddress": "52.52.41.150",
"Id": "ci-3NNHQUQ2TWB6Y",
"PrivateIpAddress": "172.21.11.216"
},
{
  "Status": {
    "Timeline": {
      "ReadyDateTime": 1433200400.031,
      "CreationDateTime": 1433199949.102
    },
    "State": "RUNNING",
    "StateChangeReason": {}
  },
  "Ec2InstanceId": "i-1feee4c2",
  "PublicDnsName": "ec2-52-63-246-32.us-west-2.compute.amazonaws.com",
  "PrivateDnsName": "ip-172-31-24-130.us-west-2.compute.internal",
  "PublicIpAddress": "52.63.246.32",
  "Id": "ci-GAOCMKNKDCV7",
  "PrivateIpAddress": "172.21.11.215"
},
{
  "Status": {
    "Timeline": {
      "ReadyDateTime": 1433200400.031,
      "CreationDateTime": 1433199949.102
    },
    "State": "RUNNING",
    "StateChangeReason": {}
  },
  "Ec2InstanceId": "i-15cfeee3",
  "PublicDnsName": "ec2-52-25-246-63.us-west-2.compute.amazonaws.com",
  "PrivateDnsName": "ip-172-31-24-129.us-west-2.compute.internal",
  "PublicIpAddress": "52.25.246.63",
```

```

        "Id": "ci-2W3TDFFB47UAD",
        "PrivateIpAddress": "172.21.11.214"
    }
]
}

```

For a fleet based cluster:

```

{
  "Instances": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1487810810.878,
          "CreationDateTime": 1487810588.367,
          "EndDateTime": 1488022990.924
        },
        "State": "TERMINATED",
        "StateChangeReason": {
          "Message": "Instance was terminated."
        }
      },
      "Ec2InstanceId": "i-xxxxx",
      "InstanceFleetId": "if-xxxxx",
      "EbsVolumes": [],
      "PublicDnsName": "ec2-xx-xxx-xxx-xxx.compute-1.amazonaws.com",
      "InstanceType": "m3.xlarge",
      "PrivateDnsName": "ip-xx-xx-xxx-xx.ec2.internal",
      "Market": "SPOT",
      "PublicIpAddress": "xx.xx.xxx.xxx",
      "Id": "ci-xxxxx",
      "PrivateIpAddress": "10.47.191.80"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListInstances](#)」の「」を参照してください。

list-security-configurations

次のコード例は、list-security-configurations を使用する方法を示しています。

AWS CLI

現在のリージョンのセキュリティ設定を一覧表示するには

コマンド:

```
aws emr list-security-configurations
```

出力:

```
{
  "SecurityConfigurations": [
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-1"
    },
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-2"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSecurityConfigurations](#)」の「」を参照してください。

list-steps

次の例は、list-steps を使用する方法を説明しています。

AWS CLI

次のコマンドは、クラスター ID `j-3SD91U2E1L2QX` を持つクラスターのすべてのステップを一覧表示します。

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSteps](#)」の「」を参照してください。

modify-cluster-attributes

次のコード例は、`modify-cluster-attributes` を使用する方法を示しています。

AWS CLI

次のコマンドは、ID を持つEMRクラスターの可視性 `j-301CDNY0J5XM4` をすべてのユーザーに設定します。

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-users
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyClusterAttributes](#)」の「」を参照してください。

modify-instance-fleet

次の例は、`modify-instance-fleet` を使用する方法を説明しています。

AWS CLI

インスタンスフリートのターゲット容量を変更するには

この例では、指定されたインスタンスフリートのオンデマンドおよびスポットターゲット容量を 1 に変更します。

コマンド:

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGH134JK' --instance-fleet InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyInstanceFleet](#)」の「」を参照してください。

put

次のコード例は、`put` を使用する方法を示しています。

AWS CLI

次のコマンドは、という名前 `healthcheck.sh` のファイルを、クラスター ID のクラスター内のマスターインスタンスにアップロードします `j-3SD91U2E1L2QX`。

```
aws emr put --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src ~/scripts/healthcheck.sh --dest /home/hadoop/bin/healthcheck.sh
```

- API 詳細については、<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/emr/put.html> 「コマンドAWS CLI リファレンスに入れる」を参照してください。

remove-tags

次のコード例は、remove-tags を使用する方法を示しています。

AWS CLI

次のコマンドは、キーを持つタグをクラスター ID を持つクラスターprodから削除しますj-3SD91U2E1L2QX。

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveTags](#)」の「」を参照してください。

schedule-hbase-backup

次の例は、schedule-hbase-backup を使用する方法を説明しています。

AWS CLI

注: このコマンドは、AMIバージョン 2.x および 3.x HBaseでのみ使用できます。

1. フルHBaseバックアップをスケジュールするには >>>>>>
06ab6d6e13564b5733d75abaf3b599f93cf39a23

コマンド:

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type full --dir s3://myBucket/backup --interval 10 --unit hours --start-time 2014-04-21T05:26:10Z --consistent
```

出力:

```
None
```

2. 増分HBaseバックアップをスケジュールするには

コマンド:

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type incremental
--dir s3://myBucket/backup --interval 30 --unit minutes --start-time
2014-04-21T05:26:10Z --consistent
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ScheduleHbaseBackup](#)」の「」を参照してください。

socks

次のコード例は、socks を使用する方法を示しています。

AWS CLI

次のコマンドは、クラスター ID が のクラスター内のマスターインスタンスとのソックス接続を開きます `j-3SD91U2E1L2QX`。

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

キーペアファイルオプションは、プライベートキーファイルへのローカルパスを取得します。

- API 詳細については、AWS CLI 「コマンドリファレンス」の「[Socks](#)」を参照してください。

ssh

次のコード例は、ssh を使用する方法を示しています。

AWS CLI

次のコマンドは、クラスター ID が のクラスター内のマスターインスタンスとの ssh 接続を開きます `j-3SD91U2E1L2QX`。

```
aws emr ssh --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

キーペアファイルオプションは、プライベートキーファイルへのローカルパスを取得します。

出力:

```
ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=10 -i /home/local/user/.ssh/mykey.pem hadoop@ec2-52-52-41-150.us-west-2.compute.amazonaws.com
Warning: Permanently added 'ec2-52-52-41-150.us-west-2.compute.amazonaws.com,52.52.41.150' (ECDSA) to the list of known hosts.
Last login: Mon Jun  1 23:15:38 2015

  _|  _|_  )
  _| (    /  Amazon Linux AMI
  __|\__|__|

https://aws.amazon.com/amazon-linux-ami/2015.03-release-notes/
26 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.

-----

Welcome to Amazon Elastic MapReduce running Hadoop and Amazon Linux.

Hadoop is installed in /home/hadoop. Log files are in /mnt/var/log/hadoop. Check
/mnt/var/log/hadoop/steps for diagnosing step failures.

The Hadoop UI can be accessed via the following commands:

ResourceManager    lynx http://ip-172-21-11-216:9026/
NameNode           lynx http://ip-172-21-11-216:9101/

-----

[hadoop@ip-172-31-16-216 ~]$
```

- API 詳細については、「コマンドリファレンス」の「[Ssh](#)」を参照してください。AWS CLI

を使用した Amazon EMR on EKS の例 AWS CLI

次のコード例は、EMR で Amazon AWS Command Line Interface を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますEKS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

update-role-trust-policy

次の例は、update-role-trust-policy を使用する方法を説明しています。

AWS CLI

EMRで Amazon で使用するIAMロールの信頼ポリシーを更新するには EKS

このコマンド例では、example_iam_role という名前のロールの信頼ポリシーを更新して、example_cluster という名前のEKSクラスターの example_namespace 名前空間EKSを使用してEMRでAmazonで使えるようにします。

コマンド:

```
aws emr-containers update-role-trust-policy \  
  --cluster example_cluster \  
  --namespace example_namespace \  
  --role-name example_iam_role
```

出力:

```
If the trust policy has already been updated, then the output will be:  
Trust policy statement already exists for role example_iam_role. No  
changes were made!
```

```
If the trust policy has not been updated yet, then the output will be:  
Successfully updated trust policy of role example_iam_role.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRoleTrustPolicy](#)」の「」を参照してください。

EventBridge を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています EventBridge。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-rule

次の例は、delete-rule を使用する方法を説明しています。

AWS CLI

CloudWatch イベントルールを削除するには

この例では、 という名前のルールを削除します EC2InstanceStateChanges。

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRule](#)」の「」を参照してください。

describe-rule

次の例は、describe-rule を使用する方法を説明しています。

AWS CLI

CloudWatch イベントルールに関する情報を表示するには

この例では、 という名前のルールに関する情報を表示します DailyLambdaFunction。

```
aws events describe-rule --name "DailyLambdaFunction"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRule](#)」の「」を参照してください。

disable-rule

次のコード例は、disable-rule を使用する方法を示しています。

AWS CLI

CloudWatch イベントルールを無効にするには

この例では、 という名前のルールを無効にします DailyLambdaFunction。ルールは削除されません。

```
aws events disable-rule --name "DailyLambdaFunction"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableRule](#)」の「」を参照してください。

enable-rule

次の例は、enable-rule を使用する方法を説明しています。

AWS CLI

CloudWatch イベントルールを有効にするには

この例では DailyLambdaFunction、以前に無効にされていた という名前のルールを有効にします。

```
aws events enable-rule --name "DailyLambdaFunction"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableRule](#)」の「」を参照してください。

list-rule-names-by-target

次の例は、list-rule-names-by-target を使用する方法を説明しています。

AWS CLI

ターゲットが指定されているルールをすべて表示するには

この例では、MyFunctionName 「」という名前の Lambda 関数をターゲットとするすべてのルールを表示します。

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRuleNamesByTarget](#)」の「」を参照してください。

list-rules

次の例は、list-rules を使用する方法を説明しています。

AWS CLI

すべての CloudWatch イベントルールのリストを表示するには

この例では、リージョン内のすべての CloudWatch イベントルールを表示します。

```
aws events list-rules
```

特定の文字列で始まる CloudWatch イベントルールのリストを表示します。

この例では、「日」で始まる名前を持つリージョン内のすべての CloudWatch イベントルールを表示します。

```
aws events list-rules --name-prefix "Daily"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRules](#)」の「」を参照してください。

list-targets-by-rule

次の例は、list-targets-by-rule を使用する方法を説明しています。

AWS CLI

CloudWatch イベントルールのすべてのターゲットを表示するには

この例では、 という名前のルールのすべてのターゲットを表示します DailyLambdaFunction。

```
aws events list-targets-by-rule --rule "DailyLambdaFunction"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTargetsByRule](#)」の「」を参照してください。

put-events

次のコード例は、put-events を使用する方法を示しています。

AWS CLI

カスタムイベントを CloudWatch イベントに送信するには

この例では、カスタムイベントを CloudWatch Events に送信します。このイベントは putevents.json ファイルに含まれています。

```
aws events put-events --entries file://putevents.json
```

putevents.json ファイルの内容は次のとおりです。

```
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
  },
  {
```

```
"Source": "com.mycompany.myapp",
"Detail": "{ \"key1\": \"value3\", \"key2\": \"value4\" }",
"Resources": [
  "resource1",
  "resource2"
],
"DetailType": "myDetailType"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutEvents](#)」の「」を参照してください。

put-rule

次の例は、put-rule を使用する方法を説明しています。

AWS CLI

CloudWatch イベントルールを作成するには

この例では、毎日午前 9:00 () にトリガーするルールを作成しますUTC。put-targets を使用して Lambda 関数をこのルールのターゲットとして追加すると、指定した時刻に Lambda 関数を毎日実行できます。

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9 * * ? *)"
```

この例では、リージョン内のEC2インスタンスの状態が変更されたときに がトリガーするルールを作成します。

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"source\": [\"aws.ec2\"],\"detail-type\": [\"EC2 Instance State-change Notification\"]}" --role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

この例では、リージョン内のEC2インスタンスが停止または終了したときにトリガーするルールを作成します。

```
aws events put-rule --name "EC2InstanceStateChangeStopOrTerminate" --event-pattern "{\"source\": [\"aws.ec2\"],\"detail-type\": [\"EC2 Instance State-change
```

```
Notification\"],\"detail\":{\"state\":[\"stopped\",\"terminated\"]}}" --role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRule](#)」の「」を参照してください。

put-targets

次の例は、put-targets を使用する方法を説明しています。

AWS CLI

CloudWatch イベントルールのターゲットを追加するには

この例は、ルールのターゲットとして Lambda 関数を追加します。

```
aws events put-targets --rule DailyLambdaFunction --targets "Id"="1", "Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

この例は、Amazon Kinesis ストリームをターゲットとして設定し、このルールによって捕捉されたイベントがストリームに中継されるようにします。

```
aws events put-targets --rule EC2InstanceStateChanges --targets "Id"="1", "Arn"="arn:aws:kinesis:us-east-1:123456789012:stream/MyStream", "RoleArn"="arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

この例は、2 つの Amazon Kinesis ストリームを 1 つのルールのターゲットとして設定します。

```
aws events put-targets --rule DailyLambdaFunction --targets "Id"="Target1", "Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/MyStream1", "RoleArn"="arn:aws:iam::379642911888:role/ MyRoleToAccessLambda" "Id"="Target2", " Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/MyStream2", "RoleArn"="arn:aws:iam::379642911888:role/MyRoleToAccessLambda"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutTargets](#)」の「」を参照してください。

remove-targets

次の例は、remove-targets を使用する方法を説明しています。

AWS CLI

イベントのターゲットを削除するには

この例では、MyStream1 という名前の Amazon Kinesis ストリームをルールターゲットから削除します DailyLambdaFunction。DailyLambdaFunction が作成されると、このストリームはターゲット 1 の Target1 として設定されました。

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveTargets](#)」の「」を参照してください。

test-event-pattern

次の例は、test-event-pattern を使用する方法を説明しています。

AWS CLI

イベントパターンが指定されたイベントと一致するかどうかを確認するには

この例では、パターン「source:com.mycompany.myapp」が指定されたイベントと一致するかどうかをテストします。この例では、出力は「true」になります。

```
aws events test-event-pattern --event-pattern "{\"source\": [\"com.mycompany.myapp\"]}" --event "{\"id\": \"1\", \"source\": \"com.mycompany.myapp\", \"detail-type\": \"myDetailType\", \"account\": \"123456789012\", \"region\": \"us-east-1\", \"time\": \"2017-04-11T20:11:04Z\"}"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [TestEventPattern](#)」の「」を参照してください。

を使用した Firewall Manager の例 AWS CLI

次のコード例は、Firewall Manager AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-admin-account

次のコード例は、`associate-admin-account` を使用する方法を示しています。

AWS CLI

Firewall Manager 管理者アカウントを設定するには

次の`associate-admin-account`例では、Firewall Manager の管理者アカウントを設定します。

```
aws fms associate-admin-account \  
  --admin-account 123456789012
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Firewall Manager」 および 「Shield Advanced Developer Guide」 の 「Firewall Manager 管理者アカウントを設定する」](#) を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateAdminAccount](#)」 の「」を参照してください。

delete-notification-channel

次の例は、`delete-notification-channel` を使用する方法を説明しています。

AWS CLI

Firewall Manager ログのSNSトピック情報を削除するには

次の`delete-notification-channel`例では、SNSトピック情報を削除します。

```
aws fms delete-notification-channel
```

このコマンドでは何も出力されません。

詳細については、[「、ファイアウォールマネージャー、Shield Advanced デベロッパーガイド」の「Amazon SNS Notifications と Amazon CloudWatch Alarms の設定」](#)を参照してください。

AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteNotificationChannel](#)」の「」を参照してください。

delete-policy

次のコード例は、delete-policy を使用する方法を示しています。

AWS CLI

Firewall Manager ポリシーを削除するには

次のdelete-policy例では、指定された ID のポリシーとそのすべてのリソースを削除します。

```
aws fms delete-policy \  
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --delete-all-policy-resources
```

このコマンドでは何も出力されません。

詳細については、、、[AWS Firewall Manager、および Shield Advanced デベロッパーガイドの「Firewall Manager ポリシーの使用」](#)を参照してください。 AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicy](#)」の「」を参照してください。

disassociate-admin-account

次の例は、disassociate-admin-account を使用する方法を説明しています。

AWS CLI

Firewall Manager 管理者アカウントを削除するには

次のdisassociate-admin-account例では、Firewall Manager から現在の管理者アカウントの関連付けを削除します。

```
aws fms disassociate-admin-account
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Firewall Manager、Shield Advanced デベロッパーガイド」の「Firewall Manager 管理者アカウントを設定する」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateAdminAccount](#)」の「」を参照してください。

get-admin-account

次のコード例は、get-admin-account を使用する方法を示しています。

AWS CLI

Firewall Manager 管理者アカウントを取得するには

次のget-admin-account例では、管理者アカウントを取得します。

```
aws fms get-admin-account
```

出力:

```
{
  "AdminAccount": "123456789012",
  "RoleStatus": "READY"
}
```

詳細については、[AWS 「Firewall Manager、Shield Advanced デベロッパーガイド」の「Firewall Manager の前提条件」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAdminAccount](#)」の「」を参照してください。

get-compliance-detail

次の例は、get-compliance-detail を使用する方法を説明しています。

AWS CLI

アカウントのコンプライアンス情報を取得するには

次のget-compliance-detail例では、指定されたポリシーとメンバーアカウントのコンプライアンス情報を取得します。

```
aws fms get-compliance-detail \  
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --member-account 123456789012
```

出力:

```
{  
  "PolicyComplianceDetail": {  
    "EvaluationLimitExceeded": false,  
    "IssueInfoMap": {},  
    "MemberAccount": "123456789012",  
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "PolicyOwner": "123456789012",  
    "Violators": []  
  }  
}
```

詳細については、AWS WAF AWS 「[ファイアウォールマネージャー、および AWS Shield Advanced デベロッパーガイド](#)」の「[ポリシーによるリソースコンプライアンスの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetComplianceDetail](#)」の「」を参照してください。

get-notification-channel

次の例は、get-notification-channel を使用する方法を説明しています。

AWS CLI

Firewall Manager ログのSNSトピック情報を取得するには

次のget-notification-channel例では、SNSトピック情報を取得します。

```
aws fms get-notification-channel
```

出力:

```
{
```

```

    "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:us-west-2-fms",
    "SnsRoleName": "arn:aws:iam::123456789012:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS"
}

```

詳細については、[「Firewall Manager、Shield Advanced デベロッパーガイド」の「Amazon SNS Notifications と Amazon CloudWatch Alarms の設定」](#)を参照してください。AWS WAF
AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetNotificationChannel](#)」の「」を参照してください。

get-policy

次のコード例は、get-policy を使用する方法を示しています。

AWS CLI

Firewall Manager ポリシーを取得するには

次のget-policy例では、指定された ID のポリシーを取得します。

```

aws fms get-policy \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

出力:

```

{
  "Policy": {
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "PolicyName": "test",
    "PolicyUpdateToken": "1:p+2RpKR4wPFx7mcrL1U0QQ==",
    "SecurityServicePolicyData": {
      "Type": "SECURITY_GROUPS_COMMON",
      "ManagedServiceData": "{\"type\":\"SECURITY_GROUPS_COMMON\",
\\revertManualSecurityGroupChanges\":true,\\exclusiveResourceSecurityGroupManagement
\\:false,\\securityGroups\":[\\id\\:\\sg-045c43ccc9724e63e\\]}\"
    },
    "ResourceType": "AWS::EC2::Instance",
    "ResourceTags": [],
    "ExcludeResourceTags": false,
    "RemediationEnabled": false
  }
}

```

```
  },
  "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/d1ac59b8-938e-42b3-
b2e0-7c620422ddc2"
}
```

詳細については、[AWS 「Firewall Manager、Shield Advanced デベロッパーガイド」の「Firewall Manager ポリシーの使用」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicy](#)」の「」を参照してください。

list-compliance-status

次の例は、list-compliance-status を使用する方法を説明しています。

AWS CLI

メンバーアカウントのポリシーコンプライアンス情報を取得するには

次のlist-compliance-status例では、指定されたポリシーのメンバーアカウントコンプライアンス情報を取得します。

```
aws fms list-compliance-status \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{
  "PolicyComplianceStatusList": [
    {
      "PolicyOwner": "123456789012",
      "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "PolicyName": "test",
      "MemberAccount": "123456789012",
      "EvaluationResults": [
        {
          "ComplianceStatus": "COMPLIANT",
          "ViolatorCount": 0,
          "EvaluationLimitExceeded": false
        },
        {
          "ComplianceStatus": "NON_COMPLIANT",
```

```
        "ViolatorCount": 2,  
        "EvaluationLimitExceeded": false  
      }  
    ],  
    "LastUpdated": 1576283774.0,  
    "IssueInfoMap": {}  
  }  
]  
}
```

詳細については、AWS WAF [AWS 「ファイアウォールマネージャー、および AWS Shield Advanced デベロッパーガイド」の「ポリシーによるリソースコンプライアンスの表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListComplianceStatus](#)」の「」を参照してください。

list-member-accounts

次の例は、list-member-accounts を使用する方法を説明しています。

AWS CLI

組織内のメンバーアカウントを取得するには

次のlist-member-accounts例では、Firewall Manager 管理者の組織内にあるすべてのメンバーアカウントを一覧表示します。

```
aws fms list-member-accounts
```

出力:

```
{  
  "MemberAccounts": [  
    "222222222222",  
    "333333333333",  
    "444444444444"  
  ]  
}
```

詳細については、[AWS 「Firewall Manager」](#)の AWS 「Firewall Manager」 および AWS 「Shield Advanced Developer Guide」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMemberAccounts](#)」の「」を参照してください。

list-policies

次のコード例は、list-policies を使用する方法を示しています。

AWS CLI

すべての Firewall Manager ポリシーを取得するには

次のlist-policies例では、アカウントのポリシーのリストを取得します。この例では、出力はリクエストごとに2つの結果に制限されています。各呼び出しNextTokenは、リストの次の結果セットを取得するために、次のlist-policies呼び出しで --starting-tokenパラメータの値として使用できる を返します。

```
aws fms list-policies \  
  --max-items 2
```

出力:

```
{  
  "PolicyList": [  
    {  
      "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "PolicyName": "test",  
      "ResourceType": "AWS::EC2::Instance",  
      "SecurityServiceType": "SECURITY_GROUPS_COMMON",  
      "RemediationEnabled": false  
    },  
    {  
      "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "PolicyId": "457c9b21-fc94-406c-ae63-21217395ba72",  
      "PolicyName": "test",  
      "ResourceType": "AWS::EC2::Instance",  
      "SecurityServiceType": "SECURITY_GROUPS_COMMON",  
      "RemediationEnabled": false  
    }  
  ],  
}
```



```
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

詳細については、[AWS 「、Firewall Manager、Shield Advanced デベロッパーガイド」の「Firewall Manager ポリシーの使用」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicies](#)」の「」を参照してください。

put-notification-channel

次のコード例は、put-notification-channel を使用する方法を示しています。

AWS CLI

Firewall Manager ログのSNSトピック情報を設定するには

次のput-notification-channel例では、SNSトピック情報を設定します。

```
aws fms put-notification-channel \  
  --sns-topic-arn arn:aws:sns:us-west-2:123456789012:us-west-2-fms \  
  --sns-role-name arn:aws:iam::123456789012:role/aws-service-role/  
fms.amazonaws.com/AWSServiceRoleForFMS
```

このコマンドでは何も出力されません。

詳細については、[「、Firewall Manager、Shield Advanced デベロッパーガイド」の「Amazon SNS Notifications と Amazon CloudWatch Alarms の設定」](#)を参照してください。AWS WAF
AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[PutNotificationChannel](#)」の「」を参照してください。

put-policy

次の例は、put-policy を使用する方法を説明しています。

AWS CLI

Firewall Manager ポリシーを作成するには

次のput-policy例では、Firewall Manager セキュリティグループポリシーを作成します。

```
aws fms put-policy \  
  --cli-input-json file://policy.json
```

policy.json の内容:

```
{  
  "Policy": {  
    "PolicyName": "test",  
    "SecurityServicePolicyData": {  
      "Type": "SECURITY_GROUPS_USAGE_AUDIT",  
      "ManagedServiceData": "{\"type\": \"SECURITY_GROUPS_USAGE_AUDIT\",  
\\\"deleteUnusedSecurityGroups\\\":false,\\\"coalesceRedundantSecurityGroups\\\":true}"  
    },  
    "ResourceType": "AWS::EC2::SecurityGroup",  
    "ResourceTags": [],  
    "ExcludeResourceTags": false,  
    "RemediationEnabled": false  
  },  
  "TagList": [  
    {  
      "Key": "foo",  
      "Value": "foo"  
    }  
  ]  
}
```

出力:

```
{  
  "Policy": {  
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "PolicyName": "test",  
    "PolicyUpdateToken": "1:X9QGexP7HASDlsFp+G31Iw==",  
    "SecurityServicePolicyData": {  
      "Type": "SECURITY_GROUPS_USAGE_AUDIT",  
      "ManagedServiceData": "{\"type\": \"SECURITY_GROUPS_USAGE_AUDIT\",  
\\\"deleteUnusedSecurityGroups\\\":false,\\\"coalesceRedundantSecurityGroups\\\":true,  
\\\"optionalDelayForUnusedInMinutes\\\":null}"  
    },  
    "ResourceType": "AWS::EC2::SecurityGroup",  
    "ResourceTags": [],  
    "ExcludeResourceTags": false,  
  }  
}
```

```
    "RemediationEnabled": false
  },
  "PolicyArn": "arn:aws:fms:us-west-2:123456789012:policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

詳細については、[AWS 「、Firewall Manager、および Shield Advanced デベロッパーガイド」の「Firewall Manager ポリシーの使用」](#)を参照してください。 AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[PutPolicy](#)」の「」を参照してください。

AWS FIS を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS FIS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-experiment-template

次のコード例は、create-experiment-template を使用する方法を示しています。

AWS CLI

実験テンプレートを作成するには

次のcreate-experiment-template例では、アカウントに AWS FIS実験テンプレートを作成します。

```
aws fis create-experiment-template \
```

```
--cli-input-json file://myfile.json
```

myfile.json の内容:

```
{
  "description": "experimentTemplate",
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"
    }
  ],
  "targets": {
    "Instances-Target-1": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": [
        "arn:aws:ec2:us-west-2:123456789012:instance/i-12a3b4c56d78e9012"
      ],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "reboot": {
      "actionId": "aws:ec2:reboot-instances",
      "description": "reboot",
      "parameters": {},
      "targets": {
        "Instances": "Instances-Target-1"
      }
    }
  },
  "roleArn": "arn:aws:iam::123456789012:role/myRole"
}
```

出力:

```
{
  "experimentTemplate": {
    "id": "ABCDE1fgHIJkLmNop",
    "description": "experimentTemplate",
    "targets": {
      "Instances-Target-1": {
        "resourceType": "aws:ec2:instance",
```

```
        "resourceArns": [
            "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
        ],
        "selectionMode": "ALL"
    }
},
"actions": {
    "reboot": {
        "actionId": "aws:ec2:reboot-instances",
        "description": "reboot",
        "parameters": {},
        "targets": {
            "Instances": "Instances-Target-1"
        }
    }
},
"stopConditions": [
    {
        "source": "aws:cloudwatch:alarm",
        "value": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:alarmName"
    }
],
"creationTime": 1616434850.659,
"lastUpdateTime": 1616434850.659,
"roleArn": "arn:aws:iam::123456789012:role/myRole",
"tags": {}
}
}
```

詳細については、AWS 「Fault Injection Simulator ユーザーガイド」の [「実験テンプレートを作成する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateExperimentTemplate](#)」の「」を参照してください。

delete-experiment-template

次の例は、delete-experiment-template を使用する方法を説明しています。

AWS CLI

実験テンプレートを削除するには

次のdelete-experiment-template例では、指定された実験テンプレートを削除します。

```
aws fis delete-experiment-template \  
  --id ABCDE1fgHIJkLmNop
```

出力:

```
{  
  "experimentTemplate": {  
    "id": "ABCDE1fgHIJkLmNop",  
    "description": "myExperimentTemplate",  
    "targets": {  
      "Instances-Target-1": {  
        "resourceType": "aws:ec2:instance",  
        "resourceArns": [  
          "arn:aws:ec2:us-west-2:123456789012:instance/  
i-12a3b4c56d78e9012"  
        ],  
        "selectionMode": "ALL"  
      }  
    },  
    "actions": {  
      "testaction": {  
        "actionId": "aws:ec2:stop-instances",  
        "parameters": {},  
        "targets": {  
          "Instances": "Instances-Target-1"  
        }  
      }  
    },  
    "stopConditions": [  
      {  
        "source": "none"  
      }  
    ],  
    "creationTime": 1616017191.124,  
    "lastUpdateTime": 1616017859.607,  
    "roleArn": "arn:aws:iam::123456789012:role/FISRole"  
  }  
}
```

詳細については、AWS「Fault Injection Simulator ユーザーガイド」の[「実験テンプレートの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteExperimentTemplate](#)」の「」を参照してください。

get-action

次の例は、get-action を使用する方法を説明しています。

AWS CLI

アクションの詳細を取得するには

次のget-action例では、指定されたアクションの詳細を取得します。

```
aws fis get-action \  
  --id aws:ec2:stop-instances
```

出力:

```
{  
  "action": {  
    "id": "aws:ec2:stop-instances",  
    "description": "Stop the specified EC2 instances.",  
    "parameters": {  
      "startInstancesAfterDuration": {  
        "description": "The time to wait before restarting the instances  
(ISO 8601 duration).",  
        "required": false  
      }  
    },  
    "targets": {  
      "Instances": {  
        "resourceType": "aws:ec2:instance"  
      }  
    },  
    "tags": {}  
  }  
}
```

詳細については、「Fault Injection Simulator ユーザーガイド」の「[アクション](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAction](#)」の「」を参照してください。

get-experiment-template

次の例は、get-experiment-template を使用する方法を説明しています。

AWS CLI

実験テンプレートの詳細を取得するには

次のget-experiment-template例では、指定された実験テンプレートの詳細を取得します。

```
aws fis get-experiment-template \  
  --id ABCDE1fgHIJkLmNop
```

出力:

```
{  
  "experimentTemplate": {  
    "id": "ABCDE1fgHIJkLmNop",  
    "description": "myExperimentTemplate",  
    "targets": {  
      "Instances-Target-1": {  
        "resourceType": "aws:ec2:instance",  
        "resourceArns": [  
          "arn:aws:ec2:us-west-2:123456789012:instance/  
i-12a3b4c56d78e9012"  
        ],  
        "selectionMode": "ALL"  
      }  
    },  
    "actions": {  
      "testaction": {  
        "actionId": "aws:ec2:stop-instances",  
        "parameters": {},  
        "targets": {  
          "Instances": "Instances-Target-1"  
        }  
      }  
    },  
    "stopConditions": [  
      {
```



```
        "source": "none"
      }
    ],
    "creationTime": 1616017191.124,
    "lastUpdateTime": 1616017331.51,
    "roleArn": "arn:aws:iam::123456789012:role/FISRole",
    "tags": {
      "key": "value"
    }
  }
}
```

詳細については、AWS 「Fault Injection Simulator ユーザーガイド」の「[実験テンプレート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetExperimentTemplate](#)」の「」を参照してください。

get-experiment

次の例は、get-experiment を使用する方法を説明しています。

AWS CLI

実験の詳細を取得するには

次のget-experiment例では、指定された実験の詳細を取得します。

```
aws fis get-experiment \  
  --id ABC12DeFGhI3jKLMNOP
```

出力:

```
{  
  "experiment": {  
    "id": "ABC12DeFGhI3jKLMNOP",  
    "experimentTemplateId": "ABCDE1fgHIJkLmNop",  
    "roleArn": "arn:aws:iam::123456789012:role/myRole",  
    "state": {  
      "status": "completed",  
      "reason": "Experiment completed."  
    },  
    "targets": {
```

```
    "Instances-Target-1": {
      "resourceType": "aws:ec2:instance",
      "resourceArns": [
        "arn:aws:ec2:us-west-2:123456789012:instance/
i-12a3b4c56d78e9012"
      ],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "reboot": {
      "actionId": "aws:ec2:reboot-instances",
      "parameters": {},
      "targets": {
        "Instances": "Instances-Target-1"
      },
      "state": {
        "status": "completed",
        "reason": "Action was completed."
      }
    }
  },
  "stopConditions": [
    {
      "source": "none"
    }
  ],
  "creationTime": 1616432509.662,
  "startTime": 1616432509.962,
  "endTime": 1616432522.307,
  "tags": {}
}
```

詳細については、[「Fault Injection Simulator ユーザーガイド」](#)の「[の実験 AWS FIS](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetExperiment](#)」の「」を参照してください。

list-actions

次の例は、list-actions を使用する方法を説明しています。

AWS CLI

アクションを一覧表示するには

次のlist-actions例では、使用可能なアクションを一覧表示します。

```
aws fis list-actions
```

出力:

```
{
  "actions": [
    {
      "id": "aws:ec2:reboot-instances",
      "description": "Reboot the specified EC2 instances.",
      "targets": {
        "Instances": {
          "resourceType": "aws:ec2:instance"
        }
      },
      "tags": {}
    },
    {
      "id": "aws:ec2:stop-instances",
      "description": "Stop the specified EC2 instances.",
      "targets": {
        "Instances": {
          "resourceType": "aws:ec2:instance"
        }
      },
      "tags": {}
    },
    {
      "id": "aws:ec2:terminate-instances",
      "description": "Terminate the specified EC2 instances.",
      "targets": {
        "Instances": {
          "resourceType": "aws:ec2:instance"
        }
      },
      "tags": {}
    },
    {
```

```
    "id": "aws:ecs:drain-container-instances",
    "description": "Drain percentage of underlying EC2 instances on an ECS
cluster.",
    "targets": {
      "Clusters": {
        "resourceType": "aws:ecs:cluster"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:eks:terminate-nodegroup-instances",
    "description": "Terminates a percentage of the underlying EC2 instances
in an EKS cluster.",
    "targets": {
      "Nodegroups": {
        "resourceType": "aws:eks:nodegroup"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:fis:inject-api-internal-error",
    "description": "Cause an AWS service to return internal error responses
for specific callers and operations.",
    "targets": {
      "Roles": {
        "resourceType": "aws:iam:role"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:fis:inject-api-throttle-error",
    "description": "Cause an AWS service to return throttled responses for
specific callers and operations.",
    "targets": {
      "Roles": {
        "resourceType": "aws:iam:role"
      }
    },
    "tags": {}
  },
  {
```

```
    "id": "aws:fis:inject-api-unavailable-error",
    "description": "Cause an AWS service to return unavailable error
responses for specific callers and operations.",
    "targets": {
      "Roles": {
        "resourceType": "aws:iam:role"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:fis:wait",
    "description": "Wait for the specified duration. Stop condition
monitoring will continue during this time.",
    "tags": {}
  },
  {
    "id": "aws:rds:failover-db-cluster",
    "description": "Failover a DB Cluster to one of the replicas.",
    "targets": {
      "Clusters": {
        "resourceType": "aws:rds:cluster"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:rds:reboot-db-instances",
    "description": "Reboot the specified DB instances.",
    "targets": {
      "DBInstances": {
        "resourceType": "aws:rds:db"
      }
    },
    "tags": {}
  },
  {
    "id": "aws:ssm:send-command",
    "description": "Run the specified SSM document.",
    "targets": {
      "Instances": {
        "resourceType": "aws:ec2:instance"
      }
    }
  },
}
```

```
        "tags": {}
      }
    ]
  }
```

詳細については、「Fault Injection Simulator ユーザーガイド」の「[アクション](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListActions](#)」の「」を参照してください。

list-experiment-templates

次の例は、list-experiment-templates を使用する方法を説明しています。

AWS CLI

実験テンプレートを一覧表示するには

次のlist-experiment-templates例では、AWS アカウントの実験テンプレートを一覧表示します。

```
aws fis list-experiment-templates
```

出力:

```
{
  "experimentTemplates": [
    {
      "id": "ABCDE1fgHIJkLmNop",
      "description": "myExperimentTemplate",
      "creationTime": 1616017191.124,
      "lastUpdateTime": 1616017191.124,
      "tags": {
        "key": "value"
      }
    }
  ]
}
```

詳細については、AWS 「Fault Injection Simulator ユーザーガイド」の「[実験テンプレート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListExperimentTemplates](#)」の「」を参照してください。

list-experiments

次のコード例は、list-experiments を使用する方法を示しています。

AWS CLI

実験を一覧表示するには

次のlist-experiments例では、AWS アカウントの実験を一覧表示します。

```
aws fis list-experiments
```

出力:

```
{
  "experiments": [
    {
      "id": "ABCdeF1GHijKLM23N0",
      "experimentTemplateId": "ABCDE1fgHIJkLmNop",
      "state": {
        "status": "running",
        "reason": "Experiment is running."
      },
      "creationTime": 1616017341.197,
      "tags": {
        "key": "value"
      }
    }
  ]
}
```

詳細については、[「Fault Injection Simulator ユーザーガイド」](#)の「[実験](#)」を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListExperiments](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたリソースのタグを一覧表示します。

```
aws fis list-tags-for-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

出力:

```
{  
  "tags": {  
    "key1": "value1",  
    "key2": "value2"  
  }  
}
```

詳細については、AWS 「Fault Injection Simulator ユーザーガイド」の [「リソースにタグを付ける AWS FIS」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListTagsForResource](#)」の「」を参照してください。

start-experiment

次のコード例は、start-experiment を使用する方法を示しています。

AWS CLI

実験を開始するには

次のstart-experiment例では、指定された実験を開始します。

```
aws fis start-experiment \  
  --experiment-template-id ABCDE1fgHIJkLmNop
```


出力:

```
{
  "experiment": {
    "id": "ABC12DeFGhI3jKLMNOP",
    "experimentTemplateId": "ABCDE1fgHIJkLmNop",
    "roleArn": "arn:aws:iam::123456789012:role/myRole",
    "state": {
      "status": "initiating",
      "reason": "Experiment is initiating."
    },
    "targets": {
      "Instances-Target-1": {
        "resourceType": "aws:ec2:instance",
        "resourceArns": [
          "arn:aws:ec2:us-west-2:123456789012:instance/i-12a3b4c56d78e9012"
        ],
        "selectionMode": "ALL"
      }
    },
    "actions": {
      "reboot": {
        "actionId": "aws:ec2:reboot-instances",
        "parameters": {},
        "targets": {
          "Instances": "Instances-Target-1"
        },
        "state": {
          "status": "pending",
          "reason": "Initial state"
        }
      }
    },
    "stopConditions": [
      {
        "source": "none"
      }
    ],
    "creationTime": 1616432464.025,
    "startTime": 1616432464.374,
    "tags": {}
  }
}
```

```
}
```

詳細については、[「Fault Injection Simulator ユーザーガイド」の「の実験 AWS FIS」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[StartExperiment](#)」の「」を参照してください。

stop-experiment

次のコード例は、stop-experiment を使用する方法を示しています。

AWS CLI

実験を停止するには

次のstop-experiment例では、指定された実験の実行を停止します。

```
aws fis stop-experiment \  
  --id ABC12DeFGhI3jKLMNOP
```

出力:

```
{  
  "experiment": {  
    "id": "ABC12DeFGhI3jKLMNOP",  
    "experimentTemplateId": "ABCDE1fgHIJkLmNop",  
    "roleArn": "arn:aws:iam::123456789012:role/myRole",  
    "state": {  
      "status": "stopping",  
      "reason": "Stopping Experiment."  
    },  
    "targets": {  
      "Instances-Target-1": {  
        "resourceType": "aws:ec2:instance",  
        "resourceArns": [  
          "arn:aws:ec2:us-west-2:123456789012:instance/  
i-12a3b4c56d78e9012"  
        ],  
        "selectionMode": "ALL"  
      }  
    }  
  }  
}
```

```
    },
    "actions": {
      "reboot": {
        "actionId": "aws:ec2:reboot-instances",
        "parameters": {},
        "targets": {
          "Instances": "Instances-Target-1"
        },
        "startAfter": [
          "wait"
        ],
        "state": {
          "status": "pending",
          "reason": "Initial state."
        }
      },
      "wait": {
        "actionId": "aws:fis:wait",
        "parameters": {
          "duration": "PT5M"
        },
        "state": {
          "status": "running",
          "reason": ""
        }
      }
    },
    "stopConditions": [
      {
        "source": "none"
      }
    ],
    "creationTime": 1616432680.927,
    "startTime": 1616432681.177,
    "tags": {}
  }
}
```

詳細については、[「Fault Injection Simulator ユーザーガイド」の「の実験 AWS FIS」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[StopExperiment](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、指定されたリソースにタグを付けます。

```
aws fis tag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP \  
  --tags key1=value1,key2=value2
```

このコマンドでは何も出力されません。

詳細については、AWS 「Fault Injection Simulator ユーザーガイド」の [「リソースにタグを付ける AWS FIS」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースのタグを解除するには

次のuntag-resource例では、指定されたリソースからタグを削除します。

```
aws fis untag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

このコマンドでは何も出力されません。

詳細については、AWS 「Fault Injection Simulator ユーザーガイド」の [「リソースにタグを付ける AWS FIS」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-experiment-template

次の例は、update-experiment-template を使用する方法を説明しています。

AWS CLI

実験テンプレートを更新するには

次のupdate-experiment-template例では、指定された実験テンプレートの説明を更新します。

```
aws fis update-experiment-template \  
  --id ABCDE1fgHIJkLmNop \  
  ---description myExperimentTemplate
```

出力:

```
{  
  "experimentTemplate": {  
    "id": "ABCDE1fgHIJkLmNop",  
    "description": "myExperimentTemplate",  
    "targets": {  
      "Instances-Target-1": {  
        "resourceType": "aws:ec2:instance",  
        "resourceArns": [  
          "arn:aws:ec2:us-west-2:123456789012:instance/  
i-12a3b4c56d78e9012"  
        ],  
        "selectionMode": "ALL"  
      }  
    },  
    "actions": {  
      "testaction": {  
        "actionId": "aws:ec2:stop-instances",  
        "parameters": {},  
        "targets": {  
          "Instances": "Instances-Target-1"  
        }  
      }  
    }  
  }  
}
```

```
    },
    "stopConditions": [
      {
        "source": "none"
      }
    ],
    "creationTime": 1616017191.124,
    "lastUpdateTime": 1616017859.607,
    "roleArn": "arn:aws:iam::123456789012:role/FISRole",
    "tags": {
      "key": "value"
    }
  }
}
```

詳細については、AWS 「Fault Injection Simulator ユーザーガイド」の「[実験テンプレートを更新する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateExperimentTemplate](#)」の「」を参照してください。

を使用した Amazon GameLift の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています GameLift。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-build

次のコード例は、create-build を使用する方法を示しています。

AWS CLI

Example1: S3 バケット内のファイルからゲームビルドを作成するには

次のcreate-build例では、カスタムゲームビルドリソースを作成します。管理しているAWS アカウントの S3 ロケーションに保存されている zip ファイルを使用します。この例では、Amazon に S3 ロケーションへのアクセス GameLift 許可を付与するIAMロールを既に作成していることを前提としています。リクエストはオペレーティングシステムを指定しないため、新しいビルドリソースのデフォルトは WINDOWS_2012 です。

```
aws gamelift create-build \  
  --storage-location file://storage-loc.json \  
  --name MegaFrogRaceServer.NA \  
  --build-version 12345.678
```

storage-loc.json の内容:

```
{  
  "Bucket": "MegaFrogRaceServer_NA_build_files"  
  "Key": "MegaFrogRaceServer_build_123.zip"  
  "RoleArn": "arn:aws:iam::123456789012:role/gamelift"  
}
```

出力:

```
{  
  "Build": {  
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "CreationTime": 1496708916.18,  
    "Name": "MegaFrogRaceServer.NA",  
    "OperatingSystem": "WINDOWS_2012",  
    "SizeOnDisk": 479303,  
    "Status": "INITIALIZED",  
    "Version": "12345.678"  
  },  
  "StorageLocation": {  
    "Bucket": "MegaFrogRaceServer_NA_build_files",  
    "Key": "MegaFrogRaceServer_build_123.zip"  
  }  
}
```

```
}
```

Example2: にファイルを手動でアップロードするためのゲームビルドリソースを作成するには GameLift

次のcreate-build例では、新しいビルドリソースを作成します。また、Amazon S3 の GameLift 場所にゲームビルドを手動でアップロードできるストレージロケーションと一時的な認証情報も取得します。ビルドが正常にアップロードされると、GameLift サービスはビルドを検証し、新しいビルドのステータスを更新します。

```
aws gamelift create-build \  
  --name MegaFrogRaceServer.NA \  
  --build-version 12345.678 \  
  --operating-system AMAZON_LINUX
```

出力:

```
{  
  "Build": {  
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "CreationTime": 1496708916.18,  
    "Name": "MegaFrogRaceServer.NA",  
    "OperatingSystem": "AMAZON_LINUX",  
    "SizeOnDisk": 0,  
    "Status": "INITIALIZED",  
    "Version": "12345.678"  
  },  
  "StorageLocation": {  
    "Bucket": "gamelift-builds-us-west-2",  
    "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  },  
  "UploadCredentials": {  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",  
    "SessionToken": "AgoGb3JpZ2luENz...EXAMPLETOKEN=="  
  }  
}
```

詳細については、[「Amazon デベロッパーガイド」の「にカスタムサーバービルドをアップロード GameLiftする」](#)を参照してください。 GameLift

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBuild](#)」の「」を参照してください。

create-fleet

次のコード例は、create-fleet を使用する方法を示しています。

AWS CLI

例 1: 基本的な Linux フリートを作成するには

次のcreate-fleet例では、カスタムサーバービルドをホストするために、最小限の設定でオンデマンド Linux インスタンスのフリートを作成します。を使用して設定を完了できますupdate-fleet。

```
aws gamelift create-fleet \  
  --name MegaFrogRaceServer.NA.v2 \  
  --description 'Hosts for v2 North America' \  
  --build-id build-1111aaaa-22bb-33cc-44dd-5555eeee66ff \  
  --certificate-configuration 'CertificateType=GENERATED' \  
  --ec2-instance-type c4.large \  
  --fleet-type ON_DEMAND \  
  --runtime-configuration 'ServerProcesses=[{LaunchPath=/local/game/release-na/  
MegaFrogRace_Server.exe, ConcurrentExecutions=1}]'
```

出力:

```
{  
  "FleetAttributes": {  
    "BuildId": "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",  
    "CertificateConfiguration": {  
      "CertificateType": "GENERATED"  
    },  
    "CreationTime": 1496365885.44,  
    "Description": "Hosts for v2 North America",  
    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/  
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",  
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",  
    "FleetType": "ON_DEMAND",  
    "InstanceType": "c4.large",  
    "MetricGroups": ["default"],  
    "Name": "MegaFrogRace.NA.v2",
```

```

    "NewGameSessionProtectionPolicy": "NoProtection",
    "OperatingSystem": "AMAZON_LINUX",
    "ServerLaunchPath": "/local/game/release-na/MegaFrogRace_Server.exe",
    "Status": "NEW"
  }
}

```

例 2: 基本的な Windows フリートを作成するには

次の `create-fleet` 例では、カスタムサーバービルドをホストするために、最小限の設定でスポット Windows インスタンスのフリートを作成します。を使用して設定を完了できません `update-fleet`。

```

aws gamelift create-fleet \
  --name MegaFrogRace.NA.v2 \
  --description 'Hosts for v2 North America' \
  --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff \
  --certificate-configuration 'CertificateType=GENERATED' \
  --ec2-instance-type c4.large \
  --fleet-type SPOT \
  --runtime-configuration 'ServerProcesses=[{LaunchPath=C:\game  

\Bin64.Release.Dedicated\MegaFrogRace_Server.exe, ConcurrentExecutions=1}]'

```

出力:

```

{
  "FleetAttributes": {
    "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
    "CertificateConfiguration": {
      "CertificateType": "GENERATED"
    },
    "CreationTime": 1496365885.44,
    "Description": "Hosts for v2 North America",
    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/  

fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetType": "SPOT",
    "InstanceType": "c4.large",
    "MetricGroups": ["default"],
    "Name": "MegaFrogRace.NA.v2",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "OperatingSystem": "WINDOWS_2012",

```

```

        "ServerLaunchPath": "C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
        "Status": "NEW"
    }
}

```

例 3: 完全に設定されたフリートを作成するには

次のcreate-fleet例では、カスタムサーバービルド用のスポット Windows インスタンスのフリートを作成し、最も一般的に使用される設定を提供します。

```

aws gamelift create-fleet \
  --name MegaFrogRace.NA.v2 \
  --description 'Hosts for v2 North America' \
  --build-id build-2222aaaa-33bb-44cc-55dd-6666eeee77ff \
  --certificate-configuration 'CertificateType=GENERATED' \
  --ec2-instance-type c4.large \
  --ec2-inbound-permissions
'FromPort=33435,ToPort=33435,IpRange=10.24.34.0/23,Protocol=UDP' \
  --fleet-type SPOT \
  --new-game-session-protection-policy FullProtection \
  --runtime-configuration file://runtime-config.json \
  --metric-groups default \
  --instance-role-arn 'arn:aws:iam::444455556666:role/GameLiftS3Access'

```

runtime-config.json の内容:

```

GameSessionActivationTimeoutSeconds=300,
MaxConcurrentGameSessionActivations=2,
ServerProcesses=[
  {LaunchPath=C:\game\Bin64.Release.Dedicated\MegaFrogRace_Server.exe,Parameters=-
debug,ConcurrentExecutions=1},
  {LaunchPath=C:\game\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe,ConcurrentExecutions=1}]

```

出力:

```

{
  "FleetAttributes": {
    "InstanceRoleArn": "arn:aws:iam::444455556666:role/GameLiftS3Access",
    "Status": "NEW",
    "InstanceType": "c4.large",

```

```

    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "Description": "Hosts for v2 North America",
    "FleetType": "SPOT",
    "OperatingSystem": "WINDOWS_2012",
    "Name": "MegaFrogRace.NA.v2",
    "CreationTime": 1569309011.11,
    "MetricGroups": [
        "default"
    ],
    "BuildId": "build-2222aaaa-33bb-44cc-55dd-6666eeee77ff",
    "ServerLaunchParameters": "abc",
    "ServerLaunchPath": "C:\\game\\Bin64.Release.Dedicated\\
\\MegaFrogRace_Server.exe",
    "NewGameSessionProtectionPolicy": "FullProtection",
    "CertificateConfiguration": {
        "CertificateType": "GENERATED"
    }
}
}
}

```

例 4: リアルタイムサーバーフリートを作成するには

次のcreate-fleet例では、Amazon にアップロードされたリアルタイム設定スクリプトを使用してスポットインスタンスのフリートを作成します GameLift。すべてのリアルタイムサーバーは Linux マシンにデプロイされます。この例では、アップロードされた Realtime スクリプトに複数のスクリプトファイルが含まれ、というスクリプトファイルにある Init()関数があると仮定しますMainScript.js。図に示すように、このファイルはランタイム設定で起動スクリプトとして識別されます。

```

aws gamelift create-fleet \
  --name MegaFrogRace.NA.realtime \
  --description 'Mega Frog Race Realtime fleet' \
  --script-id script-1111aaaa-22bb-33cc-44dd-5555eeee66ff \
  --ec2-instance-type c4.large \
  --fleet-type SPOT \
  --certificate-configuration 'CertificateType=GENERATED' --runtime-configuration
'ServerProcesses=[{LaunchPath=/local/game/MainScript.js,Parameters=+map
Winter444, ConcurrentExecutions=5}]'

```

出力:

```
{
  "FleetAttributes": {
    "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "Status": "NEW",
    "CreationTime": 1569310745.212,
    "InstanceType": "c4.large",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "CertificateConfiguration": {
      "CertificateType": "GENERATED"
    },
    "Name": "MegaFrogRace.NA.realtime",
    "ScriptId": "script-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
    "FleetArn": "arn:aws:gamelift:us-west-2:444455556666:fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
    "FleetType": "SPOT",
    "MetricGroups": [
      "default"
    ],
    "Description": "Mega Frog Race Realtime fleet",
    "OperatingSystem": "AMAZON_LINUX"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFleet](#)」の「」を参照してください。

create-game-session-queue

次の例は、create-game-session-queue を使用する方法を説明しています。

AWS CLI

Example1: 順序付けられたゲームセッションキューを設定するには

次のcreate-game-session-queue例では、2つのリージョンに宛先を持つ新しいゲームセッションキューを作成します。また、10分待ってからゲームセッションリクエストがタイムアウトするようにキューを設定します。レイテンシーポリシーは定義されていないため、最初の送信先が一覧表示されたすべてのゲームセッションを配置 GameLift しようとします。

```
aws gamelift create-game-session-queue \
  --name MegaFrogRaceServer-NA \
```

```
--destinations file://destinations.json \  
--timeout-in-seconds 600
```

destinations.json の内容:

```
{  
  "Destinations": [  
    {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },  
    {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }  
  ]  
}
```

出力:

```
{  
  "GameSessionQueues": [  
    {  
      "Name": "MegaFrogRaceServer-NA",  
      "GameSessionQueueArn": "arn:aws:gamelift:us-west-2:123456789012:gamesessionqueue/MegaFrogRaceServer-NA",  
      "TimeoutInSeconds": 600,  
      "Destinations": [  
        {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},  
        {"DestinationArn": "arn:aws:gamelift:us-west-1::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}  
      ]  
    }  
  ]  
}
```

Example2: プレイヤーのレイテンシーポリシーを使用してゲームセッションキューを設定するには

次のcreate-game-session-queue例では、2つのプレイヤーレイテンシーポリシーを持つ新しいゲームセッションキューを作成します。最初のポリシーは、ゲームセッションプレイメントの最初の1分間に強制される100ミリ秒のレイテンシー上限を設定します。2番目のポリシーでは、プレイメントリクエストが3分でタイムアウトするまで、レイテンシーの上限が200ミリ秒に引き上げられます。

```
aws gamelift create-game-session-queue \  
  --name MegaFrogRaceServer-NA \  
  --destinations file://destinations.json \  
  --player-latency-policies file://latency-policies.json \  
  --timeout-in-seconds 180
```

destinations.json の内容:

```
{  
  "Destinations": [  
    { "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" },  
    { "DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222" }  
  ]  
}
```

latency-policies.json の内容:

```
{  
  "PlayerLatencyPolicies": [  
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},  
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":  
60}  
  ]  
}
```

出力:

```
{  
  "GameSessionQueue": {  
    "Name": "MegaFrogRaceServer-NA",  
    "GameSessionQueueArn": "arn:aws:gamelift:us-  
west-2:111122223333:gamesessionqueue/MegaFrogRaceServer-NA",  
    "TimeoutInSeconds": 600,  
    "PlayerLatencyPolicies": [  
      {  
        "MaximumIndividualPlayerLatencyMilliseconds": 100,  
        "PolicyDurationSeconds": 60  
      },  
      {
```

```
        "MaximumIndividualPlayerLatencyMilliseconds": 200
      }
    ]
    "Destinations": [
      {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"},
      {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/fleet-
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"}
    ],
  }
}
```

詳細については、「Amazon [デベロッパーガイド](#)」の「[キューを作成する](#)」を参照してください。 GameLift

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateGameSessionQueue](#)」の「」を参照してください。

delete-build

次の例は、delete-build を使用する方法を説明しています。

AWS CLI

カスタムゲームビルドを削除するには

次のdelete-build例では、Amazon GameLift アカウントからビルドを削除します。ビルドが削除された後は、それを使用して新しいフリートを作成することはできません。このオペレーションは元に戻すことはできません。

```
aws gamelift delete-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteBuild](#)」の「」を参照してください。

delete-fleet

次のコード例は、delete-fleet を使用する方法を示しています。

AWS CLI

使用されていないフリートを削除するには

次のdelete-fleet例では、ゼロインスタンスにスケールダウンされたフリートを削除します。フリート容量が 0 HTTP より大きい場合、リクエストは 400 エラーで失敗します。

```
aws gamelift delete-fleet \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、「Amazon GameLift デベロッパーガイド」の [GameLift 「フリートの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteFleet](#)」の「」を参照してください。

delete-game-session-queue

次のコード例は、delete-game-session-queue を使用する方法を示しています。

AWS CLI

ゲームセッションキューを削除するには

次のdelete-game-session-queue例では、指定されたゲームセッションキューを削除します。

```
aws gamelift delete-game-session-queue \  
  --name MegaFrogRace-NA
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteGameSessionQueue](#)」の「」を参照してください。

describe-build

次のコード例は、describe-build を使用する方法を示しています。

AWS CLI

カスタムゲームビルドに関する情報を取得するには

次のdescribe-build例では、ゲームサーバービルドリソースのプロパティを取得します。

```
aws gamelift describe-build \  
--build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "Build": {  
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "CreationTime": 1496708916.18,  
    "Name": "My_Game_Server_Build_One",  
    "OperatingSystem": "AMAZON_LINUX",  
    "SizeOnDisk": 1304924,  
    "Status": "READY",  
    "Version": "12345.678"  
  }  
}
```

詳細については、「[Amazon GameLift デベロッパーガイド](#)」の「[にカスタムサーバービルドをアップロード GameLiftする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeBuild](#)」の「」を参照してください。

describe-ec2-instance-limits

次の例は、describe-ec2-instance-limits を使用する方法を説明しています。

AWS CLI

EC2インスタンスタイプのサービス制限を取得するには

次のdescribe-ec2-instance-limits例では、現在のリージョンで指定されたインスタンスタイプで使用されている最大許容インスタンスと現在のEC2インスタンスを表示します。結果は、許可された 20 個のインスタンスのうち 5 個のみが使用されていることを示します。

```
aws gamelift describe-ec2-instance-limits \  
  --ec2-instance-type m5.large
```

出力:

```
{  
  "EC2InstanceLimits": [  
    {  
      "EC2InstanceType": "m5.large",  
      "CurrentInstances": 5,  
      "InstanceLimit": 20  
    }  
  ]  
}
```

詳細については、「Amazon デベロッパーガイド」の [「コンピューティングリソースの選択」](#) を参照してください。 GameLift

- API 詳細については、「コマンドリファレンス」の [DescribeEc2InstanceLimits](#) を参照してください。 AWS CLI

describe-fleet-attributes

次の例は、describe-fleet-attributes を使用する方法を説明しています。

AWS CLI

Example1: フリートのリストの属性を表示するには

次のdescribe-fleet-attributes例では、指定された2つのフリートのフリート属性を取得します。図に示すように、リクエストされたフリートは同じビルドでデプロイされます。1つはオンデマンドインスタンス用、もう1つはスポットインスタンス用で、設定には若干の違いがあります。

```
aws gamelift describe-fleet-attributes \  
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

出力:

```
{
```

```
"FleetAttributes": [
  {
    "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "FleetType": "ON_DEMAND",
    "InstanceType": "c4.large",
    "Description": "On-demand hosts for v2 North America",
    "Name": "MegaFrogRaceServer.NA.v2-od",
    "CreationTime": 1568836191.995,
    "Status": "ACTIVE",
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "ServerLaunchPath": "C:\\\\game\\\\MegaFrogRace_Server.exe",
    "ServerLaunchParameters": "+gamelift_start_server",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "OperatingSystem": "WINDOWS_2012",
    "MetricGroups": [
      "default"
    ],
    "CertificateConfiguration": {
      "CertificateType": "DISABLED"
    }
  },
  {
    "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "FleetType": "SPOT",
    "InstanceType": "c4.large",
    "Description": "On-demand hosts for v2 North America",
    "Name": "MegaFrogRaceServer.NA.v2-spot",
    "CreationTime": 1568838275.379,
    "Status": "ACTIVATING",
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "ServerLaunchPath": "C:\\\\game\\\\MegaFrogRace_Server.exe",
    "NewGameSessionProtectionPolicy": "NoProtection",
    "OperatingSystem": "WINDOWS_2012",
    "MetricGroups": [
      "default"
    ]
  }
],
```

```

        "CertificateConfiguration": {
            "CertificateType": "GENERATED"
        }
    ]
}

```

Example2: すべてのフリートの属性をリクエストするには

以下は、ステータスが のすべてのフリートのフリート属性describe-fleet-attributesを返します。この例では、ページ分割パラメータを使用して一度に 1 つのフリートを返す方法を示しています。

```

aws gamelift describe-fleet-attributes \
  --limit 1

```

出力:

```

{
  "FleetAttributes": [
    {
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "FleetArn": "arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "FleetType": "SPOT",
      "InstanceType": "c4.large",
      "Description": "On-demand hosts for v2 North America",
      "Name": "MegaFrogRaceServer.NA.v2-spot",
      "CreationTime": 1568838275.379,
      "Status": "ACTIVATING",
      "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "ServerLaunchPath": "C:\\\\game\\\\MegaFrogRace_Server.exe",
      "NewGameSessionProtectionPolicy": "NoProtection",
      "OperatingSystem": "WINDOWS_2012",
      "MetricGroups": [
        "default"
      ],
      "CertificateConfiguration": {
        "CertificateType": "GENERATED"
      }
    }
  ]
}

```

```

    ],
    "NextToken":
    "eyJhd3NBY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
  }

```

出力には、コマンドを 2 回呼び出すときに使用できる NextToken 値が含まれます。値を `--next-token` パラメータに渡して、出力を受け取る場所を指定します。次のコマンドは、出力で 2 番目の結果を返します。

```

aws gamelift describe-fleet-attributes \
  --limit 1 \
  --next-
token eyJhd3NBY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS

```

レスポンスに NextToken 値が含まれなくなるまで繰り返します。

詳細については、「Amazon GameLift デベロッパーガイド」の [GameLift 「フリートのセットアップ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeFleetAttributes](#)」の「」を参照してください。

describe-fleet-capacity

次の例は、`describe-fleet-capacity` を使用する方法を説明しています。

AWS CLI

フリートのリストのキャパシティステータスを表示するには

次の `describe-fleet-capacity` 例では、指定された 2 つのフリートの現在の容量を取得します。

```

aws gamelift describe-fleet-capacity \
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111 fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222

```

出力:

```

{
  "FleetCapacity": [

```

```
{
  "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "InstanceType": "c5.large",
  "InstanceCounts": {
    "DESIRED": 10,
    "MINIMUM": 1,
    "MAXIMUM": 20,
    "PENDING": 0,
    "ACTIVE": 10,
    "IDLE": 3,
    "TERMINATING": 0
  }
},
{
  "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "InstanceType": "c5.large",
  "InstanceCounts": {
    "DESIRED": 13,
    "MINIMUM": 1,
    "MAXIMUM": 20,
    "PENDING": 0,
    "ACTIVE": 15,
    "IDLE": 2,
    "TERMINATING": 2
  }
}
]
```

詳細については、「Amazon GameLift デベロッパーガイド」の[GameLift 「フリートのメトリクス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFleetCapacity](#)」の「」を参照してください。

describe-fleet-events

次の例は、describe-fleet-events を使用する方法を説明しています。

AWS CLI

指定された期間にイベントをリクエストするには

次のdescribe-fleet-events例では、指定された期間に発生したすべてのフリート関連イベントの詳細を表示します。

```
aws gamelift describe-fleet-events \  
  --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --start-time 1579647600 \  
  --end-time 1579649400 \  
  --limit 5
```

出力:

```
{  
  "Events": [  
    {  
      "EventId": "a37b6892-5d07-4d3b-8b47-80244ecf66b9",  
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "EventCode": "FLEET_STATE_ACTIVE",  
      "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed state to ACTIVE",  
      "EventTime": 1579649342.191  
    },  
    {  
      "EventId": "67da4ec9-92a3-4d95-886a-5d6772c24063",  
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "EventCode": "FLEET_STATE_ACTIVATING",  
      "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed state to ACTIVATING",  
      "EventTime": 1579649321.427  
    },  
    {  
      "EventId": "23813a46-a9e6-4a53-8847-f12e6a8381ac",  
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "EventCode": "FLEET_STATE_BUILDING",  
      "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed state to BUILDING",  
      "EventTime": 1579649321.243  
    },  
    {  
      "EventId": "3bf217d0-1d44-42f9-9202-433ed475d2e8",  
      "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "EventCode": "FLEET_STATE_VALIDATING",  
      "EventTime": 1579649321.243  
    }  
  ]  
}
```



```

    "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
state to VALIDATING",
    "EventTime": 1579649197.449
  },
  {
    "EventId": "2ecd0130-5986-44eb-99a7-62df27741084",
    "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "EventCode": "FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND",
    "Message": "Failed to find a valid path",
    "EventTime": 1569319075.839,
    "PreSignedLogUrl": "https://gamelift-event-logs-prod-
us-west-2.s3.us-west-2.amazonaws.com/logs/fleet-83422059-8329-42a2-
a4d6-c4444386a6f8/events/2ecd0130-5986-44eb-99a7-62df27741084/
FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEB8aCXVzLXdlc3QtMiJHMEUCIHV5K%2FLPx8h310D
%2FAvx0%2FZxsDy5XA3cJ0wPdu3T0eBa%2FAiEA1yovokcZYy%2FV4CWW6l26aFyiSH0
%2Bxz%2FBMAhEHYHMqNcQkQMImp%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F
%2FARAAGgw3NDEwNjE10TixNzEiDI8rsZtzLzlwEDQhXSrlAtl5Ae
%2Fgo6FCIzqXPbXfB0nSvFYqeDlriZarEpKqKrUt8mXQv9iqHResqCph9AKo49lwgSYTT2QoSxnrD7%2FUgv
%2BZm2pVuczvUkUA0fxc6s0GxpjIAzdIE%2F5P%2FB7B9M%2BVZ
%2F9KF82hbJi0HTE6Y7BjKsEgFCvk4UXILhfjtAn9iQ18%2F21ZTurAcJbm7Y5tuLF9SWSK3%2BEa7VX0cCK4D401sMj
%2FIaXoHkNvg0RVTa0hIqdvpADQlsSBNdqTXbjHTu6fETE9Y9Ky%2BiJK5KiUG
%2F59GjCpDcvS1FqKeLUEmKT7wysGmvjMc2n%2Fr
%2F9VxQfte7w9srXwLLAQuwhiXAAyI5ICMz5JvzjzQwTqD4CHTVKUUDwL
%2BRZzbuuqkJ0bZm102CkRGp%2B74RTAzLbWptVqZTIIfzctiCTmWxb
%2FmKyELRYsVLRwNJ%2BGGJ7%2BCrN0RC%2FjlgfLYIZyeAqjPgAu5HjgX
%2BM7jCo9M7wBTInAXK0FQuf9dvA84SuwX0JFp17LYGjrHMKv0qC3GfbTMrZ6kzeNV9awKCpXB2Gnx9z2KvI1JdqirW
%2F9C6%2B4jIZPME3jXmZcEHqqw5uvAVF7aeIavtUZU8pxpDIWT0YE4p3Kriy2AA7ziCRKtVfjv839InyLk8LUjsioW
%2BYUq8%2FDTL1Lxqj1S%2Fi04TI0Wo7ilAo%2FKKWWF4guuNDexj8E00ynSp1yImB
%2BZf2Fua3044W4eEXAMPLE33333&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170621T231808Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20170621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
  }
],
"NextToken":
"eyJhd3NBZ2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZE1kIjpw7InMi0iJidWlsZC01NWYxZTZmMS"
}

```

詳細については、「Amazon GameLift デベロッパーガイド」の「[デバッグ GameLift フリートの問題](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeFleetEvents](#)」の「」を参照してください。

describe-fleet-port-settings

次の例は、describe-fleet-port-settings を使用する方法を説明しています。

AWS CLI

フリートのインバウンド接続アクセス許可を表示するには

次のdescribe-fleet-port-settings例では、指定されたフリートの接続設定を取得します。

```
aws gamelift describe-fleet-port-settings \  
  --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111
```

出力:

```
{  
  "InboundPermissions": [  
    {  
      "FromPort": 33400,  
      "ToPort": 33500,  
      "IpRange": "0.0.0.0/0",  
      "Protocol": "UDP"  
    },  
    {  
      "FromPort": 1900,  
      "ToPort": 2000,  
      "IpRange": "0.0.0.0/0",  
      "Protocol": "TCP"  
    }  
  ]  
}
```

詳細については、「Amazon GameLift デベロッパーガイド」の [GameLift 「フリートのセットアップ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeFleetPortSettings](#)」の「」を参照してください。

describe-fleet-utilization

次のコード例は、describe-fleet-utilization を使用する方法を示しています。

AWS CLI

Example1: フリートのリストの使用状況データを表示するには

次のdescribe-fleet-utilization例では、指定された 1 つのフリートの現在の使用状況情報を取得します。

```
aws gamelift describe-fleet-utilization \  
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-  
  EXAMPLE11111
```

出力:

```
{  
  "FleetUtilization": [  
    {  
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ActiveServerProcessCount": 100,  
      "ActiveGameSessionCount": 62,  
      "CurrentPlayerSessionCount": 329,  
      "MaximumPlayerSessionCount": 1000  
    }  
  ]  
}
```

Example2: すべてのフリートの使用状況データをリクエストするには

以下は、ステータスが のすべてのフリートのフリート使用状況データdescribe-fleet-utilizationを返します。この例では、ページ分割パラメータを使用して、一度に 2 つのフリートのデータを返します。

```
aws gamelift describe-fleet-utilization \  
  --limit 2
```

出力:

```
{  
  "FleetUtilization": [  

```

```

    {
      "FleetId": "fleet-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
      "ActiveServerProcessCount": 100,
      "ActiveGameSessionCount": 13,
      "CurrentPlayerSessionCount": 98,
      "MaximumPlayerSessionCount": 1000
    },
    {
      "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
      "ActiveServerProcessCount": 100,
      "ActiveGameSessionCount": 62,
      "CurrentPlayerSessionCount": 329,
      "MaximumPlayerSessionCount": 1000
    }
  ],
  "NextToken":
  "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS"
}

```

コマンドを 2 回呼び出し、NextToken 値を引数として `--next-token` パラメータに渡して、次の 2 つの結果を確認します。

```

aws gamelift describe-fleet-utilization \
  --limit 2 \
  --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS

```

レスポンスに出力 NextToken の値が含まれなくなるまで繰り返します。

詳細については、「Amazon GameLift デベロッパーガイド」の [GameLift 「フリートのメトリクス」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeFleetUtilization](#)」の「」を参照してください。

describe-game-session-queues

次のコード例は、`describe-game-session-queues` を使用する方法を示しています。

AWS CLI

ゲームセッションキューを表示するには

次のdescribe-game-session-queues例では、指定された 2 つのキューのプロパティを取得します。

```
aws gamelift describe-game-session-queues \  
--names MegaFrogRace-NA MegaFrogRace-EU
```

出力:

```
{  
  "GameSessionQueues": [{  
    "Destinations": [{  
      "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
    },  
    {  
      "DestinationArn": "arn:aws:gamelift:us-west-2::fleet/fleet-  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"  
    }  
  ],  
  "Name": "MegaFrogRace-NA",  
  "TimeoutInSeconds": 600,  
  "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/  
MegaFrogRace-NA",  
  "PlayerLatencyPolicies": [{  
    "MaximumIndividualPlayerLatencyMilliseconds": 200  
  },  
  {  
    "MaximumIndividualPlayerLatencyMilliseconds": 100,  
    "PolicyDurationSeconds": 60  
  }  
  ],  
  "FilterConfiguration": {  
    "AllowedLocations": ["us-west-2", "ap-south-1", "us-east-1"]  
  },  
  "PriorityConfiguration": {  
    "PriorityOrder": ["LOCATION", "FLEET_TYPE", "DESTINATION"],  
    "LocationOrder": ["us-west-2", "ap-south-1", "us-east-1"]  
  }  
  },  
  {  
    "Destinations": [{  
      "DestinationArn": "arn:aws:gamelift:eu-west-3::fleet/fleet-  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"  
    }  
  ]  
}
```

```
    ]],  
    "Name": "MegaFrogRace-EU",  
    "TimeoutInSeconds": 600,  
    "GameSessionQueueArn": "arn:aws:gamelift:us-west-2::gamesessionqueue/  
MegaFrogRace-EU"  
  }  
]  
}
```

詳細については、「Amazon GameLift デベロッパーガイド」の「[マルチリージョンキューの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGameSessionQueues](#)」の「」を参照してください。

describe-runtime-configuration

次のコード例は、describe-runtime-configuration を使用する方法を示しています。

AWS CLI

フリートのランタイム設定をリクエストするには

次のdescribe-runtime-configuration例では、指定されたフリートの現在のランタイム設定の詳細を取得します。

```
aws gamelift describe-runtime-configuration \  
--fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "RuntimeConfiguration": {  
    "ServerProcesses": [  
      {  
        "LaunchPath": "C:\\game\\Bin64.Release.Dedicated  
\\MegaFrogRace_Server.exe",  
        "Parameters": "+gamelift_start_server",  
        "ConcurrentExecutions": 3  
      },  
      {
```

```

        "LaunchPath": "C:\\game\\Bin64.Release.Dedicated
\MegaFrogRace_Server.exe",
        "Parameters": "+gamelift_start_server +debug",
        "ConcurrentExecutions": 1
    }
],
"MaxConcurrentGameSessionActivations": 2147483647,
"GameSessionActivationTimeoutSeconds": 300
}
}

```

詳細については、[「Amazon デベロッパーガイド」の「フリートで複数のプロセスを実行する」](#)を参照してください。 GameLift

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRuntimeConfiguration](#)」の「」を参照してください。

list-builds

次の例は、list-builds を使用する方法を説明しています。

AWS CLI

Example1: カスタムゲームビルドのリストを取得するには

次のlist-builds例では、現在のリージョンのすべてのゲームサーバービルドのプロパティを取得します。サンプルリクエストは、ページ分割パラメータ Limitと を使用してNextTokenシーケンシャルセットで結果を取得する方法を示しています。最初のコマンドは、最初の 2 つのビルドを取得します。利用可能な結果が 3 つ以上あるため、レスポンスにはより多くの結果が利用可能であることを示す NextToken が含まれます。

```
aws gamelift list-builds \
  --limit 2
```

出力:

```
{
  "Builds": [
    {
      "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",

```

```

        "CreationTime": 1495664528.723,
        "Name": "My_Game_Server_Build_One",
        "OperatingSystem": "WINDOWS_2012",
        "SizeOnDisk": 8567781,
        "Status": "READY",
        "Version": "12345.678"
    },
    {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "CreationTime": 1495528748.555,
        "Name": "My_Game_Server_Build_Two",
        "OperatingSystem": "AMAZON_LINUX_2",
        "SizeOnDisk": 8567781,
        "Status": "FAILED",
        "Version": "23456.789"
    }
],
"NextToken":
"eyJhd3NBWY2NvdW50SWQlOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS"
}

```

その後、次のように `--next-token` パラメータを使用して コマンドを再度呼び出して、次の 2 つのビルドを確認できます。

```

aws gamelift list-builds \
  --limit 2
  --next-
token eyJhd3NBWY2NvdW50SWQlOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS

```

レスポンスに `NextToken` 値が含まれなくなるまで繰り返します。

Example2: 失敗ステータスのカスタムゲームビルドのリストを取得するには

次の `list-builds` 例では、現在ステータスが `FAILED` の現在のリージョン内のすべてのゲームサーバービルドのプロパティを取得します。

```

aws gamelift list-builds \
  --status FAILED

```

出力:


```
{
  "Builds": [
    {
      "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "CreationTime": 1495528748.555,
      "Name": "My_Game_Server_Build_Two",
      "OperatingSystem": "AMAZON_LINUX_2",
      "SizeOnDisk": 8567781,
      "Status": "FAILED",
      "Version": "23456.789"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBuilds](#)」の「」を参照してください。

list-fleets

次のコード例は、list-fleets を使用する方法を示しています。

AWS CLI

Example1: リージョン内のすべてのフリートのリストを取得するには

次のlist-fleets例では、現在のリージョンのすべてのフリートIDsのフリートを表示します。この例では、ページ分割パラメータを使用してIDs、一度に2つのフリートを取得します。レスポンスには、取得する結果が多いことを示す next-token 属性が含まれます。

```
aws gamelift list-fleets \
  --limit 2
```

出力:

```
{
  "FleetIds": [
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
  ],
}
```

```

    "NextToken":
    "eyJhd3NBWY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
  }

```

次の 2 つの結果を取得するには、ここに示すように、次のコマンドで前のレスポンスの NextToken 値を渡すことができます。

```

aws gamelift list-fleets \
  --limit 2 \
  --next-
token eyJhd3NBWY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRLZj

```

Example2: 特定のビルドまたはスクリプトを持つリージョン内のすべてのフリートのリストを取得するには

次の list-builds 例では、指定されたゲームビルドでデプロイされたフリート IDs の を取得します。リアルタイムサーバーを使用している場合は、ビルド ID の代わりにスクリプト ID を指定できます。この例では制限パラメータを指定しないため、結果には最大 16 個のフリートを含めることができます IDs。

```

aws gamelift list-fleets \
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

出力:

```

{
  "FleetIds": [
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE44444"
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListFleets](#)」の「」を参照してください。

request-upload-credentials

次のコード例は、request-upload-credentials を使用する方法を示しています。

AWS CLI

ビルドをアップロードするためのアクセス認証情報を更新するには

次のcreate-build例では、Amazon S3 ロケーションに GameLift ビルドファイルをアップロードするための新しい有効なアクセス認証情報を取得します。認証情報の有効期間は限られています。ビルド ID は、元のCreateBuildリクエストへのレスポンスから取得されます。

```
aws gamelift request-upload-credentials \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "StorageLocation": {  
    "Bucket": "gamelift-builds-us-west-2",  
    "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  },  
  "UploadCredentials": {  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",  
    "SessionToken": "AgoGb3JpZ22luEnz...EXAMPLETOKEN=="  
  }  
}
```

詳細については、[「Amazon デベロッパーガイド」の「にカスタムサーバービルドをアップロード GameLiftする」](#)を参照してください。 GameLift

- API 詳細については、AWS CLI 「コマンドリファレンス[RequestUploadCredentials](#)」の「」を参照してください。

start-fleet-actions

次のコード例は、start-fleet-actions を使用する方法を示しています。

AWS CLI

フリートの自動スケーリングアクティビティを再起動するには

次のstart-fleet-actions例では、指定されたフリートに定義されているが、stop-fleet-actions`` を呼び出すことで停止されたすべてのスケーリングポリシーの使用を再開します。開始すると、スケーリングポリシーはすぐにそれぞれのメトリクスの追跡を開始します。

```
aws gamelift start-fleet-actions \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --actions AUTO_SCALING
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartFleetActions](#)」の「」を参照してください。

stop-fleet-actions

次の例は、stop-fleet-actions を使用する方法を説明しています。

AWS CLI

フリートの自動スケーリングアクティビティを停止するには

次のstop-fleet-actions例では、指定されたフリートに定義されているすべてのスケーリングポリシーの使用を停止します。ポリシーが一時停止された後も、手動で調整しない限り、フリート容量は同じアクティブなインスタンス数のままになります。

```
aws gamelift start-fleet-actions \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --actions AUTO_SCALING
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopFleetActions](#)」の「」を参照してください。

update-build

次の例は、update-build を使用する方法を説明しています。

AWS CLI

カスタムゲームビルドを更新するには

次のupdate-build例では、指定されたビルドリソースに関連付けられている名前とバージョン情報を変更します。返されたビルドオブジェクトは、変更が正常に行われたことを確認します。

```
aws gamelift update-build \  
  --build-id EXAMPLE11111 \  
  --name EXAMPLE11111 \  
  --version EXAMPLE11111
```

```
--build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
--name MegaFrogRaceServer.NA.east \  
--build-version 12345.east
```

出力:

```
{  
  "Build": {  
    "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "CreationTime": 1496708916.18,  
    "Name": "MegaFrogRaceServer.NA.east",  
    "OperatingSystem": "AMAZON_LINUX_2",  
    "SizeOnDisk": 1304924,  
    "Status": "READY",  
    "Version": "12345.east"  
  }  
}
```

詳細については、「Amazon デベロッパーガイド」の [「ビルドファイルの更新」](#) を参照してください。 GameLift

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateBuild](#)」の「」を参照してください。

update-game-session-queue

次の例は、update-game-session-queue を使用する方法を説明しています。

AWS CLI

ゲームセッションキュー設定を更新するには

次のupdate-game-session-queue例では、新しい送信先を追加し、既存のゲームセッションキューのプレイヤーレイテンシーポリシーを更新します。

```
aws gamelift update-game-session-queue \  
  --name MegaFrogRace-NA \  
  --destinations file://destinations.json \  
  --player-latency-policies file://latency-policies.json
```

destinations.json の内容:

```
{
  "Destinations": [
    {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
    {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
    {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
  ]
}
```

latency-policies.json の内容:

```
{
  "PlayerLatencyPolicies": [
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},
    {"MaximumIndividualPlayerLatencyMilliseconds": 150, "PolicyDurationSeconds":
120},
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
120}
  ]
}
```

出力:

```
{
  "GameSessionQueue": {
    "Destinations": [
      {"DestinationArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d"},
      {"DestinationArn": "arn:aws:gamelift:us-east-1::fleet/
fleet-5c6d3c4d-5e6f-7a8b-9c0d-1e2f3a4b5a2b"},
      {"DestinationArn": "arn:aws:gamelift:us-east-1::alias/
alias-11aa22bb-3c4d-5e6f-000a-1111aaaa22bb"}
    ],
    "GameSessionQueueArn": "arn:aws:gamelift:us-
west-2:111122223333:gamesessionqueue/MegaFrogRace-NA",
    "Name": "MegaFrogRace-NA",
    "TimeoutInSeconds": 600,
    "PlayerLatencyPolicies": [
```

```

        {"MaximumIndividualPlayerLatencyMilliseconds": 200},
        {"MaximumIndividualPlayerLatencyMilliseconds": 150,
"PolicyDurationSeconds": 120},
        {"MaximumIndividualPlayerLatencyMilliseconds": 100,
"PolicyDurationSeconds": 120}
    ]
}
}

```

詳細については、「[Amazon GameLift デベロッパーガイド](#)」の「[マルチリージョンキューの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス UpdateGameSessionQueue](#)」の「」を参照してください。

upload-build

次の例は、upload-build を使用する方法を説明しています。

AWS CLI

Example1: Linux ゲームサーバービルドをアップロードするには

次のupload-build例では、Linux ゲームサーバーのビルドファイルをファイルディレクトリから GameLift サービスにアップロードし、ビルドリソースを作成します。

```

aws gamelift upload-build \
  --name MegaFrogRaceServer.NA \
  --build-version 2.0.1 \
  --build-root ~/MegaFrogRace_Server/release-na \
  --operating-system AMAZON_LINUX_2 \
  --server-sdk-version 4.0.2

```

出力:

```

Uploading ~/MegaFrogRace_Server/release-na: 16.0 KiB / 74.6 KiB (21.45%)
Uploading ~/MegaFrogRace_Server/release-na: 32.0 KiB / 74.6 KiB (42.89%)
Uploading ~/MegaFrogRace_Server/release-na: 48.0 KiB / 74.6 KiB (64.34%)
Uploading ~/MegaFrogRace_Server/release-na: 64.0 KiB / 74.6 KiB (85.79%)
Uploading ~/MegaFrogRace_Server/release-na: 74.6 KiB / 74.6 KiB (100.00%)
Successfully uploaded ~/MegaFrogRace_Server/release-na to AWS GameLift
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Example2: Windows ゲームサーバービルドをアップロードするには

次のupload-build例では、Windows ゲームサーバーのビルドファイルを ディレクトリから GameLift サービスにアップロードし、ビルドレコードを作成します。

```
aws gamelift upload-build \  
  --name MegaFrogRaceServer.NA \  
  --build-version 2.0.1 \  
  --build-root C:\MegaFrogRace_Server\release-na \  
  --operating-system WINDOWS_2012 \  
  --server-sdk-version 4.0.2
```

出力:

```
Uploading C:\MegaFrogRace_Server\release-na: 16.0 KiB / 74.6 KiB (21.45%)  
Uploading C:\MegaFrogRace_Server\release-na: 32.0 KiB / 74.6 KiB (42.89%)  
Uploading C:\MegaFrogRace_Server\release-na: 48.0 KiB / 74.6 KiB (64.34%)  
Uploading C:\MegaFrogRace_Server\release-na: 64.0 KiB / 74.6 KiB (85.79%)  
Uploading C:\MegaFrogRace_Server\release-na: 74.6 KiB / 74.6 KiB (100.00%)  
Successfully uploaded C:\MegaFrogRace_Server\release-na to AWS GameLift  
Build ID: build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

詳細については、「[Amazon GameLift デベロッパーガイド](#)」の「[にカスタムサーバービルドをアップロード GameLiftする](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUploadBuild](#)」の「」を参照してください。

を使用した Global Accelerator の例 AWS CLI

次のコード例は、Global Accelerator AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-custom-routing-endpoints

次の例は、add-custom-routing-endpoints を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレーターのエンドポイントグループにVPCサブネットエンドポイントを追加するには

次のadd-custom-routing-endpoints例では、カスタムルーティングアクセラレーターのエンドポイントグループにVPCサブネットエンドポイントを追加します。

```
aws globalaccelerator add-custom-routing-endpoints \  
  --endpoint-group-  
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd \  
  --endpoint-configurations "EndpointId=subnet-1234567890abcdef0"
```

出力:

```
{  
  "EndpointDescriptions": [  
    {  
      "EndpointId": "subnet-1234567890abcdef0"  
    }  
  ],  
  
  "EndpointGroupArn": "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd"  
}
```

詳細については、[VPC AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのサブネットエンドポイントAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddCustomRoutingEndpoints](#)」の「」を参照してください。

advertise-byoip-cidr

次の例は、advertise-byoip-cidr を使用する方法を説明しています。

AWS CLI

アドレス範囲をアドバタイズするには

次のadvertise-byoip-cidr例では、AWS リソースで使用するようプロビジョニングしたアドレス範囲をアドバタイズ AWS するようリクエストしています。

```
aws globalaccelerator advertise-byoip-cidr \  
  --cidr 198.51.100.0/24
```

出力:

```
{  
  "ByoipCidr": {  
    "Cidr": "198.51.100.0/24",  
    "State": "PENDING_ADVERTISING"  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator での独自の IP アドレスの持ち込みAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AdvertiseByoipCidr](#)」の「」を参照してください。

allow-custom-routing-traffic

次のコード例は、allow-custom-routing-traffic を使用する方法を示しています。

AWS CLI

カスタムルーティングアクセラレーターのVPCサブネット内の特定の Amazon EC2インスタンスの送信先へのトラフィックを許可するには

次のallow-custom-routing-traffic例では、カスタムルーティングアクセラレーターでトラフィックを受信できるVPCサブネットエンドポイントの特定の Amazon EC2インスタンス (送信先) IP アドレスとポートへのトラフィックが許可されていることを指定します。

```
aws globalaccelerator allow-custom-routing-traffic \  
  --endpoint-group-  
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab8888example \  
  --endpoint-id subnet-abcd123example \  
  --destination-addresses "172.31.200.6" "172.31.200.7" \  
  --destination-ports 80 81
```

このコマンドでは何も出力されません。

詳細については、[VPC AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのサブネットエンドポイントAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AllowCustomRoutingTraffic](#)」の「」を参照してください。

create-accelerator

次の例は、create-accelerator を使用する方法を説明しています。

AWS CLI

アクセラレータを作成するには

次のcreate-accelerator例では、2つの静的 IP アドレスを持つ 2 BYOIP つのタグを持つアクセラレータを作成します。アクセラレータを作成または更新するには、US-West-2 (Oregon)リージョンを指定する必要があります。

```
aws globalaccelerator create-accelerator \  
  --name ExampleAccelerator \  
  --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \  
  --ip-addresses 192.0.2.250 198.51.100.52
```

出力:

```
{  
  "Accelerator": {  
    "AcceleratorArn":  
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh",
```

```
    "IpAddressType": "IPV4",
    "Name": "ExampleAccelerator",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
          "192.0.2.250",
          "198.51.100.52"
        ],
        "IpFamily": "IPv4"
      }
    ],
    "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
    "CreatedTime": 1542394847.0,
    "LastModifiedTime": 1542394847.0
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のAWS アクセラレータ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAccelerator](#)」の「」を参照してください。

create-custom-routing-accelerator

次の例は、create-custom-routing-accelerator を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレータを作成するには

次のcreate-custom-routing-accelerator例では、タグNameと を使用してカスタムルーティングアクセラレータを作成しますProject。

```
aws globalaccelerator create-custom-routing-accelerator \
  --name ExampleCustomRoutingAccelerator \
  --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project" \
  --ip-addresses 192.0.2.250 198.51.100.52
```

出力:

```
{
  "Accelerator": {
    "AcceleratorArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
    "Name": "ExampleCustomRoutingAccelerator",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
          "192.0.2.250",
          "198.51.100.52"
        ],
        "IpFamily": "IPv4"
      }
    ],
    "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
    "CreatedTime": 1542394847.0,
    "LastModifiedTime": 1542394847.0
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレータAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomRoutingAccelerator](#)」の「」を参照してください。

create-custom-routing-endpoint-group

次のコード例は、create-custom-routing-endpoint-group を使用する方法を示しています。

AWS CLI

カスタムルーティングアクセラレーターのエンドポイントグループを作成するには

次のcreate-custom-routing-endpoint-group例では、カスタムルーティングアクセラレーターのエンドポイントグループを作成します。

```
aws globalaccelerator create-custom-routing-endpoint-group \
```

```
--listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \  
--endpoint-group-region us-east-2 \  
--destination-configurations "FromPort=80,ToPort=81,Protocols=TCP,UDP"
```

出力:

```
{  
  "EndpointGroup": {  
    "EndpointGroupArn":  
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd",  
    "EndpointGroupRegion": "us-east-2",  
    "DestinationDescriptions": [  
      {  
        "FromPort": 80,  
        "ToPort": 81,  
        "Protocols": [  
          "TCP",  
          "UDP"  
        ]  
      }  
    ],  
    "EndpointDescriptions": []  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのエンドポイントグループAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomRoutingEndpointGroup](#)」の「」を参照してください。

create-custom-routing-listener

次の例は、create-custom-routing-listener を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレーターのリスナーを作成するには

次のcreate-custom-routing-listener例では、カスタムルーティングアクセラレーターのポート範囲が 5000 ~ 10000 のリスナーを作成します。

```
aws globalaccelerator create-custom-routing-listener \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --port-ranges FromPort=5000,ToPort=10000
```

出力:

```
{  
  "Listener": {  
    "PortRange": [  
      "FromPort": 5000,  
      "ToPort": 10000  
    ],  
    "ListenerArn":  
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz"  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのリスナーAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomRoutingListener](#)」の「」を参照してください。

create-endpoint-group

次の例は、create-endpoint-group を使用する方法を説明しています。

AWS CLI

エンドポイントグループを作成するには

次のcreate-endpoint-group例では、1つのエンドポイントを持つエンドポイントグループを作成します。

```
aws globalaccelerator create-endpoint-group \  
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \  
  --endpoint-group-region us-east-1 \  
  --endpoint-configurations EndpointId=i-1234567890abcdef0,Weight=128
```

出力:

```
{
  "EndpointGroup": {
    "TrafficDialPercentage": 100.0,
    "EndpointDescriptions": [
      {
        "Weight": 128,
        "EndpointId": "i-1234567890abcdef0"
      }
    ],
    "EndpointGroupArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefggh/listener/0123vxyz/endpoint-group/098765zyxwvu",
    "EndpointGroupRegion": "us-east-1"
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のエンドポイントグループAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEndpointGroup](#)」の「」を参照してください。

create-listener

次のコード例は、create-listener を使用する方法を示しています。

AWS CLI

リスナーを作成するには

次のcreate-listener例では、2つのポートを持つリスナーを作成します。

```
aws globalaccelerator create-listener \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
  abcd-1234-abcd-1234abcdefggh \
  --port-ranges FromPort=80,ToPort=80 FromPort=81,ToPort=81 \
  --protocol TCP
```

出力:


```
{
  "Listener": {
    "PortRanges": [
      {
        "ToPort": 80,
        "FromPort": 80
      },
      {
        "ToPort": 81,
        "FromPort": 81
      }
    ],
    "ClientAffinity": "NONE",
    "Protocol": "TCP",
    "ListenerArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefgh/listener/0123vxyz"
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のリスナー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateListener](#)」の「」を参照してください。

deny-custom-routing-traffic

次のコード例は、deny-custom-routing-traffic を使用する方法を示しています。

AWS CLI

カスタムルーティングアクセラレーターでトラフィックを受信できない送信先アドレスを指定するには

次のdeny-custom-routing-traffic例では、カスタムルーティングアクセラレータのトラフィックを受信できないサブネットエンドポイントの送信先アドレスを指定します。複数の送信先アドレスを指定するには、アドレスをスペースで区切ります。通話が成功 deny-custom-routing-trafficしても応答はありません。

```
aws globalaccelerator deny-custom-routing-traffic \
```

```
--endpoint-group-  
arn "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab8888example" \  
--endpoint-id "subnet-abcd123example" \  
--destination-addresses "198.51.100.52"
```

このコマンドでは何も出力されません。

詳細については、[VPC AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのサブネットエンドポイントAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DenyCustomRoutingTraffic](#)」の「」を参照してください。

deprovision-byoip-cidr

次のコード例は、`deprovision-byoip-cidr` を使用する方法を示しています。

AWS CLI

アドレス範囲のプロビジョニングを解除するには

次の`deprovision-byoip-cidr`例では、AWS リソースで使用するようプロビジョニングした指定されたアドレス範囲を解放します。

```
aws globalaccelerator deprovision-byoip-cidr \  
--cidr "198.51.100.0/24"
```

出力:

```
{  
  "ByoipCidr": {  
    "Cidr": "198.51.100.0/24",  
    "State": "PENDING_DEPROVISIONING"  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator で独自の IP アドレスを取得するAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeprovisionByoipCidr](#)」の「」を参照してください。

describe-accelerator-attributes

次の例は、describe-accelerator-attributes を使用する方法を説明しています。

AWS CLI

アクセラレータの属性を記述するには

次のdescribe-accelerator-attributes例では、アクセラレータの属性の詳細を取得します。

```
aws globalaccelerator describe-accelerator-attributes \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

出力:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": true  
    "FlowLogsS3Bucket": flowlogs-abc  
    "FlowLogsS3Prefix": bucketprefix-abc  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のAWS アクセラレータ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAcceleratorAttributes](#)」の「」を参照してください。

describe-accelerator

次の例は、describe-accelerator を使用する方法を説明しています。

AWS CLI

アクセラレータを記述するには

次のdescribe-accelerator例では、指定されたアクセラレーターの詳細を取得します。

```
aws globalaccelerator describe-accelerator \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

出力:

```
{  
  "Accelerator": {  
    "AcceleratorArn":  
    "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh",  
    "IpAddressType": "IPV4",  
    "Name": "ExampleAccelerator",  
    "Enabled": true,  
    "Status": "IN_PROGRESS",  
    "IpSets": [  
      {  
        "IpAddresses": [  
          "192.0.2.250",  
          "198.51.100.52"  
        ],  
        "IpFamily": "IPv4"  
      }  
    ],  
    "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",  
    "CreatedTime": 1542394847,  
    "LastModifiedTime": 1542395013  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」のAWS 「Global Accelerator のアクセラレータ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccelerator](#)」の「」を参照してください。

describe-custom-routing-accelerator-attributes

次の例は、describe-custom-routing-accelerator-attributes を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレータの属性を記述するには

次のdescribe-custom-routing-accelerator-attributes例では、カスタムルーティングアクセラレータの属性について説明します。

```
aws globalaccelerator describe-custom-routing-accelerator-attributes \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

出力:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": false  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレータAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCustomRoutingAcceleratorAttributes](#)」の「」を参照してください。

describe-custom-routing-accelerator

次の例は、describe-custom-routing-accelerator を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレーターを記述するには

次のdescribe-custom-routing-accelerator例では、指定されたカスタムルーティングアクセラレーターの詳細を取得します。

```
aws globalaccelerator describe-custom-routing-accelerator \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

出力:

```
{
  "Accelerator": {
    "AcceleratorArn":
      "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
    "Name": "ExampleCustomRoutingAccelerator",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
          "192.0.2.250",
          "198.51.100.52"
        ],
        "IpFamily": "IPv4"
      }
    ],
    "DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
    "CreatedTime": 1542394847,
    "LastModifiedTime": 1542395013
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレータAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCustomRoutingAccelerator](#)」の「」を参照してください。

describe-custom-routing-endpoint-group

次のコード例は、describe-custom-routing-endpoint-group を使用する方法を示しています。

AWS CLI

カスタムルーティングアクセラレーターのエンドポイントグループを記述するには

次のdescribe-custom-routing-endpoint-group例では、カスタムルーティングアクセラレーターのエンドポイントグループについて説明します。

```
aws globalaccelerator describe-custom-routing-endpoint-group \
  --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab8888example
```

出力:

```
{
  "EndpointGroup": {
    "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab8888example",
    "EndpointGroupRegion": "us-east-2",
    "DestinationDescriptions": [
      {
        "FromPort": 5000,
        "ToPort": 10000,
        "Protocols": [
          "UDP"
        ]
      }
    ],
    "EndpointDescriptions": [
      {
        "EndpointId": "subnet-1234567890abcdef0"
      }
    ]
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのエンドポイントグループAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeCustomRoutingEndpointGroup](#)」の「」を参照してください。

describe-custom-routing-listener

次の例は、describe-custom-routing-listener を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレーターのリスナーを記述するには

次のdescribe-custom-routing-listener例では、カスタムルーティングアクセラレーターのリスナーについて説明します。

```
aws globalaccelerator describe-custom-routing-listener \  
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

出力:

```
{  
  "Listener": {  
    "PortRanges": [  
      "FromPort": 5000,  
      "ToPort": 10000  
    ],  
    "ListenerArn":  
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/abcdef1234"  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのリスナーAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCustomRoutingListener](#)」の「」を参照してください。

describe-endpoint-group

次の例は、describe-endpoint-group を使用する方法を説明しています。

AWS CLI

エンドポイントグループを記述するには

次のdescribe-endpoint-group例では、Amazon EC2インスタンス、ALBおよびNLBのエンドポイントを持つエンドポイントグループの詳細を取得しますNLB。


```
aws globalaccelerator describe-endpoint-group \  
  --endpoint-group-  
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/  
ab8888example
```

出力:

```
{  
  "EndpointGroup": {  
    "TrafficDialPercentage": 100.0,  
    "EndpointDescriptions": [  
      {  
        "Weight": 128,  
        "EndpointId": "i-1234567890abcdef0"  
      },  
      {  
        "Weight": 128,  
        "EndpointId": "arn:aws:elasticloadbalancing:us-  
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"  
      },  
      {  
        "Weight": 128,  
        "EndpointId": "arn:aws:elasticloadbalancing:us-  
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"  
      }  
    ],  
    "EndpointGroupArn":  
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-  
group/4321abcd-abcd-4321-abcd-4321abcdefg",  
    "EndpointGroupRegion": "us-east-1"  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のエンドポイントグループAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEndpointGroup](#)」の「」を参照してください。

describe-listener

次のコード例は、describe-listener を使用する方法を示しています。

AWS CLI

リスナーを記述するには

次のdescribe-listener例では、リスナーについて説明します。

```
aws globalaccelerator describe-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

出力:

```
{
  "Listener": {
    "ListenerArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
    "PortRanges": [
      {
        "FromPort": 80,
        "ToPort": 80
      }
    ],
    "Protocol": "TCP",
    "ClientAffinity": "NONE"
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のリスナー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeListener](#)」の「」を参照してください。

list-accelerators

次の例は、list-accelerators を使用する方法を説明しています。


```
    }
    {
      "Cidr": "203.0.113.25/24",
      "State": "READY"
    }
  ]
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator で独自の IP アドレスを取得するAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListByoipCidr](#)」の「」を参照してください。

list-custom-routing-accelerators

次のコード例は、`list-custom-routing-accelerators` を使用する方法を示しています。

AWS CLI

カスタムルーティングアクセラレーターを一覧表示するには

次の`list-custom-routing-accelerators`例では、AWS アカウントのカスタムルーティングアクセラレーターを一覧表示します。

```
aws globalaccelerator list-custom-routing-accelerators
```

出力:

```
{
  "Accelerators": [
    {
      "AcceleratorArn":
"arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
      "Name": "TestCustomRoutingAccelerator",
      "IpAddressType": "IPV4",
      "Enabled": true,
      "IpSets": [
        {
          "IpFamily": "IPv4",
          "IpAddresses": [
```


AWS CLI

カスタムルーティングアクセラレーターでリスナーのエンドポイントグループを一覧表示するには

次のlist-custom-routing-endpoint-groups例では、カスタムルーティングアクセラレーターのリスナーのエンドポイントグループを一覧表示します。

```
aws globalaccelerator list-custom-routing-endpoint-groups \  
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

出力:

```
{  
  "EndpointGroups": [  
    {  
      "EndpointGroupArn":  
        "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",  
      "EndpointGroupRegion": "eu-central-1",  
      "DestinationDescriptions": [  
        {  
          "FromPort": 80,  
          "ToPort": 80,  
          "Protocols": [  
            "TCP",  
            "UDP"  
          ]  
        }  
      ]  
      "EndpointDescriptions": [  
        {  
          "EndpointId": "subnet-abcd123example"  
        }  
      ]  
    }  
  ]  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのエンドポイントグループAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCustomRoutingEndpointGroups](#)」の「」を参照してください。

list-custom-routing-listeners

次のコード例は、list-custom-routing-listeners を使用する方法を示しています。

AWS CLI

カスタムルーティングアクセラレーターのリスナーを一覧表示するには

次のlist-custom-routing-listeners例では、カスタムルーティングアクセラレーターのリスナーを一覧表示します。

```
aws globalaccelerator list-custom-routing-listeners \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

出力:

```
{  
  "Listeners": [  
    {  
      "ListenerArn":  
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/abcdef1234",  
      "PortRanges": [  
        {  
          "FromPort": 5000,  
          "ToPort": 10000  
        }  
      ],  
      "Protocol": "TCP"  
    }  
  ]  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのリスナー-AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCustomRoutingListeners](#)」の「」を参照してください。

list-custom-routing-port-mappings-by-destination

次の例は、list-custom-routing-port-mappings-by-destination を使用する方法を説明しています。

AWS CLI

特定のカスタムルーティングアクセラレーターの送信先のポートマッピングを一覧表示するには

次のlist-custom-routing-port-mappings-by-destination例では、カスタムルーティングアクセラレーターの特定の宛先EC2サーバー (宛先アドレス) のポートマッピングを示します。

```
aws globalaccelerator list-custom-routing-port-mappings-by-destination \  
  --endpoint-id subnet-abcd123example \  
  --destination-address 198.51.100.52
```

出力:

```
{  
  "DestinationPortMappings": [  
    {  
      "AcceleratorArn":  
        "arn:aws:globalaccelerator::402092451327:accelerator/24ea29b8-  
d750-4489-8919-3095f3c4b0a7",  
      "AcceleratorSocketAddresses": [  
        {  
          "IpAddress": "192.0.2.250",  
          "Port": 65514  
        },  
        {  
          "IpAddress": "192.10.100.99",  
          "Port": 65514  
        }  
      ],  
      "EndpointGroupArn":  
        "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example",  
      "EndpointId": "subnet-abcd123example",  
      "EndpointGroupRegion": "us-west-2",  
      "DestinationSocketAddress": {  
        "IpAddress": "198.51.100.52",
```

```

        "Port": 80
      },
      "IpAddressType": "IPv4",
      "DestinationTrafficState": "ALLOW"
    }
  ]
}

```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator でのカスタムルーティングアクセラレータの仕組みAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListCustomRoutingPortMappingsByDestination](#)」の「」を参照してください。

list-custom-routing-port-mappings

次の例は、list-custom-routing-port-mappings を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレーターのポートマッピングを一覧表示するには

次のlist-custom-routing-port-mappings例は、カスタムルーティングアクセラレーターのポートマッピングの一部リストを示しています。

```

aws globalaccelerator list-custom-routing-port-mappings \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh

```

出力:

```

{
  "PortMappings": [
    {
      "AcceleratorPort": 40480,
      "EndpointGroupArn":
        "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
      "EndpointId": "subnet-1234567890abcdef0",
      "DestinationSocketAddress": {
        "IpAddress": "192.0.2.250",

```

```
        "Port": 80
      },
      "Protocols": [
        "TCP",
        "UDP"
      ],
      "DestinationTrafficState": "ALLOW"
    }
  {
    "AcceleratorPort": 40481,
    "EndpointGroupArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
    "EndpointId": "subnet-1234567890abcdef0",
    "DestinationSocketAddress": {
      "IpAddress": "192.0.2.251",
      "Port": 80
    },
    "Protocols": [
      "TCP",
      "UDP"
    ],
    "DestinationTrafficState": "ALLOW"
  }
]
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator でのカスタムルーティングアクセラレータの仕組みAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCustomRoutingPortMappings](#)」の「」を参照してください。

list-endpoint-groups

次の例は、list-endpoint-groups を使用する方法を説明しています。

AWS CLI

エンドポイントグループを一覧表示するには

次のlist-endpoint-groups例では、リスナーのエンドポイントグループを一覧表示します。このリスナーには2つのエンドポイントグループがあります。

```
aws globalaccelerator --region us-west-2 list-endpoint-groups \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

出力:

```
{
  "EndpointGroups": [
    {
      "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
      "EndpointGroupRegion": "eu-central-1",
      "EndpointDescriptions": [],
      "TrafficDialPercentage": 100.0,
      "HealthCheckPort": 80,
      "HealthCheckProtocol": "TCP",
      "HealthCheckIntervalSeconds": 30,
      "ThresholdCount": 3
    }
    {
      "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab99999example",
      "EndpointGroupRegion": "us-east-1",
      "EndpointDescriptions": [],
      "TrafficDialPercentage": 50.0,
      "HealthCheckPort": 80,
      "HealthCheckProtocol": "TCP",
      "HealthCheckIntervalSeconds": 30,
      "ThresholdCount": 3
    }
  ]
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のエンドポイントグループAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEndpointGroups](#)」の「」を参照してください。

list-listeners

次の例は、list-listeners を使用する方法を説明しています。

AWS CLI

リスナーを一覧表示するには

次のlist-listeners例では、アクセラレーターのリスナーを一覧表示します。

```
aws globalaccelerator list-listeners \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

出力:

```
{  
  "Listeners": [  
    {  
      "ListenerArn":  
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/abcdef1234",  
      "PortRanges": [  
        {  
          "FromPort": 80,  
          "ToPort": 80  
        }  
      ],  
      "Protocol": "TCP",  
      "ClientAffinity": "NONE"  
    }  
  ]  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のリスナー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListListeners](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

アクセラレータのタグを一覧表示するには

次のlist-tags-for-resource例では、特定のアクセラレータのタグを一覧表示します。

```
aws globalaccelerator list-tags-for-resource \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Project",  
      "Value": "A123456"  
    }  
  ]  
}
```

詳細については、[AWS「Global Accelerator デベロッパーガイド」の「Global Accelerator でのタグ付け」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

provision-byoip-cidr

次のコード例は、provision-byoip-cidr を使用する方法を示しています。

AWS CLI

アドレス範囲をプロビジョニングするには

次のprovision-byoip-cidr例では、AWS リソースで使用する指定されたアドレス範囲をプロビジョニングします。

```
aws globalaccelerator provision-byoip-cidr \  
  --cidr 192.0.2.250/24 \  
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

出力:

```
{
  "ByoipCidr": {
    "Cidr": "192.0.2.250/24",
    "State": "PENDING_PROVISIONING"
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator での独自の IP アドレスの取得AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ProvisionByoipCidr](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

アクセラレータにタグを付けるには

次のtag-resource例では、タグ名とプロジェクトをアクセラレータに追加し、それぞれに対応する値を追加します。

```
aws globalaccelerator tag-resource \
  --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \
  --tags Key="Name",Value="Example Name" Key="Project",Value="Example Project"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator でのタグ付け」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

アクセラレータからタグを削除するには

次の`untag-resource`例では、アクセラレータから名前とプロジェクトタグを削除します。

```
aws globalaccelerator untag-resource \  
  --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --tag-keys Key="Name" Key="Project"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator でのタグ付け」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-accelerator-attributes

次のコード例は、`update-accelerator-attributes` を使用する方法を示しています。

AWS CLI

アクセラレータの属性を更新するには

次の`update-accelerator-attributes`例では、フローログを有効にするためにアクセラレーターを更新します。アクセラレータ属性を作成または更新するには、US-West-2 (Oregon)リージョンを指定する必要があります。

```
aws globalaccelerator update-accelerator-attributes \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --flow-logs-enabled \  
  --flow-logs-s3-bucket flowlogs-abc \  
  --flow-logs-s3-prefix bucketprefix-abc
```

出力:

```
{  
  "AcceleratorAttributes": {
```



```
    "FlowLogsEnabled": true
    "FlowLogsS3Bucket": flowlogs-abc
    "FlowLogsS3Prefix": bucketprefix-abc
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のAWS アクセラレータ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAcceleratorAttributes](#)」の「」を参照してください。

update-accelerator

次のコード例は、update-accelerator を使用する方法を示しています。

AWS CLI

アクセラレーターを更新するには

次のupdate-accelerator例では、アクセラレータ名を に変更するようにアクセラレータを変更しますExampleAcceleratorNew。アクセラレータを作成または更新するには、US-West-2 (Oregon)リージョンを指定する必要があります。

```
aws globalaccelerator update-accelerator \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
  --name ExampleAcceleratorNew
```

出力:

```
{
  "Accelerator": {
    "AcceleratorArn":
"arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
    "Name": "ExampleAcceleratorNew",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
```

```

        "IpAddresses": [
            "192.0.2.250",
            "198.51.100.52"
        ],
        "IpFamily": "IPv4"
    }
],
"DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
"CreatedTime": 1232394847,
"LastModifiedTime": 1232395654
}
}

```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のAWS アクセラレータ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAccelerator](#)」の「」を参照してください。

update-custom-routing-accelerator-attributes

次の例は、update-custom-routing-accelerator-attributes を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレーターの属性を更新するには

次のupdate-custom-routing-accelerator-attributes例では、カスタムルーティングアクセラレータを更新してフローログを有効にします。

```

aws globalaccelerator update-custom-routing-accelerator-attributes \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \
  --flow-logs-enabled \
  --flow-logs-s3-bucket flowlogs-abc \
  --flow-logs-s3-prefix bucketprefix-abc

```

出力:

```

{
  "AcceleratorAttributes": {

```

```
    "FlowLogsEnabled": true
    "FlowLogsS3Bucket": flowlogs-abc
    "FlowLogsS3Prefix": bucketprefix-abc
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレータAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateCustomRoutingAcceleratorAttributes](#)」の「」を参照してください。

update-custom-routing-accelerator

次の例は、`update-custom-routing-accelerator` を使用する方法を説明しています。

AWS CLI

カスタムルーティングアクセラレーターを更新するには

次の`update-custom-routing-accelerator`例では、カスタムルーティングアクセラレーターを変更して、アクセラレーター名を変更します。

```
aws globalaccelerator --region us-west-2 update-custom-routing-accelerator \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
  --name ExampleCustomRoutingAcceleratorNew
```

出力:

```
{
  "Accelerator": {
    "AcceleratorArn":
      "arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh",
    "IpAddressType": "IPV4",
    "Name": "ExampleCustomRoutingAcceleratorNew",
    "Enabled": true,
    "Status": "IN_PROGRESS",
    "IpSets": [
      {
        "IpAddresses": [
```

```
        "192.0.2.250",
        "198.51.100.52"
    ],
    "IpFamily": "IPv4"
}
],
"DnsName": "a1234567890abcdef.awsglobalaccelerator.com",
"CreatedTime": 1232394847,
"LastModifiedTime": 1232395654
}
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレータAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCustomRoutingAccelerator](#)」の「」を参照してください。

update-custom-routing-listener

次のコード例は、update-custom-routing-listener を使用する方法を示しています。

AWS CLI

カスタムルーティングアクセラレーターのリスナーを更新するには

次のupdate-custom-routing-listener例では、リスナーを更新してポート範囲を変更します。

```
aws globalaccelerator update-custom-routing-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
  --port-ranges FromPort=10000,ToPort=20000
```

出力:

```
{
  "Listener": {
    "ListenerArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz
    "PortRanges": [
      {
```

```

        "FromPort": 10000,
        "ToPort": 20000
    }
],
"Protocol": "TCP"
}
}

```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のカスタムルーティングアクセラレーターのリスナーAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCustomRoutingListener](#)」の「」を参照してください。

update-endpoint-group

次の例は、update-endpoint-group を使用する方法を説明しています。

AWS CLI

エンドポイントグループを更新するには

次のupdate-endpoint-group例では、Elastic IP アドレス、、ALBの3つのエンドポイントをエンドポイントグループに追加しますNLB。

```

aws globalaccelerator update-endpoint-group \
  --endpoint-group-
  arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
  abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
  ab8888example \
  --endpoint-configurations \
    EndpointId=eipalloc-eip01234567890abc,Weight=128 \
    EndpointId=arn:aws:elasticloadbalancing:us-east-1:000123456789:loadbalancer/
  app/ALBTesting/alb01234567890xyz,Weight=128 \
    EndpointId=arn:aws:elasticloadbalancing:us-east-1:000123456789:loadbalancer/
  net/NLBTesting/alb01234567890qrs,Weight=128

```

出力:

```

{
  "EndpointGroup": {
    "TrafficDialPercentage": 100,
    "EndpointDescriptions": [

```

```
{
  "Weight": 128,
  "EndpointId": "eip01234567890abc"
},
{
  "Weight": 128,
  "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"
},
{
  "Weight": 128,
  "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"
}
],
"EndpointGroupArn":
"arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
"EndpointGroupRegion": "us-east-1"
}
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のエンドポイントグループAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateEndpointGroup](#)」の「」を参照してください。

update-listener

次のコード例は、update-listener を使用する方法を示しています。

AWS CLI

リスナーを更新するには

次のupdate-listener例では、リスナーを更新してポートを 100 に変更します。

```
aws globalaccelerator update-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
  --port-ranges FromPort=100, ToPort=100
```

出力:

```
{
  "Listener": {
    "ListenerArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefgh/listener/0123vxyz
    "PortRanges": [
      {
        "FromPort": 100,
        "ToPort": 100
      }
    ],
    "Protocol": "TCP",
    "ClientAffinity": "NONE"
  }
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator のリスナー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateListener](#)」の「」を参照してください。

withdraw-byoip-cidr

次の例は、`withdraw-byoip-cidr` を使用する方法を説明しています。

AWS CLI

アドレス範囲を取り消すには

次の`withdraw-byoip-cidr`例では、以前に AWS リソースで使用するためにアドバタイズした AWS Global Accelerator からアドレス範囲を取り消します。

```
aws globalaccelerator withdraw-byoip-cidr \
  --cidr 192.0.2.250/24
```

出力:

```
{
  "ByoipCidr": {
```

```
    "Cidr": "192.0.2.250/24",  
    "State": "PENDING_WITHDRAWING"  
  }  
}
```

詳細については、[AWS 「Global Accelerator デベロッパーガイド」の「Global Accelerator で独自の IP アドレスを取得するAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[WithdrawByoipCidr](#)」の「」を参照してください。

AWS Glue を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Glue。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-stop-job-run

次のコード例は、batch-stop-job-run を使用する方法を示しています。

AWS CLI

ジョブの実行を停止するには

次のbatch-stop-job-run例では、ジョブの実行を停止します。

```
aws glue batch-stop-job-run \  
  --job-name "my-testing-job" \  
  --job-run-id jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f
```


出力:

```
{
  "SuccessfulSubmissions": [
    {
      "JobName": "my-testing-job",
      "JobRunId":
"jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f"
    }
  ],
  "Errors": [],
  "ResponseMetadata": {
    "RequestId": "66bd6b90-01db-44ab-95b9-6aeff0e73d88",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "date": "Fri, 16 Oct 2020 20:54:51 GMT",
      "content-type": "application/x-amz-json-1.1",
      "content-length": "148",
      "connection": "keep-alive",
      "x-amzn-requestid": "66bd6b90-01db-44ab-95b9-6aeff0e73d88"
    },
    "RetryAttempts": 0
  }
}
```

詳細については、「AWS Glue デベロッパーガイド」の「[ジョブの実行](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchStopJobRun](#)」の「」を参照してください。

create-connection

次の例は、create-connection を使用する方法を説明しています。

AWS CLI

AWS Glue データストアの接続を作成するには

次のcreate-connection例では、Kafka データストアの接続情報を提供する接続を AWS Glue Data Catalog に作成します。

```
aws glue create-connection \  
  --connection-input '{ \
```

```

    "Name": "conn-kafka-custom", \
    "Description": "kafka connection with ssl to custom kafka", \
    "ConnectionType": "KAFKA", \
    "ConnectionProperties": { \
        "KAFKA_BOOTSTRAP_SERVERS": "<Kafka-broker-server-url>:<SSL-Port>", \
        "KAFKA_SSL_ENABLED": "true", \
        "KAFKA_CUSTOM_CERT": "s3://bucket/prefix/cert-file.pem" \
    }, \
    "PhysicalConnectionRequirements": { \
        "SubnetId": "subnet-1234", \
        "SecurityGroupIdList": ["sg-1234"], \
        "AvailabilityZone": "us-east-1a" \
    } \
} \
--region us-east-1
--endpoint https://glue.us-east-1.amazonaws.com

```

このコマンドでは何も出力されません。

詳細については、[AWS Glue デベロッパーガイドの Glue データカタログの接続の定義](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConnection](#)」の「」を参照してください。

create-database

次のコード例は、create-database を使用する方法を示しています。

AWS CLI

データベースを作成するには

次のcreate-database例では、AWS Glue データカタログにデータベースを作成します。

```

aws glue create-database \
  --database-input "{\"Name\": \"tempdb\"}" \
  --profile my_profile \
  --endpoint https://glue.us-east-1.amazonaws.com

```

このコマンドでは何も出力されません。

詳細については、「AWS Glue デベロッパーガイド」の「[データカタログにデータベースを定義する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDatabase](#)」の「」を参照してください。

create-job

次の例は、create-job を使用する方法を説明しています。

AWS CLI

データを変換するジョブを作成するには

次の create-job 例では、S3 に保存されているスクリプトを実行するストリーミングジョブを作成します。

```
aws glue create-job \  
  --name my-testing-job \  
  --role AWSGlueServiceRoleDefault \  
  --command '{ \  
    "Name": "gluestreaming", \  
    "ScriptLocation": "s3://DOC-EXAMPLE-BUCKET/folder/" \  
  }' \  
  --region us-east-1 \  
  --output json \  
  --default-arguments '{ \  
    "--job-language":"scala", \  
    "--class":"GlueApp" \  
  }' \  
  --profile my-profile \  
  --endpoint https://glue.us-east-1.amazonaws.com
```

test_script.scala の内容:

```
import com.amazonaws.services.glue.ChoiceOption  
import com.amazonaws.services.glue.GlueContext  
import com.amazonaws.services.glue.MappingSpec  
import com.amazonaws.services.glue.ResolveSpec  
import com.amazonaws.services.glue.errors.CallSite  
import com.amazonaws.services.glue.util.GlueArgParser  
import com.amazonaws.services.glue.util.Job  
import com.amazonaws.services.glue.util.JsonOptions  
import org.apache.spark.SparkContext  
import scala.collection.JavaConverters._
```

```

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs,
Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    // @type: DataSource
    // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx
= "datasource0"]
    // @return: datasource0
    // @inputs: []
    val datasource0 = glueContext.getCatalogSource(database = "tempdb",
tableName = "s3-source", redshiftTmpDir = "", transformationContext =
"datasource0").getDynamicFrame()
    // @type: ApplyMapping
    // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
"status", "string")], transformation_ctx = "applymapping1"]
    // @return: applymapping1
    // @inputs: [frame = datasource0]
    val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
"int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
"int"), ("status", "string", "status", "string")), caseSensitive = false,
transformationContext = "applymapping1")
    // @type: SelectFields
    // @args: [paths = ["sensorid", "currenttemperature", "status"],
transformation_ctx = "selectfields2"]
    // @return: selectfields2
    // @inputs: [frame = applymapping1]
    val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
"currenttemperature", "status"), transformationContext = "selectfields2")
    // @type: ResolveChoice
    // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-
s3-sink", transformation_ctx = "resolvechoice3"]
    // @return: resolvechoice3
    // @inputs: [frame = selectfields2]
    val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
Some("my-s3-sink"), transformationContext = "resolvechoice3")
    // @type: DataSink

```

```

    // @args: [database = "tempdb", table_name = "my-s3-sink",
transformation_ctx = "datasink4"]
    // @return: datasink4
    // @inputs: [frame = resolvechoice3]
    val datasink4 = glueContext.getCatalogSink(database = "tempdb",
tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
"datasink4").writeDynamicFrame(resolvechoice3)
    Job.commit()
  }
}

```

出力:

```

{
  "Name": "my-testing-job"
}

```

詳細については、[AWS Glue デベロッパーガイドの「Glue でのジョブの作成」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateJob](#)」の「」を参照してください。

create-table

次のコード例は、create-table を使用する方法を示しています。

AWS CLI

例 1: Kinesis データストリームのテーブルを作成するには

次のcreate-table例では、Kinesis データストリームを記述するテーブルを AWS Glue Data Catalog に作成します。

```

aws glue create-table \
  --database-name tempdb \
  --table-input '{"Name":"test-kinesis-input", "StorageDescriptor":{ \
    "Columns":[ \
      {"Name":"sensorid", "Type":"int"}, \
      {"Name":"currenttemperature", "Type":"int"}, \
      {"Name":"status", "Type":"string"} \
    ], \

```

```

    "Location": "my-testing-stream", \
    "Parameters": { \
      "typeOfData": "kinesis", "streamName": "my-testing-stream", \
      "kinesisUrl": "https://kinesis.us-east-1.amazonaws.com" \
    }, \
    "SerdeInfo": { \
      "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe" \
    }, \
    "Parameters": { \
      "classification": "json" \
    } \
  } \
--profile my-profile \
--endpoint https://glue.us-east-1.amazonaws.com

```

このコマンドでは何も出力されません。

詳細については、[AWS Glue デベロッパーガイドの Glue データカタログの「テーブルの定義」](#)を参照してください。AWS

例 2: Kafka データストアのテーブルを作成するには

次の create-table 例では、Kafka データストアを記述するテーブルを AWS Glue Data Catalog に作成します。

```

aws glue create-table \
  --database-name tempdb \
  --table-input '{"Name": "test-kafka-input", "StorageDescriptor": { \
    "Columns": [ \
      {"Name": "sensorid", "Type": "int"}, \
      {"Name": "currenttemperature", "Type": "int"}, \
      {"Name": "status", "Type": "string"} \
    ], \
    "Location": "glue-topic", \
    "Parameters": { \
      "typeOfData": "kafka", "topicName": "glue-topic", \
      "connectionName": "my-kafka-connection" \
    }, \
    "SerdeInfo": { \
      "SerializationLibrary": "org.apache.hadoop.hive.serde2.OpenCSVSerde" \
    } \
  } \
  --separatorChar "," \

```

```
}' \  
--profile my-profile \  
--endpoint https://glue.us-east-1.amazonaws.com
```

このコマンドでは何も出力されません。

詳細については、[AWS Glue デベロッパーガイドの Glue データカタログの「テーブルの定義」](#)を参照してください。AWS

例 3: AWS S3 データストアのテーブルを作成するには

次のcreate-table例では、AWS Simple Storage Service (AWS S3) データストアを記述するテーブルを AWS Glue Data Catalog に作成します。

```
aws glue create-table \  
  --database-name tempdb \  
  --table-input '{"Name":"s3-output", "StorageDescriptor":{ \  
    "Columns":[ \  
      {"Name":"s1", "Type":"string"}, \  
      {"Name":"s2", "Type":"int"}, \  
      {"Name":"s3", "Type":"string"} \  
    ], \  
    "Location":"s3://bucket-path/", \  
    "SerdeInfo":{ \  
      "SerializationLibrary":"org.openx.data.jsonserde.JsonSerDe"} \  
    }, \  
    "Parameters":{ \  
      "classification":"json"} \  
  }' \  
  --profile my-profile \  
  --endpoint https://glue.us-east-1.amazonaws.com
```

このコマンドでは何も出力されません。

詳細については、[AWS Glue デベロッパーガイドの Glue データカタログの「テーブルの定義」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTable](#)」の「」を参照してください。

delete-job

次の例は、delete-job を使用する方法を説明しています。

AWS CLI

ジョブを削除するには

次の `delete-job` 例では、不要になったジョブを削除します。

```
aws glue delete-job \  
  --job-name my-testing-job
```

出力:

```
{  
  "JobName": "my-testing-job"  
}
```

詳細については、[AWS Glue デベロッパーガイドの「Glue コンソールでのジョブの操作」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteJob](#)」の「」を参照してください。

get-databases

次のコード例は、`get-databases` を使用する方法を示しています。

AWS CLI

AWS Glue Data Catalog の一部またはすべてのデータベースの定義を一覧表示するには

次の `get-databases` の例では、データカタログのデータベースに関する情報を返します。

```
aws glue get-databases
```

出力:

```
{  
  "DatabaseList": [  
    {  
      "Name": "default",  
      "Description": "Default Hive database",  
      "LocationUri": "file:/spark-warehouse",  
      "CreateTime": 1602084052.0,  
    }  
  ]  
}
```



```
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ],
    "CatalogId": "111122223333"
  },
  {
    "Name": "flights-db",
    "CreateTime": 1587072847.0,
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ],
    "CatalogId": "111122223333"
  },
  {
    "Name": "legislators",
    "CreateTime": 1601415625.0,
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ],
    "CatalogId": "111122223333"
  },
  {
    "Name": "tempdb",
```

```
    "CreateTime": 1601498566.0,
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ],
    "CatalogId": "111122223333"
  }
]
```

詳細については、「AWS Glue デベロッパーガイド」の「[データカタログにデータベースを定義する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDatabases](#)」の「」を参照してください。

get-job-run

次の例は、get-job-run を使用する方法を説明しています。

AWS CLI

ジョブの実行に関する情報を取得するには

次の get-job-run の例では、ジョブ実行に関する情報を取得します。

```
aws glue get-job-run \
  --job-name "Combine legislators data" \
  --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

出力:

```
{
  "JobRun": {
    "Id": "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
    "Attempt": 0,
  }
}
```

```
"JobName": "Combine legislators data",
"StartedOn": 1602873931.255,
"LastModifiedOn": 1602874075.985,
"CompletedOn": 1602874075.985,
"JobRunState": "SUCCEEDED",
"Arguments": {
  "--enable-continuous-cloudwatch-log": "true",
  "--enable-metrics": "",
  "--enable-spark-ui": "true",
  "--job-bookmark-option": "job-bookmark-enable",
  "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-east-1/
sparkHistoryLogs/"
},
"PredecessorRuns": [],
"AllocatedCapacity": 10,
"ExecutionTime": 117,
"Timeout": 2880,
"MaxCapacity": 10.0,
"WorkerType": "G.1X",
"NumberOfWorkers": 10,
"LogGroupName": "/aws-glue/jobs",
"GlueVersion": "2.0"
}
}
```

詳細については、「AWS Glue デベロッパーガイド」の「[ジョブの実行](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetJobRun](#)」の「」を参照してください。

get-job-runs

次のコード例は、get-job-runs を使用する方法を示しています。

AWS CLI

ジョブのすべてのジョブ実行に関する情報を取得するには

次の get-job-runs の例では、ジョブのジョブ実行に関する情報を取得します。

```
aws glue get-job-runs \
  --job-name "my-testing-job"
```

出力:

```
{
  "JobRuns": [
    {
      "Id":
"jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
      "Attempt": 0,
      "JobName": "my-testing-job",
      "StartedOn": 1602873931.255,
      "LastModifiedOn": 1602874075.985,
      "CompletedOn": 1602874075.985,
      "JobRunState": "SUCCEEDED",
      "Arguments": {
        "--enable-continuous-cloudwatch-log": "true",
        "--enable-metrics": "",
        "--enable-spark-ui": "true",
        "--job-bookmark-option": "job-bookmark-enable",
        "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-
east-1/sparkHistoryLogs/"
      },
      "PredecessorRuns": [],
      "AllocatedCapacity": 10,
      "ExecutionTime": 117,
      "Timeout": 2880,
      "MaxCapacity": 10.0,
      "WorkerType": "G.1X",
      "NumberOfWorkers": 10,
      "LogGroupName": "/aws-glue/jobs",
      "GlueVersion": "2.0"
    },
    {
      "Id":
"jr_03cc19ddb11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_2",
      "Attempt": 2,
      "PreviousRunId":
"jr_03cc19ddb11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
      "JobName": "my-testing-job",
      "StartedOn": 1602811168.496,
      "LastModifiedOn": 1602811282.39,
      "CompletedOn": 1602811282.39,
      "JobRunState": "FAILED",
      "ErrorMessage": "An error occurred while calling
o122.pyWriteDynamicFrame.
```

```

        Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
AccessDenied;
        Request ID: 021AAB703DB20A2D;
        S3 Extended Request ID: teZk24Y09TkXzBvMPG502L5VJBhe9DJuWA9/
TXtuG0qfByajkfL/Tlqt5JBGdEGpigAqzdMDM/U=)",
        "PredecessorRuns": [],
        "AllocatedCapacity": 10,
        "ExecutionTime": 110,
        "Timeout": 2880,
        "MaxCapacity": 10.0,
        "WorkerType": "G.1X",
        "NumberOfWorkers": 10,
        "LogGroupName": "/aws-glue/jobs",
        "GlueVersion": "2.0"
    },
    {
        "Id":
"jr_03cc19ddb11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
        "Attempt": 1,
        "PreviousRunId":
"jr_03cc19ddb11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f",
        "JobName": "my-testing-job",
        "StartedOn": 1602811020.518,
        "LastModifiedOn": 1602811138.364,
        "CompletedOn": 1602811138.364,
        "JobRunState": "FAILED",
        "ErrorMessage": "An error occurred while calling
o122.pyWriteDynamicFrame.
        Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
AccessDenied;
        Request ID: 2671D37856AE7ABB;
        S3 Extended Request ID: RLJCJw20brV
+PpC6Gp0RahyF2fp9f1B5SSb2bTGPNUSPVizLXR11PN3QZ1db+v1o9qRVktNYbW8=)",
        "PredecessorRuns": [],
        "AllocatedCapacity": 10,
        "ExecutionTime": 113,
        "Timeout": 2880,
        "MaxCapacity": 10.0,
        "WorkerType": "G.1X",
        "NumberOfWorkers": 10,
        "LogGroupName": "/aws-glue/jobs",
        "GlueVersion": "2.0"
    }
]

```

```
}
```

詳細については、「AWS Glue デベロッパーガイド」の「[ジョブの実行](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetJobRuns](#)」の「」を参照してください。

get-job

次のコード例は、get-job を使用する方法を示しています。

AWS CLI

ジョブに関する情報を取得するには

次の get-job の例では、ジョブに関する情報を取得します。

```
aws glue get-job \  
  --job-name my-testing-job
```

出力:

```
{  
  "Job": {  
    "Name": "my-testing-job",  
    "Role": "Glue_DefaultRole",  
    "CreatedOn": 1602805698.167,  
    "LastModifiedOn": 1602805698.167,  
    "ExecutionProperty": {  
      "MaxConcurrentRuns": 1  
    },  
    "Command": {  
      "Name": "gluestreaming",  
      "ScriptLocation": "s3://janetst-bucket-01/Scripts/test_script.scala",  
      "PythonVersion": "2"  
    },  
    "DefaultArguments": {  
      "--class": "GlueApp",  
      "--job-language": "scala"  
    },  
    "MaxRetries": 0,  
    "AllocatedCapacity": 10,  
  }  
}
```

```
    "MaxCapacity": 10.0,  
    "GlueVersion": "1.0"  
  }  
}
```

詳細については、「AWS Glue デベロッパーガイド」の「[ジョブ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetJob](#)」の「」を参照してください。

get-plan

次の例は、get-plan を使用する方法を説明しています。

AWS CLI

ソーステーブルからターゲットテーブルにデータをマッピングするために生成されたコードを取得するには

以下は、データソースからデータターゲットに列をマッピングするために生成されたコードget-planを取得します。

```
aws glue get-plan --mapping '[ \n  { \n    "SourcePath": "sensorid", \n    "SourceTable": "anything", \n    "SourceType": "int", \n    "TargetPath": "sensorid", \n    "TargetTable": "anything", \n    "TargetType": "int" \n  }, \n  { \n    "SourcePath": "currenttemperature", \n    "SourceTable": "anything", \n    "SourceType": "int", \n    "TargetPath": "currenttemperature", \n    "TargetTable": "anything", \n    "TargetType": "int" \n  }, \n  { \n    "SourcePath": "status", \n    "SourceTable": "anything", \n
```

```

    "SourceType": "string", \
    "TargetPath": "status", \
    "TargetTable": "anything", \
    "TargetType": "string" \
  }}' \
--source '{ \
    "DatabaseName": "tempdb", \
    "TableName": "s3-source" \
}' \
--sinks '[' \
    { \
        "DatabaseName": "tempdb", \
        "TableName": "my-s3-sink" \
    } \
  ]' \
--language "scala"
--endpoint https://glue.us-east-1.amazonaws.com
--output "text"

```

出力:

```

import com.amazonaws.services.glue.ChoiceOption
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.ResolveSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    // @type: DataSource
    // @args: [database = "tempdb", table_name = "s3-source", transformation_ctx =
    "datasource0"]
    // @return: datasource0
    // @inputs: []
  }
}

```



```

    val datasource0 = glueContext.getCatalogSource(database = "tempdb",
tableName = "s3-source", redshiftTmpDir = "", transformationContext =
"datasource0").getDynamicFrame()
    // @type: ApplyMapping
    // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
"status", "string")], transformation_ctx = "applymapping1"]
    // @return: applymapping1
    // @inputs: [frame = datasource0]
    val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
"int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
"int"), ("status", "string", "status", "string")), caseSensitive = false,
transformationContext = "applymapping1")
    // @type: SelectFields
    // @args: [paths = ["sensorid", "currenttemperature", "status"],
transformation_ctx = "selectfields2"]
    // @return: selectfields2
    // @inputs: [frame = applymapping1]
    val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
"currenttemperature", "status"), transformationContext = "selectfields2")
    // @type: ResolveChoice
    // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name = "my-s3-
sink", transformation_ctx = "resolvechoice3"]
    // @return: resolvechoice3
    // @inputs: [frame = selectfields2]
    val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
Some("my-s3-sink"), transformationContext = "resolvechoice3")
    // @type: DataSink
    // @args: [database = "tempdb", table_name = "my-s3-sink", transformation_ctx =
"datasink4"]
    // @return: datasink4
    // @inputs: [frame = resolvechoice3]
    val datasink4 = glueContext.getCatalogSink(database = "tempdb",
tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
"datasink4").writeDynamicFrame(resolvechoice3)
    Job.commit()
}
}

```

詳細については、[AWS Glue デベロッパーガイドの「Glue でのスクリプトの編集」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPlan](#)」の「」を参照してください。

get-tables

次の例は、get-tables を使用する方法を説明しています。

AWS CLI

指定されたデータベース内で、一部またはすべてのテーブルの定義を一覧表示するには

次の get-tables の例では、指定されたデータベース内のテーブルに関する情報を返します。

```
aws glue get-tables --database-name 'tempdb'
```

出力:

```
{
  "TableList": [
    {
      "Name": "my-s3-sink",
      "DatabaseName": "tempdb",
      "CreateTime": 1602730539.0,
      "UpdateTime": 1602730539.0,
      "Retention": 0,
      "StorageDescriptor": {
        "Columns": [
          {
            "Name": "sensorid",
            "Type": "int"
          },
          {
            "Name": "currenttemperature",
            "Type": "int"
          },
          {
            "Name": "status",
            "Type": "string"
          }
        ]
      },
      "Location": "s3://janetst-bucket-01/test-s3-output/",
      "Compressed": false,
      "NumberOfBuckets": 0,
    }
  ]
}
```

```
    "SerdeInfo": {
      "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
    },
    "SortColumns": [],
    "StoredAsSubDirectories": false
  },
  "Parameters": {
    "classification": "json"
  },
  "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
  "IsRegisteredWithLakeFormation": false,
  "CatalogId": "007436865787"
},
{
  "Name": "s3-source",
  "DatabaseName": "tempdb",
  "CreateTime": 1602730658.0,
  "UpdateTime": 1602730658.0,
  "Retention": 0,
  "StorageDescriptor": {
    "Columns": [
      {
        "Name": "sensorid",
        "Type": "int"
      },
      {
        "Name": "currenttemperature",
        "Type": "int"
      },
      {
        "Name": "status",
        "Type": "string"
      }
    ]
  },
  "Location": "s3://janetst-bucket-01/",
  "Compressed": false,
  "NumberOfBuckets": 0,
  "SortColumns": [],
  "StoredAsSubDirectories": false
},
  "Parameters": {
    "classification": "json"
  },
  "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
```

```
    "IsRegisteredWithLakeFormation": false,
    "CatalogId": "007436865787"
  },
  {
    "Name": "test-kinesis-input",
    "DatabaseName": "tempdb",
    "CreateTime": 1601507001.0,
    "UpdateTime": 1601507001.0,
    "Retention": 0,
    "StorageDescriptor": {
      "Columns": [
        {
          "Name": "sensorid",
          "Type": "int"
        },
        {
          "Name": "currenttemperature",
          "Type": "int"
        },
        {
          "Name": "status",
          "Type": "string"
        }
      ],
      "Location": "my-testing-stream",
      "Compressed": false,
      "NumberOfBuckets": 0,
      "SerdeInfo": {
        "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
      },
      "SortColumns": [],
      "Parameters": {
        "kinesisUrl": "https://kinesis.us-east-1.amazonaws.com",
        "streamName": "my-testing-stream",
        "typeOfData": "kinesis"
      },
      "StoredAsSubDirectories": false
    },
    "Parameters": {
      "classification": "json"
    },
    "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
    "IsRegisteredWithLakeFormation": false,
    "CatalogId": "007436865787"
  }
]
```

```
    }  
  ]  
}
```

詳細については、[AWS Glue デベロッパーガイドの Glue データカタログの表の定義](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTables](#)」の「」を参照してください。

start-crawler

次の例は、start-crawler を使用する方法を説明しています。

AWS CLI

クローラーを開始するには

次の start-crawler の例では、クローラーを開始します。

```
aws glue start-crawler --name my-crawler
```

出力:

```
None
```

詳細については、「AWS Glue デベロッパーガイド」の「[クローラーの定義](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartCrawler](#)」の「」を参照してください。

start-job-run

次のコード例は、start-job-run を使用する方法を示しています。

AWS CLI

ジョブの実行を開始するには

次の start-job-run の例ではジョブを開始します。

```
aws glue start-job-run \  
  --job-name my-job
```

出力:

```
{  
  "JobRunId":  
  "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"  
}
```

詳細については、「AWS Glue デベロッパーガイド」の「[ジョブの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartJobRun](#)」の「」を参照してください。

GuardDuty を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています GuardDuty。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-invitation

次の例は、accept-invitation を使用する方法を説明しています。

AWS CLI

現在のリージョンのメンバー GuardDuty アカウントになるための招待を受け入れるには

次のaccept-invitation例は、現在のリージョンのメンバー GuardDuty アカウントになるための招待を受け入れる方法を示しています。

```
aws guardduty accept-invitation \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --master-id 123456789111 \  
  --invitation-id d6b94fb03a66ff665f7db8764example
```

このコマンドでは何も出力されません。

詳細については、GuardDuty 「ユーザーガイド」の [「招待による GuardDuty アカウントの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AcceptInvitation](#)」の「」を参照してください。

archive-findings

次のコード例は、archive-findings を使用する方法を示しています。

AWS CLI

現在のリージョンで結果をアーカイブするには

この例では、現在のリージョンで結果をアーカイブする方法を示しています。

```
aws guardduty archive-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-ids d6b94fb03a66ff665f7db8764example 3eb970e0de00c16ec14e6910fexample
```

このコマンドでは何も出力されません。

詳細については、GuardDuty 「ユーザーガイド」の [「招待による GuardDuty アカウントの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ArchiveFindings](#)」の「」を参照してください。

create-detector

次のコード例は、create-detector を使用する方法を示しています。

AWS CLI

現在のリージョン GuardDuty で を有効にするには

この例では、現在のリージョン GuardDuty で を有効にする新しいディテクターを作成する方法を示しています。

```
aws guardduty create-detector \  
  --enable
```

出力:

```
{  
  "DetectorId": "b6b992d6d2f48e64bc59180bfexample"  
}
```

詳細については、「ユーザーガイド」の「[Amazon を有効にする GuardDuty](#)」を参照してください。 GuardDuty

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDetector](#)」の「」を参照してください。

create-filter

次のコード例は、create-filter を使用する方法を示しています。

AWS CLI

現在のリージョンの新しいフィルターを作成するには

この例では、特定のイメージから作成されたインスタンスのすべての portcan 検出結果に一致するフィルターを作成します。

```
aws guardduty create-filter \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --action ARCHIVE \  
  --name myFilter \  
  --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/  
Portscan"]}, "resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

出力:


```
{
  "Name": "myFilter"
}
```

詳細については、GuardDuty 「ユーザーガイド」の「[結果のフィルタリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFilter](#)」の「」を参照してください。

create-ip-set

次のコード例は、create-ip-set を使用する方法を示しています。

AWS CLI

信頼された IP セットを作成するには

次のcreate-ip-set例では、現在のリージョンで信頼された IP セットを作成してアクティブ化します。

```
aws guardduty create-ip-set \
  --detector-id 12abc34d567e8fa901bc2d34eexample \
  --name new-ip-set \
  --format TXT \
  --location s3://AWSDOC-EXAMPLE-BUCKET/customtrustlist.csv \
  --activate
```

出力:

```
{
  "IpSetId": "d4b94fc952d6912b8f3060768example"
}
```

詳細については、[「ユーザーガイド」の「信頼された IP リストと脅威リストの使用」](#)を参照してください。GuardDuty

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateIpSet](#)」の「」を参照してください。

create-members

次の例は、create-members を使用する方法を説明しています。

AWS CLI

新しいメンバーを現在のリージョンの GuardDuty マスターアカウントに関連付けるには。

この例では、現在のアカウントが管理するメンバーアカウントを GuardDuty マスターとして関連付ける方法を示しています。

```
aws guardduty create-members
  --detector-id b6b992d6d2f48e64bc59180bfexample \
  --account-details AccountId=111122223333,Email=first
+member@example.com AccountId=111111111111 ,Email=another+member@example.com
```

出力:

```
{
  "UnprocessedAccounts": []
}
```

詳細については、GuardDuty 「ユーザーガイド」の [「複数のアカウントの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateMembers](#)」の「」を参照してください。

create-publishing-destination

次のコード例は、create-publishing-destination を使用する方法を示しています。

AWS CLI

現在のリージョン GuardDuty の結果をエクスポートする発行先を作成するには。

この例では、GuardDuty 検出結果の発行先を作成する方法を示しています。

```
aws guardduty create-publishing-destination \
  --detector-id b6b992d6d2f48e64bc59180bfexample \
  --destination-type S3 \
```

```
--destination-  
properties DestinationArn=arn:aws:s3:::yourbucket,KmsKeyArn=arn:aws:kms:us-  
west-1:111122223333:key/84cee9c5-dea1-401a-ab6d-e1de7example
```

出力:

```
{  
  "DestinationId": "46b99823849e1bbc242dfbe3cexample"  
}
```

詳細については、GuardDuty 「ユーザーガイド」の「[検出結果のエクスポート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePublishingDestination](#)」の「」を参照してください。

create-sample-findings

次のコード例は、create-sample-findings を使用する方法を示しています。

AWS CLI

現在のリージョンでサンプル GuardDuty 検出結果を作成します。

この例では、指定されたタイプのサンプル検出結果を作成する方法を示します。

```
aws guardduty create-sample-findings \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --finding-types UnauthorizedAccess:EC2/TorClient UnauthorizedAccess:EC2/TorRelay
```

このコマンドでは何も出力されません。

詳細については、GuardDuty 「ユーザーガイド」の「[検出結果のサンプル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSampleFindings](#)」の「」を参照してください。

create-threat-intel-set

次の例は、create-threat-intel-set を使用する方法を説明しています。

AWS CLI

現在のリージョンに新しい脅威インテリジェンスセットを作成するには。

この例では、 に設定された脅威インテリジェンスをアップロード GuardDuty し、すぐにアクティブ化する方法を示しています。

```
aws guardduty create-threat-intel-set \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --name myThreatSet \  
  --format TXT \  
  --location s3://EXAMPLEBUCKET/threatlist.csv \  
  --activate
```

出力:

```
{  
  "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"  
}
```

詳細については、GuardDuty 「ユーザーガイド」の [「信頼された IP と脅威のリスト」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateThreatIntelSet](#)」の「」を参照してください。

decline-invitations

次のコード例は、decline-invitations を使用する方法を示しています。

AWS CLI

現在のリージョンの別のアカウントで Guardduty を管理する招待を拒否するには。

この例では、メンバーシップの招待を拒否する方法を示しています。

```
aws guardduty decline-invitations \  
  --account-ids 111122223333
```

出力:

```
{
```

```
"UnprocessedAccounts": []  
}
```

詳細については、GuardDuty 「ユーザーガイド」の [「招待による GuardDuty アカウントの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeclineInvitations](#)」の「」を参照してください。

delete-detector

次の例は、delete-detector を使用する方法を説明しています。

AWS CLI

現在のリージョンでディテクターを削除し GuardDuty、 を無効にするには。

この例では、ディテクターを削除する方法を示しています。成功すると、そのディテクターに関連付けられたリージョン GuardDuty で無効になります。

```
aws guardduty delete-detector \  
  --detector-id b6b992d6d2f48e64bc59180bfexample
```

このコマンドでは何も出力されません。

詳細については、GuardDuty 「ユーザーガイド」の [「一時停止または無効化 GuardDuty」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDetector](#)」の「」を参照してください。

delete-filter

次のコード例は、delete-filter を使用する方法を示しています。

AWS CLI

現在のリージョンの既存のフィルターを削除するには

この例では、フィルターの削除を作成する方法を示しています。

```
aws guardduty delete-filter \  
  --filter-id example
```

```
--detector-id b6b992d6d2f48e64bc59180bfexample \  
--filter-name byebyeFilter
```

このコマンドでは何も出力されません。

詳細については、GuardDuty 「ユーザーガイド」の [「結果のフィルタリング」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteFilter](#)」の「」を参照してください。

disable-organization-admin-account

次のコード例は、`disable-organization-admin-account` を使用する方法を示しています。

AWS CLI

組織 GuardDuty 内の の委任管理者としてアカウントを削除するには

この例では、 の委任管理者としてアカウントを削除する方法を示しています GuardDuty。

```
aws guardduty disable-organization-admin-account \  
--admin-account-id 111122223333
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の [AWS 「組織でのアカウントの管理」](#) を参照してください。GuardDuty

- API 詳細については、AWS CLI 「コマンドリファレンス [DisableOrganizationAdminAccount](#)」の「」を参照してください。

disassociate-from-master-account

次のコード例は、`disassociate-from-master-account` を使用する方法を示しています。

AWS CLI

現在のリージョンの現在のマスターアカウントとの関連付けを解除するには

次の`disassociate-from-master-account`例では、アカウントを現在の AWS リージョンの現在の GuardDuty マスターアカウントから関連付け解除します。

```
aws guardduty disassociate-from-master-account \  
--detector-id d4b040365221be2b54a6264dcexample
```

このコマンドでは何も出力されません。

詳細については、GuardDuty「[ユーザーガイド](#)」の [GuardDuty「マスターアカウントとメンバーアカウントの関係を理解する](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDisassociateFromMasterAccount](#)」の「」を参照してください。

get-detector

次のコード例は、get-detector を使用する方法を示しています。

AWS CLI

特定のディテクターの詳細を取得するには

次のget-detector例では、指定されたディテクターの設定の詳細を表示します。

```
aws guardduty get-detector \  
--detector-id 12abc34d567e8fa901bc2d34eexample
```

出力:

```
{  
  "Status": "ENABLED",  
  "ServiceRole": "arn:aws:iam::111122223333:role/aws-service-role/  
guardduty.amazonaws.com/AWSServiceRoleForAmazonGuardDuty",  
  "Tags": {},  
  "FindingPublishingFrequency": "SIX_HOURS",  
  "UpdatedAt": "2018-11-07T03:24:22.938Z",  
  "CreatedAt": "2017-12-22T22:51:31.940Z"  
}
```

詳細については、[「ユーザーガイド」の「概念と用語」](#)を参照してください。GuardDuty

- API 詳細については、AWS CLI「[コマンドリファレンスGetDetector](#)」の「」を参照してください。

get-findings

次の例は、get-findings を使用する方法を説明しています。

AWS CLI

例 1: 特定の検出結果の詳細を取得するには

次のget-findings例では、指定されたJSON検出結果の完全な検出結果の詳細を取得します。

```
aws guardduty get-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-id 1ab92989eaf0e742df4a014d5example
```

出力:

```
{  
  "Findings": [  
    {  
      "Resource": {  
        "ResourceType": "AccessKey",  
        "AccessKeyDetails": {  
          "UserName": "testuser",  
          "UserType": "IAMUser",  
          "PrincipalId": "AIDACKCEVSQ6C2EXAMPLE",  
          "AccessKeyId": "ASIASZ4SI7REEEXAMPLE"  
        }  
      },  
      "Description": "APIs commonly used to discover the users, groups,  
policies and permissions in an account, was invoked by IAM principal testuser under  
unusual circumstances. Such activity is not typically seen from this principal.",  
      "Service": {  
        "Count": 5,  
        "Archived": false,  
        "ServiceName": "guardduty",  
        "EventFirstSeen": "2020-05-26T22:02:24Z",  
        "ResourceRole": "TARGET",  
        "EventLastSeen": "2020-05-26T22:33:55Z",  
        "DetectorId": "d4b040365221be2b54a6264dcexample",  
        "Action": {  
          "ActionType": "AWS_API_CALL",  
          "AwsApiCallAction": {  
            "RemoteIpDetails": {
```



```

        "GeoLocation": {
            "Lat": 51.5164,
            "Lon": -0.093
        },
        "City": {
            "CityName": "London"
        },
        "IpAddressV4": "52.94.36.7",
        "Organization": {
            "Org": "Amazon.com",
            "Isp": "Amazon.com",
            "Asn": "16509",
            "AsnOrg": "AMAZON-02"
        },
        "Country": {
            "CountryName": "United Kingdom"
        }
    },
    "Api": "ListPolicyVersions",
    "ServiceName": "iam.amazonaws.com",
    "CallerType": "Remote IP"
}
}
},
"Title": "Unusual user permission reconnaissance activity by testuser.",
"Type": "Recon:IAMUser/UserPermissions",
"Region": "us-east-1",
"Partition": "aws",
"Arn": "arn:aws:guardduty:us-east-1:111122223333:detector/
d4b040365221be2b54a6264dcexample/finding/1ab92989eaf0e742df4a014d5example",
"UpdatedAt": "2020-05-26T22:55:21.703Z",
"SchemaVersion": "2.0",
"Severity": 5,
"Id": "1ab92989eaf0e742df4a014d5example",
"CreatedAt": "2020-05-26T22:21:48.385Z",
"AccountId": "111122223333"
}
]
}

```

詳細については、GuardDuty 「ユーザーガイド」の「[調査結果](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFindings](#)」の「」を参照してください。

get-ip-set

次の例は、get-ip-set を使用する方法を説明しています。

AWS CLI

指定された信頼された IP セットの詳細を一覧表示するには

次のget-ip-set例は、指定された信頼された IP セットのステータスと詳細を示しています。

```
aws guardduty get-ip-set \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --ip-set-id d4b94fc952d6912b8f3060768example
```

出力:

```
{  
  "Status": "ACTIVE",  
  "Location": "s3://AWSDOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com/  
customlist.csv",  
  "Tags": {},  
  "Format": "TXT",  
  "Name": "test-ip-set"  
}
```

詳細については、[「ユーザーガイド」の「信頼された IP リストと脅威リストの使用」](#)を参照してください。 GuardDuty

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIpSet](#)」の「」を参照してください。

get-master-account

次のコード例は、get-master-account を使用する方法を示しています。

AWS CLI

現在のリージョンのマスターアカウントの詳細を取得するには

次のget-master-account例では、現在のリージョンのディテクターに関連付けられたマスターアカウントのステータスと詳細を表示します。

```
aws guardduty get-master-account \  
--detector-id 12abc34d567e8fa901bc2d34eexample
```

出力:

```
{  
  "Master": {  
    "InvitationId": "04b94d9704854a73f94e061e8example",  
    "InvitedAt": "2020-06-09T22:23:04.970Z",  
    "RelationshipStatus": "Enabled",  
    "AccountId": "123456789111"  
  }  
}
```

詳細については、GuardDuty「[ユーザーガイド](#)」のGuardDuty「[マスターアカウントとメンバーアカウントの関係を理解する](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetMasterAccount](#)」の「」を参照してください。

list-detectors

次の例は、list-detectors を使用する方法を説明しています。

AWS CLI

現在のリージョンで使用可能なディテクターを一覧表示するには

次のlist-detectors例では、現在のAWSリージョンで使用可能なディテクターを一覧表示します。

```
aws guardduty list-detectors
```

出力:

```
{  
  "DetectorIds": [  
    "12abc34d567e8fa901bc2d34eexample"  
  ]  
}
```

詳細については、[「ユーザーガイド」の「概念と用語」](#)を参照してください。GuardDuty

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDetectors](#)」の「」を参照してください。

list-findings

次の例は、list-findings を使用する方法を説明しています。

AWS CLI

例 1: 現在のリージョンのすべての検出結果を一覧表示するには

次のlist-findings例では、現在のリージョン findingIds のすべてのリストを重要度の高い順にソートして表示します。

```
aws guardduty list-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --sort-criteria '{"AttributeName": "severity", "OrderBy": "DESC"}'
```

出力:

```
{  
  "FindingIds": [  
    "04b8ab50fd29c64fc771b232dexample",  
    "5ab8ab50fd21373735c826d3aexample",  
    "90b93de7aba69107f05bbe60bexample",  
    ...  
  ]  
}
```

詳細については、GuardDuty [「ユーザーガイド」の「調査結果」](#)を参照してください。

例 2: 特定の検出結果基準に一致する現在のリージョンの検出結果を一覧表示するには

次のlist-findings例では、指定された検出結果タイプ findingIds に一致するすべての のリストを表示します。

```
aws guardduty list-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-criteria '{"Criterion":{"type": {"Eq":["UnauthorizedAccess:EC2/  
SSHBruteForce"]}}}'
```

出力:

```
{
  "FindingIds": [
    "90b93de7aba69107f05bbe60bexample",
    "6eb9430d7023d30774d6f05e3example",
    "2eb91a2d060ac9a21963a5848example",
    "44b8ab50fd2b0039a9e48f570example",
    "9eb8ab4cd2b7e5b66ba4f5e96example",
    "e0b8ab3a38e9b0312cc390ceeexample"
  ]
}
```

詳細については、GuardDuty「[ユーザーガイド](#)」の「[結果](#)」を参照してください。

例 3: JSON ファイル内で定義された特定の検出基準のセットに一致する現在のリージョンの検出結果を一覧表示するには

次のlist-findings例では、JSONファイルで指定されているように、アーカイブ findingIds されていないすべてのリストと「testuser」という名前のIAMユーザーを表示します。

```
aws guardduty list-findings \
  --detector-id 12abc34d567e8fa901bc2d34eexample \
  --finding-criteria file://myfile.json
```

myfile.json の内容:

```
{"Criterion": {
  "resource.accessKeyDetails.userName": {
    "Eq": [
      "testuser"
    ]
  },
  "service.archived": {
    "Eq": [
      "false"
    ]
  }
}
```

出力:

```
{
  "FindingIds": [
    "1ab92989eaf0e742df4a014d5example"
  ]
}
```

詳細については、GuardDuty 「ユーザーガイド」の「[調査結果](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFindings](#)」の「」を参照してください。

list-invitations

次の例は、list-invitations を使用する方法を説明しています。

AWS CLI

現在のリージョンのメンバーアカウントになるための招待の詳細を一覧表示するには

次のlist-invitations例では、現在のリージョンのメンバー GuardDuty アカウントになるための招待の詳細とステータスを一覧表示します。

```
aws guardduty list-invitations
```

出力:

```
{
  "Invitations": [
    {
      "InvitationId": "d6b94fb03a66ff665f7db8764example",
      "InvitedAt": "2020-06-10T17:56:38.221Z",
      "RelationshipStatus": "Invited",
      "AccountId": "123456789111"
    }
  ]
}
```

詳細については、GuardDuty 「ユーザーガイド」の「[招待による GuardDuty アカウントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInvitations](#)」の「」を参照してください。

list-ip-sets

次のコード例は、list-ip-sets を使用する方法を示しています。

AWS CLI

現在のリージョンで信頼された IP セットを一覧表示するには

次のlist-ip-sets例では、現在の AWS リージョンの信頼された IP セットを一覧表示します。

```
aws guardduty list-ip-sets \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

出力:

```
{  
  "IpSetIds": [  
    "d4b94fc952d6912b8f3060768example"  
  ]  
}
```

詳細については、[「ユーザーガイド」の「信頼された IP リストと脅威リストの使用」](#)を参照してください。 GuardDuty

- API 詳細については、AWS CLI 「コマンドリファレンス[ListIpSets](#)」の「」を参照してください。

list-members

次のコード例は、list-members を使用する方法を示しています。

AWS CLI

現在のリージョンのすべてのメンバーを一覧表示するには

次のlist-members例では、現在のリージョンのすべてのメンバーアカウントとその詳細を一覧表示します。

```
aws guardduty list-members \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

```
--detector-id 12abc34d567e8fa901bc2d34eexample
```

出力:

```
{
  "Members": [
    {
      "RelationshipStatus": "Enabled",
      "InvitedAt": "2020-06-09T22:49:00.910Z",
      "MasterId": "123456789111",
      "DetectorId": "7ab8b2f61b256c87f793f6a86example",
      "UpdatedAt": "2020-06-09T23:08:22.512Z",
      "Email": "your+member@example.com",
      "AccountId": "123456789222"
    }
  ]
}
```

詳細については、GuardDuty 「ユーザーガイド」の [GuardDuty 「マスターアカウントとメンバーアカウントの関係を理解する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListMembers](#)」の「」を参照してください。

update-ip-set

次の例は、update-ip-set を使用する方法を説明しています。

AWS CLI

信頼された IP セットを更新するには

次のupdate-ip-set例は、信頼された IP セットの詳細を更新する方法を示しています。

```
aws guardduty update-ip-set \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --ip-set-id d4b94fc952d6912b8f3060768example \  
  --location https://AWSDOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com/  
customtrustlist2.csv
```

このコマンドでは何も出力されません。

詳細については、「[ユーザーガイド](#)」の「[信頼された IP リストと脅威リストの使用](#)」を参照してください。 GuardDuty

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIpSet](#)」の「」を参照してください。

AWS Health を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Health。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-affected-entities

次の例は、describe-affected-entities を使用する方法を説明しています。

AWS CLI

指定された AWS ヘルスイベントによって影響を受けるエンティティを一覧表示するには

次のdescribe-affected-entities例では、指定された AWS ヘルスイベントによって影響を受けるエンティティを一覧表示します。このイベントは、AWS アカウントの請求通知です。

```
aws health describe-affected-entities \  
  --filter "eventArns=arn:aws:health:global::event/BILLING/  
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-  
EXAMPLE11145" \  
  --region us-east-1
```

出力:

```
{
  "entities": [
    {
      "entityArn": "arn:aws:health:global:123456789012:entity/
EXAMPLEimSMoULmWHpb",
      "eventArn": "arn:aws:health:global::event/BILLING/
AWS_BILLING_NOTIFICATION/AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-
EXAMPLE11145",
      "entityValue": "AWS_ACCOUNT",
      "awsAccountId": "123456789012",
      "lastUpdatedTime": 1588356454.08
    }
  ]
}
```

詳細については、「ヘルスユーザーガイド」の「[イベントログ](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAffectedEntities](#)」の「」を参照してください。

describe-event-details

次のコード例は、describe-event-details を使用する方法を示しています。

AWS CLI

Health AWS イベントに関する情報を一覧表示するには

次のdescribe-event-details例は、指定された AWS ヘルスイベントに関する情報を示しています。

```
aws health describe-event-details \
  --event-arns "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111" \
  --region us-east-1
```

出力:

```
{
  "successfulSet": [
    {
```

```
    "event": {
      "arn": "arn:aws:health:us-east-1::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",
      "service": "EC2",
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
      "eventTypeCategory": "issue",
      "region": "us-east-1",
      "startTime": 1587462325.096,
      "endTime": 1587464204.774,
      "lastUpdatedTime": 1587464204.865,
      "statusCode": "closed"
    },
    "eventDescription": {
      "latestDescription": "[RESOLVED] Increased API Error Rates and
Latencies\n\n[02:45 AM PDT] We are investigating increased API error rates and
latencies in the US-EAST-1 Region.\n\n[03:16 AM PDT] Between 2:10 AM and 2:59 AM
PDT we experienced increased API error rates and latencies in the US-EAST-1 Region.
The issue has been resolved and the service is operating normally."
    }
  }
],
  "failedSet": []
}
```

詳細については、「ヘルスユーザーガイド」の「[イベントの詳細ペイン](#)」を参照してください。
AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEventDetails](#)」の「」を参照してください。

describe-events

次の例は、describe-events を使用する方法を説明しています。

AWS CLI

例 1: AWS ヘルスイベントを一覧表示するには

次のdescribe-events例では、最近の AWS ヘルスイベントを一覧表示します。

```
aws health describe-events \  
  --region us-east-1
```

出力:

```
{
  "events": [
    {
      "arn": "arn:aws:health:us-west-1::event/ECS/AWS_ECS_OPERATIONAL_ISSUE/
AWS_ECS_OPERATIONAL_ISSUE_KWQPY_EXAMPLE111",
      "service": "ECS",
      "eventTypeCode": "AWS_ECS_OPERATIONAL_ISSUE",
      "eventTypeCategory": "issue",
      "region": "us-west-1",
      "startTime": 1589077890.53,
      "endTime": 1589086345.597,
      "lastUpdatedTime": 1589086345.905,
      "statusCode": "closed",
      "eventScopeCode": "PUBLIC"
    },
    {
      "arn": "arn:aws:health:global::event/BILLING/AWS_BILLING_NOTIFICATION/
AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-EXAMPLE1118b",
      "service": "BILLING",
      "eventTypeCode": "AWS_BILLING_NOTIFICATION",
      "eventTypeCategory": "accountNotification",
      "region": "global",
      "startTime": 1588356000.0,
      "lastUpdatedTime": 1588356524.358,
      "statusCode": "open",
      "eventScopeCode": "ACCOUNT_SPECIFIC"
    },
    {
      "arn": "arn:aws:health:us-west-2::event/
CLOUDFORMATION/AWS_CLOUDFORMATION_OPERATIONAL_ISSUE/
AWS_CLOUDFORMATION_OPERATIONAL_ISSUE_OHTWY_EXAMPLE111",
      "service": "CLOUDFORMATION",
      "eventTypeCode": "AWS_CLOUDFORMATION_OPERATIONAL_ISSUE",
      "eventTypeCategory": "issue",
      "region": "us-west-2",
      "startTime": 1588279630.761,
      "endTime": 1588284650.0,
      "lastUpdatedTime": 1588284691.941,
      "statusCode": "closed",
      "eventScopeCode": "PUBLIC"
    },
    {
```

```
    "arn": "arn:aws:health:ap-northeast-1::event/LAMBDA/
AWS_LAMBDA_OPERATIONAL_ISSUE/AWS_LAMBDA_OPERATIONAL_ISSUE_JZDND_EXAMPLE111",
    "service": "LAMBDA",
    "eventTypeCode": "AWS_LAMBDA_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "ap-northeast-1",
    "startTime": 1587379534.08,
    "endTime": 1587391771.0,
    "lastUpdatedTime": 1587395689.316,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
    "service": "EC2",
    "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "us-east-1",
    "startTime": 1586473044.284,
    "endTime": 1586479706.091,
    "lastUpdatedTime": 1586479706.153,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:global::event/SECURITY/AWS_SECURITY_NOTIFICATION/
AWS_SECURITY_NOTIFICATION_42007387-8129-42da-8c88-EXAMPLE11139",
    "service": "SECURITY",
    "eventTypeCode": "AWS_SECURITY_NOTIFICATION",
    "eventTypeCategory": "accountNotification",
    "region": "global",
    "startTime": 1585674000.0,
    "lastUpdatedTime": 1585674004.132,
    "statusCode": "open",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:global::event/CLOUDFRONT/
AWS_CLOUDFRONT_OPERATIONAL_ISSUE/AWS_CLOUDFRONT_OPERATIONAL_ISSUE_FRQXG_EXAMPLE111",
    "service": "CLOUDFRONT",
    "eventTypeCode": "AWS_CLOUDFRONT_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "global",
```

```
    "startTime": 1585610898.589,
    "endTime": 1585617671.0,
    "lastUpdatedTime": 1585620638.869,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:us-east-1::event/SES/AWS_SES_OPERATIONAL_ISSUE/
AWS_SES_OPERATIONAL_ISSUE_URNDF_EXAMPLE111",
    "service": "SES",
    "eventTypeCode": "AWS_SES_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "us-east-1",
    "startTime": 1585342008.46,
    "endTime": 1585344017.0,
    "lastUpdatedTime": 1585344355.989,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  },
  {
    "arn": "arn:aws:health:global::event/IAM/
AWS_IAM_OPERATIONAL_NOTIFICATION/
AWS_IAM_OPERATIONAL_NOTIFICATION_b6771c34-6ecd-4aea-9d3e-EXAMPLE1117e",
    "service": "IAM",
    "eventTypeCode": "AWS_IAM_OPERATIONAL_NOTIFICATION",
    "eventTypeCategory": "accountNotification",
    "region": "global",
    "startTime": 1584978300.0,
    "lastUpdatedTime": 1584978553.572,
    "statusCode": "open",
    "eventScopeCode": "ACCOUNT_SPECIFIC"
  },
  {
    "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
    "service": "EC2",
    "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
    "eventTypeCategory": "issue",
    "region": "ap-southeast-2",
    "startTime": 1583881487.483,
    "endTime": 1583885056.785,
    "lastUpdatedTime": 1583885057.052,
    "statusCode": "closed",
    "eventScopeCode": "PUBLIC"
  }
}
```

```
    }  
  ]  
}
```

詳細については、[「ヘルスユーザーガイド」の AWS 「パーソナルヘルスダッシュボードの開始方法」](#)を参照してください。AWS

例 2: サービスおよびイベントステータスコード別に AWS ヘルスイベントを一覧表示するには次の describe-events 例では、イベントステータスが閉じられている Amazon Elastic Compute Cloud (Amazon EC2) の AWS ヘルスイベントを一覧表示します。

```
aws health describe-events \  
  --filter "services=EC2,eventStatusCodes=closed"
```

出力:

```
{  
  "events": [  
    {  
      "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/  
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",  
      "service": "EC2",  
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",  
      "eventTypeCategory": "issue",  
      "region": "us-east-1",  
      "startTime": 1587462325.096,  
      "endTime": 1587464204.774,  
      "lastUpdatedTime": 1587464204.865,  
      "statusCode": "closed",  
      "eventScopeCode": "PUBLIC"  
    },  
    {  
      "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/  
AWS_EC2_OPERATIONAL_ISSUE_COBJXJ_EXAMPLE111",  
      "service": "EC2",  
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",  
      "eventTypeCategory": "issue",  
      "region": "us-east-1",  
      "startTime": 1586473044.284,  
      "endTime": 1586479706.091,  
      "lastUpdatedTime": 1586479706.153,  
    }  
  ]  
}
```

```
        "statusCode": "closed",
        "eventScopeCode": "PUBLIC"
    },
    {
        "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
        "eventTypeCategory": "issue",
        "region": "ap-southeast-2",
        "startTime": 1583881487.483,
        "endTime": 1583885056.785,
        "lastUpdatedTime": 1583885057.052,
        "statusCode": "closed",
        "eventScopeCode": "PUBLIC"
    }
]
}
```

詳細については、[「ヘルスユーザーガイド」の AWS 「パーソナルヘルスダッシュボードの開始方法」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

HealthImaging を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています HealthImaging。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

copy-image-set

次の例は、copy-image-set を使用する方法を説明しています。

AWS CLI

例 1：コピー先を指定せずに画像セットをコピーするには。

次のcopy-image-set例では、送信先のないイメージセットの複製コピーを作成します。

```
aws medical-imaging copy-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" } }'
```

出力:

```
{  
  "destinationImageSetProperties": {  
    "latestVersionId": "2",  
    "imageSetWorkflowStatus": "COPYING",  
    "updatedAt": 1680042357.432,  
    "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",  
    "imageSetState": "LOCKED",  
    "createdAt": 1680042357.432  
  },  
  "sourceImageSetProperties": {  
    "latestVersionId": "1",  
    "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",  
    "updatedAt": 1680042357.432,  
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
    "imageSetState": "LOCKED",  
    "createdAt": 1680027126.436  
  },  
  "datastoreId": "12345678901234567890123456789012"  
}
```

例 2：コピー先を指定して画像セットをコピーするには。

次のcopy-image-set例では、送信先を持つイメージセットの複製コピーを作成します。

```
aws medical-imaging copy-image-set \
  --datastore-id 12345678901234567890123456789012 \
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" },
  "destinationImageSet": { "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
  "latestVersionId": "1"} }'
```

出力:

```
{
  "destinationImageSetProperties": {
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "COPYING",
    "updatedAt": 1680042505.135,
    "imageSetId": "b9a06fef182a5f992842f77f8e0868e5",
    "imageSetState": "LOCKED",
    "createdAt": 1680042357.432
  },
  "sourceImageSetProperties": {
    "latestVersionId": "1",
    "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
    "updatedAt": 1680042505.135,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436
  },
  "datastoreId": "12345678901234567890123456789012"
}
```

例 3: ソースイメージセットから宛先イメージセットにインスタンスのサブセットをコピーするには。

次のcopy-image-set例では、ソースイメージセットから宛先イメージセットに 1 つのDICOM インスタンスをコピーします。フォースパラメータは、患者、スタディ、シリーズレベルの属性の不整合を上書きするために提供されます。

```
aws medical-imaging copy-image-set \
  --datastore-id 12345678901234567890123456789012 \
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId":
  "1", "DICOMCopies": {"copiableAttributes": {"\SchemaVersion\":"1.1", "\Study\":"
```

```
{\"Series\":{\\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3666.0\":
{\"Instances\":
{\\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3669.0\":
{}}}}}\"}}, \"destinationImageSet\": {\"imageSetId\":
\"b9eb50d8ee682eb9fcf4acbf92f62bb7\", \"latestVersionId\": \"1\"}}' \\
--force
```

出力:

```
{
  \"destinationImageSetProperties\": {
    \"latestVersionId\": \"2\",
    \"imageSetWorkflowStatus\": \"COPYING\",
    \"updatedAt\": 1680042505.135,
    \"imageSetId\": \"b9eb50d8ee682eb9fcf4acbf92f62bb7\",
    \"imageSetState\": \"LOCKED\",
    \"createdAt\": 1680042357.432
  },
  \"sourceImageSetProperties\": {
    \"latestVersionId\": \"1\",
    \"imageSetWorkflowStatus\": \"COPYING_WITH_READ_ONLY_ACCESS\",
    \"updatedAt\": 1680042505.135,
    \"imageSetId\": \"ea92b0d8838c72a3f25d00d13616f87e\",
    \"imageSetState\": \"LOCKED\",
    \"createdAt\": 1680027126.436
  },
  \"datastoreId\": \"12345678901234567890123456789012\"
}
```

詳細については、AWS HealthImaging 「[デベロッパーガイド](#)」の「[イメージセットのコピー](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCopyImageSet](#)」の「」を参照してください。

create-datastore

次のコード例は、create-datastore を使用する方法を示しています。

AWS CLI

データストアを作成するには

次の create-datastore コード例では、my-datastore という名が付けられたデータストアを作成しています。

```
aws medical-imaging create-datastore \  
  --datastore-name "my-datastore"
```

出力:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "datastoreStatus": "CREATING"  
}
```

詳細については、「AWS HealthImaging デベロッパーガイド」の [「データストアの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDatastore](#)」の「」を参照してください。

delete-datastore

次の例は、delete-datastore を使用する方法を説明しています。

AWS CLI

データストアを削除するには

次の delete-datastore コード例では、データストアを削除しています。

```
aws medical-imaging delete-datastore \  
  --datastore-id "12345678901234567890123456789012"
```

出力:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "datastoreStatus": "DELETING"  
}
```

詳細については、「AWS HealthImaging デベロッパーガイド」の [「データストアの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDatastore](#)」の「」を参照してください。

delete-image-set

次のコード例は、delete-image-set を使用する方法を示しています。

AWS CLI

画像セットを削除するには

以下の delete-image-set コード例は画像セットを削除しています。

```
aws medical-imaging delete-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

出力:

```
{  
  "imageSetWorkflowStatus": "DELETING",  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "datastoreId": "12345678901234567890123456789012"  
}
```

詳細については、「AWS HealthImaging デベロッパーガイド」の「[イメージセットの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteImageSet](#)」の「」を参照してください。

get-datastore

次のコード例は、get-datastore を使用する方法を示しています。

AWS CLI

データストアのプロパティを取得するには

次の get-datastore コード例では、データストアのプロパティを取得しています。

```
aws medical-imaging get-datastore \  
--datastore-id 12345678901234567890123456789012
```

出力:

```
{  
  "datastoreProperties": {  
    "datastoreId": "12345678901234567890123456789012",  
    "datastoreName": "TestDatastore123",  
    "datastoreStatus": "ACTIVE",  
    "datastoreArn": "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012",  
    "createdAt": "2022-11-15T23:33:09.643000+00:00",  
    "updatedAt": "2022-11-15T23:33:09.643000+00:00"  
  }  
}
```

詳細については、AWS HealthImaging 「デベロッパーガイド」の「[データストアプロパティの取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDatastore](#)」の「」を参照してください。

get-dicom-import-job

次のコード例は、get-dicom-import-job を使用する方法を示しています。

AWS CLI

DICOM インポートジョブのプロパティを取得するには

次の get-dicom-import-job コード例では、DICOM インポートジョブのプロパティを取得しています。

```
aws medical-imaging get-dicom-import-job \  
--datastore-id "12345678901234567890123456789012" \  
--job-id "09876543210987654321098765432109"
```

出力:

```
{
```

```
"jobProperties": {
  "jobId": "09876543210987654321098765432109",
  "jobName": "my-job",
  "jobStatus": "COMPLETED",
  "datastoreId": "12345678901234567890123456789012",
  "dataAccessRoleArn": "arn:aws:iam::123456789012:role/
ImportJobDataAccessRole",
  "endedAt": "2022-08-12T11:29:42.285000+00:00",
  "submittedAt": "2022-08-12T11:28:11.152000+00:00",
  "inputS3Uri": "s3://medical-imaging-dicom-input/dicom_input/",
  "outputS3Uri": "s3://medical-imaging-output/
job_output/12345678901234567890123456789012-
DicomImport-09876543210987654321098765432109/"
}
```

詳細については、AWS HealthImaging 「デベロッパーガイド」の [「インポートジョブプロパティの取得」](#) を参照してください。

- API 詳細については、「コマンドリファレンス」の [「GetDICOMImportジョブ」](#) を参照してください。AWS CLI

get-image-frame

次のコード例は、get-image-frame を使用する方法を示しています。

AWS CLI

画像セットのピクセルデータを取得するには

次の get-image-frame コード例では、画像フレームを取得しています。

```
aws medical-imaging get-image-frame \
  --datastore-id "12345678901234567890123456789012" \
  --image-set-id "98765412345612345678907890789012" \
  --image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \
  imageframe.jpg
```

注: このコード例では、GetImageFrame アクションがピクセルデータのストリームを imageframe.jpg ファイルに戻すため、出力は含まれません。イメージフレームのデコードと表示については、HTJ2K「デコードライブラリ」を参照してください。

詳細については、AWS HealthImaging 「デベロッパーガイド」の「[イメージセットピクセルデータの取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetImageFrame](#)」の「」を参照してください。

get-image-set-metadata

次の例は、get-image-set-metadata を使用する方法を説明しています。

AWS CLI

例 1：画像セットのメタデータをバージョンなしで取得するには

次の get-image-set-metadata コード例では、バージョンを指定せずに画像セットのメタデータを取得しています。

注：outfile は必須のパラメータです

```
aws medical-imaging get-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  studymetadata.json.gz
```

返されたメタデータは gzip で圧縮され、studymetadata.json.gz ファイルに保存されます。返されたJSONオブジェクトの内容を表示するには、まず解凍する必要があります。

出力:

```
{  
  "contentType": "application/json",  
  "contentEncoding": "gzip"  
}
```

例 2：画像セットのメタデータをバージョン付きで取得するには

次の get-image-set-metadata コード例では、指定されたバージョンの画像セットのメタデータを取得しています。

注：outfile は必須のパラメータです

```
aws medical-imaging get-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --version 1.0.0 \  
  studymetadata.json.gz
```



```
--datastore-id 12345678901234567890123456789012 \  
--image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
--version-id 1 \  
studymetadata.json.gz
```

返されたメタデータは gzip で圧縮され、studymetadata.json.gz ファイルに保存されます。返されたJSONオブジェクトの内容を表示するには、まず解凍する必要があります。

出力:

```
{  
  "contentType": "application/json",  
  "contentEncoding": "gzip"  
}
```

詳細については、AWS HealthImaging 「デベロッパーガイド」の「[イメージセットメタデータの取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetImageSetMetadata](#)」の「」を参照してください。

get-image-set

次のコード例は、get-image-set を使用する方法を示しています。

AWS CLI

画像セットのプロパティを取得するには

以下の get-image-set コード例では、画像セットのプロパティを取得しています。

```
aws medical-imaging get-image-set \  
--datastore-id 12345678901234567890123456789012 \  
--image-set-id 18f88ac7870584f58d56256646b4d92b \  
--version-id 1
```

出力:

```
{  
  "versionId": "1",  
  "imageSetWorkflowStatus": "COPIED",  
  "updatedAt": 1680027253.471,
```

```
"imageSetId": "18f88ac7870584f58d56256646b4d92b",
"imageSetState": "ACTIVE",
"createdAt": 1679592510.753,
"datastoreId": "12345678901234567890123456789012"
}
```

詳細については、「AWS HealthImaging デベロッパーガイド」の[「イメージセットプロパティの取得」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetImageSet](#)」の「」を参照してください。

list-datastores

次のコード例は、list-datastores を使用する方法を示しています。

AWS CLI

データストアを一覧表示するには

次の list-datastores コード例では、利用可能なデータストアを一覧表示しています。

```
aws medical-imaging list-datastores
```

出力:

```
{
  "datastoreSummaries": [
    {
      "datastoreId": "12345678901234567890123456789012",
      "datastoreName": "TestDatastore123",
      "datastoreStatus": "ACTIVE",
      "datastoreArn": "arn:aws:medical-imaging:us-east-1:123456789012:datastore/12345678901234567890123456789012",
      "createdAt": "2022-11-15T23:33:09.643000+00:00",
      "updatedAt": "2022-11-15T23:33:09.643000+00:00"
    }
  ]
}
```

詳細については、AWS HealthImaging 「デベロッパーガイド」の[「データストアの一覧表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDatastores](#)」の「」を参照してください。

list-dicom-import-jobs

次の例は、list-dicom-import-jobs を使用する方法を説明しています。

AWS CLI

DICOM インポートジョブを一覧表示するには

次の list-dicom-import-jobs コード例では、インポートジョブを一覧表示します。

```
aws medical-imaging list-dicom-import-jobs \  
--datastore-id "12345678901234567890123456789012"
```

出力:

```
{  
  "jobSummaries": [  
    {  
      "jobId": "09876543210987654321098765432109",  
      "jobName": "my-job",  
      "jobStatus": "COMPLETED",  
      "datastoreId": "12345678901234567890123456789012",  
      "dataAccessRoleArn": "arn:aws:iam::123456789012:role/  
ImportJobDataAccessRole",  
      "endedAt": "2022-08-12T11:21:56.504000+00:00",  
      "submittedAt": "2022-08-12T11:20:21.734000+00:00"  
    }  
  ]  
}
```

詳細については、AWS HealthImaging 「[デベロッパーガイド](#)」の「[インポートジョブの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス」の「[ListDICOMImportジョブ](#)」を参照してください。

list-image-set-versions

次の例は、list-image-set-versions を使用する方法を説明しています。

AWS CLI

画像セットバージョンを一覧表示するには

次の `list-image-set-versions` コード例では、画像セットのバージョン履歴を一覧表示しています。

```
aws medical-imaging list-image-set-versions \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

出力:

```
{  
  "imageSetPropertiesList": [  
    {  
      "ImageSetWorkflowStatus": "UPDATED",  
      "versionId": "4",  
      "updatedAt": 1680029436.304,  
      "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
      "imageSetState": "ACTIVE",  
      "createdAt": 1680027126.436  
    },  
    {  
      "ImageSetWorkflowStatus": "UPDATED",  
      "versionId": "3",  
      "updatedAt": 1680029163.325,  
      "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
      "imageSetState": "ACTIVE",  
      "createdAt": 1680027126.436  
    },  
    {  
      "ImageSetWorkflowStatus": "COPY_FAILED",  
      "versionId": "2",  
      "updatedAt": 1680027455.944,  
      "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
      "imageSetState": "ACTIVE",  
      "message": "INVALID_REQUEST: Series of SourceImageSet and  
DestinationImageSet don't match.",  
      "createdAt": 1680027126.436  
    },  
    {  
      "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
```

```
        "imageSetState": "ACTIVE",
        "versionId": "1",
        "ImageSetWorkflowStatus": "COPIED",
        "createdAt": 1680027126.436
    }
]
}
```

詳細については、AWS HealthImaging 「[デベロッパーガイド](#)」の「[イメージセットのバージョンを一覧表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListImageSetVersions](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

例 1：データストアリソースのタグを一覧表示するには

次の list-tags-for-resource コード例では、データストアのタグを一覧表示しています。

```
aws medical-imaging list-tags-for-resource \
  --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012"
```

出力:

```
{
  "tags":{
    "Deployment":"Development"
  }
}
```

例 2：画像セットリソースのタグを一覧表示するには

次の list-tags-for-resource コード例では、画像セットのタグを一覧表示しています。

```
aws medical-imaging list-tags-for-resource \
```

```
--resource-arn "arn:aws:medical-imaging:us-east-1:123456789012:datastore/12345678901234567890123456789012/imageset/18f88ac7870584f58d56256646b4d92b"
```

出力:

```
{
  "tags": {
    "Deployment": "Development"
  }
}
```

詳細については、AWS HealthImaging 「[デベロッパーガイド](#)」の「[でのリソースのタグ付け](#)」
[AWS HealthImaging](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListTagsForResource](#)」の「」を参照してください。

search-image-sets

次のコード例は、search-image-sets を使用する方法を示しています。

AWS CLI

例 1: EQUAL演算子を使用してイメージセットを検索するには

次のsearch-image-setsコード例では、EQUAL演算子を使用して、特定の値に基づいてイメージセットを検索します。

```
aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json
```

search-criteria.json の内容

```
{
  "filters": [{
    "values": [{"DICOMPatientId" : "SUBJECT08701"}],
    "operator": "EQUAL"
  }]
}
```

```
}

```

出力:

```
{
  "imageSetsMetadataSummaries": [{
    "imageSetId": "09876543210987654321098765432109",
    "createdAt": "2022-12-06T21:40:59.429000+00:00",
    "version": 1,
    "DICOMTags": {
      "DICOMStudyId": "2011201407",
      "DICOMStudyDate": "19991122",
      "DICOMPatientSex": "F",
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
      "DICOMPatientBirthDate": "19201120",
      "DICOMStudyDescription": "UNKNOWN",
      "DICOMPatientId": "SUBJECT08701",
      "DICOMPatientName": "Melissa844 Huel628",
      "DICOMNumberOfStudyRelatedInstances": 1,
      "DICOMStudyTime": "140728",
      "DICOMNumberOfStudyRelatedSeries": 1
    },
    "updatedAt": "2022-12-06T21:40:59.429000+00:00"
  ]
}
```

例 2: DICOMStudyDateと を使用してBETWEEN、演算子でイメージセットを検索するには DICOMStudyTime

次のsearch-image-setsコード例では、1990年1月1日(午前 12:00) から 2023年1月1日(午前 12:00) の間に生成されたDICOMスタディを含むイメージセットを検索します。

注: DICOMStudyTimeはオプションです。入力されていない場合は、フィルターで指定された日付の時間値は午前 0 時 (1 日の始まり) になります。

```
aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json
```

search-criteria.json の内容

```
{
```

```

"filters": [{
  "values": [{
    "DICOMStudyDateAndTime": {
      "DICOMStudyDate": "19900101",
      "DICOMStudyTime": "000000"
    }
  },
  {
    "DICOMStudyDateAndTime": {
      "DICOMStudyDate": "20230101",
      "DICOMStudyTime": "000000"
    }
  }
  ],
  "operator": "BETWEEN"
}]
}

```

出力:

```

{
  "imageSetsMetadataSummaries": [{
    "imageSetId": "09876543210987654321098765432109",
    "createdAt": "2022-12-06T21:40:59.429000+00:00",
    "version": 1,
    "DICOMTags": {
      "DICOMStudyId": "2011201407",
      "DICOMStudyDate": "19991122",
      "DICOMPatientSex": "F",
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
      "DICOMPatientBirthDate": "19201120",
      "DICOMStudyDescription": "UNKNOWN",
      "DICOMPatientId": "SUBJECT08701",
      "DICOMPatientName": "Melissa844 Huel628",
      "DICOMNumberOfStudyRelatedInstances": 1,
      "DICOMStudyTime": "140728",
      "DICOMNumberOfStudyRelatedSeries": 1
    },
    "updatedAt": "2022-12-06T21:40:59.429000+00:00"
  }]
}

```

例 3: を使用してBETWEEN演算子でイメージセットを検索するには createdAt (タイムスタディが以前に保持されていた)

次のsearch-image-setsコード例では、UTCタイムゾーンの HealthImaging 時間範囲内に DICOMスタディが保持されているイメージセットを検索します。

注: サンプル形式で指定 createdAt します (1985-04-12T23:20:50.52Z))。

```
aws medical-imaging search-image-sets \  
  --datastore-id 12345678901234567890123456789012 \  
  --search-criteria file://search-criteria.json
```

search-criteria.json の内容

```
{  
  "filters": [{  
    "values": [{  
      "createdAt": "1985-04-12T23:20:50.52Z"  
    },  
    {  
      "createdAt": "2022-04-12T23:20:50.52Z"  
    }],  
    "operator": "BETWEEN"  
  }]  
}
```

出力:

```
{  
  "imageSetsMetadataSummaries": [{  
    "imageSetId": "09876543210987654321098765432109",  
    "createdAt": "2022-12-06T21:40:59.429000+00:00",  
    "version": 1,  
    "DICOMTags": {  
      "DICOMStudyId": "2011201407",  
      "DICOMStudyDate": "19991122",  
      "DICOMPatientSex": "F",  
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",  
      "DICOMPatientBirthDate": "19201120",  
      "DICOMStudyDescription": "UNKNOWN",  
      "DICOMPatientId": "SUBJECT08701",  
      "DICOMPatientName": "Melissa844 Huel628",  
      "DICOMNumberOfStudyRelatedInstances": 1,  
      "DICOMStudyTime": "140728",  
      "DICOMNumberOfStudyRelatedSeries": 1  
    }  
  }]  
}
```

```
    },
    "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
  ]
}
```

例 4: EQUAL オペレータが ON DICOMSeriesInstanceUID と BETWEEN オンのイメージセットを検索 updatedAt し、 updatedAt フィールドでレスポンスを ASC 順番にソートするには

次の search-image-sets コード例では、DICOMSeriesInstanceUID と BETWEEN の EQUAL 演算子を使用してイメージセットを検索 updatedAt し、 updatedAt フィールドでレスポンスを ASC 順番にソートします。

注: サンプル形式で指定 updatedAt します (1985-04-12T23:20:50.52Z) 。

```
aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json
```

search-criteria.json の内容

```
{
  "filters": [{
    "values": [{
      "updatedAt": "2024-03-11T15:00:05.074000-07:00"
    }, {
      "updatedAt": "2024-03-11T16:00:05.074000-07:00"
    }],
    "operator": "BETWEEN"
  }, {
    "values": [{
      "DICOMSeriesInstanceUID": "1.2.840.99999999.84710745.943275268089"
    }],
    "operator": "EQUAL"
  }],
  "sort": {
    "sortField": "updatedAt",
    "sortOrder": "ASC"
  }
}
```

出力:

```
{
  "imageSetsMetadataSummaries": [{
    "imageSetId": "09876543210987654321098765432109",
    "createdAt": "2022-12-06T21:40:59.429000+00:00",
    "version": 1,
    "DICOMTags": {
      "DICOMStudyId": "2011201407",
      "DICOMStudyDate": "19991122",
      "DICOMPatientSex": "F",
      "DICOMStudyInstanceUID": "1.2.840.99999999.84710745.943275268089",
      "DICOMPatientBirthDate": "19201120",
      "DICOMStudyDescription": "UNKNOWN",
      "DICOMPatientId": "SUBJECT08701",
      "DICOMPatientName": "Melissa844 Huel628",
      "DICOMNumberOfStudyRelatedInstances": 1,
      "DICOMStudyTime": "140728",
      "DICOMNumberOfStudyRelatedSeries": 1
    },
    "lastUpdatedAt": "2022-12-06T21:40:59.429000+00:00"
  ]
}
```

詳細については、AWS HealthImaging [「デベロッパーガイド」の「イメージセットの検索」](#)を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスSearchImageSets」](#)の「」を参照してください。

start-dicom-import-job

次のコード例は、start-dicom-import-job を使用する方法を示しています。

AWS CLI

DICOM インポートジョブを開始するには

次の start-dicom-import-job コード例では、DICOM インポートジョブを開始しています。

```
aws medical-imaging start-dicom-import-job \
  --job-name "my-job" \
  --datastore-id "12345678901234567890123456789012" \
  --input-s3-uri "s3://medical-imaging-dicom-input/dicom_input/" \
  --output-s3-uri "s3://medical-imaging-output/job_output/" \
```

```
--data-access-role-arn "arn:aws:iam::123456789012:role/ImportJobDataAccessRole"
```

出力:

```
{
  "datastoreId": "12345678901234567890123456789012",
  "jobId": "09876543210987654321098765432109",
  "jobStatus": "SUBMITTED",
  "submittedAt": "2022-08-12T11:28:11.152000+00:00"
}
```

詳細については、AWS HealthImaging 「デベロッパーガイド」の [「インポートジョブの開始」](#) を参照してください。

- API 詳細については、「コマンドリファレンス」の [「StartDICOMImportジョブ」](#) を参照してください。AWS CLI

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

例 1：データストアにタグを付けるには

次の tag-resource コード例では、データストアにタグを付けています。

```
aws medical-imaging tag-resource \
  --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/1234567890123456789012" \
  --tags '{"Deployment":"Development"}'
```

このコマンドでは何も出力されません。

例 2：画像セットにタグを付けるには

次の tag-resource コード例では、画像セットにタグを付けています。

```
aws medical-imaging tag-resource \
  --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/1234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b" \
```

```
--tags '{"Deployment":"Development"}'
```

このコマンドでは何も出力されません。

詳細については、AWS HealthImaging 「[デベロッパーガイド](#)」の「[でのリソースのタグ付け AWS HealthImaging](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用方法を示しています。

AWS CLI

例 1：データストアのタグを削除するには

次の untag-resource コード例では、データストアにタグを削除します。

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012" \  
  --tag-keys ["Deployment"]
```

このコマンドでは何も出力されません。

例 2：画像セットにタグを削除するには

次の untag-resource コード例では、画像セットにタグを削除します。

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/1234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b" \  
  --tag-keys ["Deployment"]
```

このコマンドでは何も出力されません。

詳細については、AWS HealthImaging 「[デベロッパーガイド](#)」の「[でのリソースのタグ付け AWS HealthImaging](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-image-set-metadata

次の例は、update-image-set-metadata を使用する方法を説明しています。

AWS CLI

例 1: イメージセットメタデータに属性を挿入または更新するには

次のupdate-image-set-metadata例では、イメージセットメタデータに属性を挿入または更新します。

```
aws medical-imaging update-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --latest-version-id 1 \  
  --cli-binary-format raw-in-base64-out \  
  --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json の内容

```
{  
  "DICOMUpdates": {  
    "updatableAttributes": "{\"SchemaVersion\":1.1,\"Patient\":{\"DICOM\":{\"PatientName\": \"MX^MX\"}}}"  
  }  
}
```

出力:

```
{  
  "latestVersionId": "2",  
  "imageSetWorkflowStatus": "UPDATING",  
  "updatedAt": 1680042257.908,  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "createdAt": 1680027126.436,  
  "datastoreId": "12345678901234567890123456789012"  
}
```

例 2: イメージセットメタデータから属性を削除するには

次のupdate-image-set-metadata例では、イメージセットメタデータから属性を削除します。

```
aws medical-imaging update-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --latest-version-id 1 \  
  --cli-binary-format raw-in-base64-out \  
  --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json の内容

```
{  
  "DICOMUpdates": {  
    "removableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"DICOM\":  
    {\"StudyDescription\":\"CHEST\"}}}"  
  }  
}
```

出力:

```
{  
  "latestVersionId": "2",  
  "imageSetWorkflowStatus": "UPDATING",  
  "updatedAt": 1680042257.908,  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "createdAt": 1680027126.436,  
  "datastoreId": "12345678901234567890123456789012"  
}
```

例 3: イメージセットメタデータからインスタンスを削除するには

次のupdate-image-set-metadata例では、イメージセットメタデータからインスタンスを削除します。

```
aws medical-imaging update-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --latest-version-id 1 \  
  --update-image-set-metadata-updates file://metadata-updates.json
```

```
--cli-binary-format raw-in-base64-out \  
--update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json の内容

```
{  
  "DICOMUpdates": {  
    "removableAttributes": "{\"SchemaVersion\": 1.1, \"Study\": {\"Series  
\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances\":  
  {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {}}}}}}}"  
  }  
}
```

出力:

```
{  
  "latestVersionId": "2",  
  "imageSetWorkflowStatus": "UPDATING",  
  "updatedAt": 1680042257.908,  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "createdAt": 1680027126.436,  
  "datastoreId": "12345678901234567890123456789012"  
}
```

例 4: イメージセットを以前のバージョンに戻すには

次の update-image-set-metadata 例は、イメージセットを以前のバージョンに戻す方法を示しています。CopyImageSet UpdateImageSetMetadata アクションは新しいバージョンのイメージセットを作成します。

```
aws medical-imaging update-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \  
  --latest-version-id 3 \  
  --cli-binary-format raw-in-base64-out \  
  --update-image-set-metadata-updates '{"revertToVersionId": "1"}'
```

出力:

```
{
```



```

    "datastoreId": "12345678901234567890123456789012",
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "latestVersionId": "4",
    "imageSetState": "LOCKED",
    "imageSetWorkflowStatus": "UPDATING",
    "createdAt": 1680027126.436,
    "updatedAt": 1680042257.908
  }

```

例 5: インスタンスにプライベートDICOMデータ要素を追加するには

次のupdate-image-set-metadata例は、イメージセット内の指定されたインスタンスにプライベート要素を追加する方法を示しています。このDICOM標準では、標準データ要素に含めることができない情報の通信にプライベートデータ要素を許可します。UpdateImageSetMetadata アクションを使用して、プライベートデータ要素を作成、更新、削除できます。

```

aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
  --latest-version-id 1 \
  --cli-binary-format raw-in-base64-out \
  --force \
  --update-image-set-metadata-updates file://metadata-updates.json

```

metadata-updates.json の内容

```

{
  "DICOMUpdates": {
    "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"DICOM\": {\"001910F9\": \"97\"},\"DICOMVRs\": {\"001910F9\": \"DS\"}}}}}}}"
  }
}

```

出力:

```

{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,

```

```

    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
  }

```

例 6: プライベートDICOMデータ要素をインスタンスに更新するには

次のupdate-image-set-metadata例は、イメージセット内のインスタンスに属するプライベートデータ要素の値を更新する方法を示しています。

```

aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
  --latest-version-id 1 \
  --cli-binary-format raw-in-base64-out \
  --force \
  --update-image-set-metadata-updates file://metadata-updates.json

```

metadata-updates.json の内容

```

{
  "DICOMUpdates": {
    "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\\\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\\\": {\"Instances
\\\": {\"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\\\": {\"DICOM\":
{\\\"00091001\\\": \\\"GE_GENESIS_DD\\\"}}}}}}}"
  }
}

```

出力:

```

{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}

```

例 7: force パラメータSOPInstanceUIDを使用して を更新するには

次のupdate-image-set-metadata例は、force パラメータを使用してDICOMメタデータの制約をオーバーライドしSOPInstanceUID、 を更新する方法を示しています。

```
aws medical-imaging update-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \  
  --latest-version-id 1 \  
  --cli-binary-format raw-in-base64-out \  
  --force \  
  --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json の内容

```
{  
  "DICOMUpdates": {  
    "updatableAttributes": "{ \"SchemaVersion\":1.1,\"Study\":{ \"Series\":  
  { \"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3656.0\":{ \"Instances  
  \":{ \"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.0\":{ \"DICOM\":  
  { \"SOPInstanceUID\":  
  \"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.9\"}}}}}}}"  
  }  
}
```

出力:

```
{  
  "latestVersionId": "2",  
  "imageSetWorkflowStatus": "UPDATING",  
  "updatedAt": 1680042257.908,  
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",  
  "imageSetState": "LOCKED",  
  "createdAt": 1680027126.436,  
  "datastoreId": "12345678901234567890123456789012"  
}
```

詳細については、AWS HealthImaging 「デベロッパーガイド」の [「イメージセットメタデータの更新」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateImageSetMetadata](#)」の「」を参照してください。

HealthLake を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています HealthLake。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-fhir-datastore

次のコード例は、create-fhir-datastore を使用する方法を示しています。

AWS CLI

FHIR データストアを作成するには。

次のcreate-fhir-datastore例は、Amazon で新しいデータストアを作成する方法を示しています HealthLake。

```
aws healthlake create-fhir-datastore \  
  --region us-east-1 \  
  --datastore-type-version R4 \  
  --datastore-type-version R4 \  
  --datastore-name "FhirTestDatastore"
```

出力:

```
{  
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
(Datastore ID)/r4/",  
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/  
(Datastore ID)",
```

```
"DatastoreStatus": "CREATING",
"DatastoreId": "(Datastore ID)"
}
```

詳細については、「Amazon HealthLake デベロッパーガイド」の[FHIR「データストアの作成とモニタリング」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateFhirDatastore](#)」の「」を参照してください。

delete-fhir-datastore

次の例は、delete-fhir-datastore を使用する方法を説明しています。

AWS CLI

FHIR データストアを削除するには

次のdelete-fhir-datastore例は、Amazon で Data Store とそのすべてのコンテンツを削除する方法を示しています HealthLake。

```
aws healthlake delete-fhir-datastore \
  --datastore-id (Data Store ID) \
  --region us-east-1
```

出力:

```
{
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
(Datastore ID)/r4/",
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/
(Datastore ID)",
  "DatastoreStatus": "DELETING",
  "DatastoreId": "(Datastore ID)"
}
```

詳細については、「Amazon HealthLake デベロッパーガイド」のFHIR「Data Store <<https://docs.aws.amazon.com/healthlake/latest/devguide/working-withFHIR-healthlake.html>> の作成とモニタリング」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteFhirDatastore](#)」の「」を参照してください。

describe-fhir-datastore

次のコード例は、describe-fhir-datastore を使用する方法を示しています。

AWS CLI

FHIR データストアを記述するには

次のdescribe-fhir-datastore例は、Amazon で Data Store のプロパティを検索する方法を示しています HealthLake。

```
aws healthlake describe-fhir-datastore \  
  --datastore-id "1f2f459836ac6c513ce899f9e4f66a59" \  
  --region us-east-1
```

出力:

```
{  
  "DatastoreProperties": {  
    "PreloadDataConfig": {  
      "PreloadDataType": "SYNTHEA"  
    },  
    "DatastoreName": "FhirTestDatastore",  
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/  
(Datastore ID)",  
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
(Datastore ID)/r4/",  
    "DatastoreStatus": "CREATING",  
    "DatastoreTypeVersion": "R4",  
    "DatastoreId": "(Datastore ID)"  
  }  
}
```

詳細については、「Amazon HealthLake デベロッパーガイド」の[FHIR「データストアの作成とモニタリング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFhirDatastore](#)」の「」を参照してください。

describe-fhir-export-job

次のコード例は、describe-fhir-export-job を使用する方法を示しています。

AWS CLI

FHIRエクスポートジョブを記述するには

次のdescribe-fhir-export-job例は、Amazon でFHIRエクスポートジョブのプロパティを検索する方法を示しています HealthLake。

```
aws healthlake describe-fhir-export-job \  
  --datastore-id (Datastore ID) \  
  --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

出力:

```
{  
  "ExportJobProperties": {  
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",  
    "JobStatus": "IN_PROGRESS",  
    "JobId": "9009813e9d69ba7cf79bcb3468780f16",  
    "SubmitTime": 1609175692.715,  
    "OutputDataConfig": {  
      "S3Uri": "s3://(Bucket Name)/(Prefix  
Name)/59593b2d0367ce252b5e66bf5fd6b574-  
FHIR_EXPORT-9009813e9d69ba7cf79bcb3468780f16/"  
    },  
    "DatastoreId": "(Datastore ID)"  
  }  
}
```

詳細については、「[Amazon HealthLake デベロッパーガイド](#)」のFHIR「[データストアからのファイルのエクスポート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFhirExportJob](#)」の「」を参照してください。

describe-fhir-import-job

次のコード例は、describe-fhir-import-job を使用する方法を示しています。

AWS CLI

FHIRインポートジョブを記述するには

次のdescribe-fhir-import-job例は、Amazon を使用してFHIRインポートジョブのプロパティを学習する方法を示しています HealthLake。

```
aws healthlake describe-fhir-import-job \  
  --datastore-id (Datastore ID) \  
  --job-id c145fbb27b192af392f8ce6e7838e34f \  
  --region us-east-1
```

出力:

```
{  
  "ImportJobProperties": {  
    "InputDataConfig": {  
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"  
      { "arrayitem2": 2 }  
    },  
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",  
    "JobStatus": "COMPLETED",  
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",  
    "SubmitTime": 1606272542.161,  
    "EndTime": 1606272609.497,  
    "DatastoreId": "(Datastore ID)"  
  }  
}
```

詳細については、「[Amazon HealthLake デベロッパーガイド](#)」のFHIR「[Data Store へのファイルのインポート](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeFhirImportJob](#)」の「」を参照してください。

list-fhir-datastores

次のコード例は、list-fhir-datastores を使用する方法を示しています。

AWS CLI

FHIR データストアを一覧表示するには

次のlist-fhir-datastores例は、コマンドの使用方法和、ユーザーが Amazon の Data Store ステータスに基づいて結果をフィルタリングする方法を示しています HealthLake。


```
aws healthlake list-fhir-datastores \  
  --region us-east-1 \  
  --filter DatastoreStatus=ACTIVE
```

出力:

```
{  
  "DatastorePropertiesList": [  
    {  
      "PreloadDataConfig": {  
        "PreloadDataType": "SYNTHEA"  
      },  
      "DatastoreName": "FhirTestDatastore",  
      "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/  
<Datastore ID>",  
      "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
<Datastore ID>/r4/",  
      "DatastoreStatus": "ACTIVE",  
      "DatastoreTypeVersion": "R4",  
      "CreatedAt": 1605574003.209,  
      "DatastoreId": "<Datastore ID>"  
    },  
    {  
      "DatastoreName": "Demo",  
      "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/  
<Datastore ID>",  
      "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
<Datastore ID>/r4/",  
      "DatastoreStatus": "ACTIVE",  
      "DatastoreTypeVersion": "R4",  
      "CreatedAt": 1603761064.881,  
      "DatastoreId": "<Datastore ID>"  
    }  
  ]  
}
```

詳細については、「Amazon HealthLake デベロッパーガイド」の[FHIR「データストアの作成とモニタリング」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListFhirDatastores](#)」の「」を参照してください。

list-fhir-export-jobs

次のコード例は、list-fhir-export-jobs を使用する方法を示しています。

AWS CLI

すべてのFHIRエクスポートジョブを一覧表示するには

次のlist-fhir-export-jobs例は、コマンドを使用して、アカウントに関連付けられたエクスポートジョブのリストを表示する方法を示しています。

```
aws healthlake list-fhir-export-jobs \  
  --datastore-id (Datastore ID) \  
  --submitted-before (DATE Like 2024-10-13T19:00:00Z) \  
  --submitted-after (DATE Like 2020-10-13T19:00:00Z) \  
  --job-name "FHIR-EXPORT" \  
  --job-status SUBMITTED \  
  --max-results (Integer between 1 and 500)
```

出力:

```
{  
  "ExportJobProperties": {  
    "OutputDataConfig": {  
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"  
      "S3Configuration": {  
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",  
        "KmsKeyId" : "(KmsKey Id)"  
      },  
    },  
  },  
  "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",  
  "JobStatus": "COMPLETED",  
  "JobId": "c145fbb27b192af392f8ce6e7838e34f",  
  "JobName": "FHIR-EXPORT",  
  "SubmitTime": 1606272542.161,  
  "EndTime": 1606272609.497,  
  "DatastoreId": "(Datastore ID)"  
}  
"NextToken": String
```

詳細については、「Amazon HealthLake デベロッパーガイド」の[FHIR「データストアからのファイルのエクスポート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFhirExportJobs](#)」の「」を参照してください。

list-fhir-import-jobs

次の例は、list-fhir-import-jobs を使用する方法を説明しています。

AWS CLI

すべてのFHIRインポートジョブを一覧表示するには

次のlist-fhir-import-jobs例は、コマンドを使用して、アカウントに関連付けられているすべてのインポートジョブのリストを表示する方法を示しています。

```
aws healthlake list-fhir-import-jobs \  
  --datastore-id (Datastore ID) \  
  --submitted-before (DATE Like 2024-10-13T19:00:00Z) \  
  --submitted-after (DATE Like 2020-10-13T19:00:00Z) \  
  --job-name "FHIR-IMPORT" \  
  --job-status SUBMITTED \  
  --max-results (Integer between 1 and 500)
```

出力:

```
{  
  "ImportJobProperties": {  
    "OutputDataConfig": {  
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",  
      "S3Configuration": {  
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",  
        "KmsKeyId" : "(KmsKey Id)"  
      },  
    },  
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",  
    "JobStatus": "COMPLETED",  
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",  
    "JobName": "FHIR-IMPORT",  
    "SubmitTime": 1606272542.161,  
    "EndTime": 1606272609.497,  
    "DatastoreId": "(Datastore ID)"  
  }  
}
```

```
"NextToken": String
```

詳細については、Amazon HealthLake デベロッパーガイドの[FHIR 「Data Store へのファイルのインポート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFhirImportJobs](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

データストアのタグを一覧表示するには

次のlist-tags-for-resource例では、指定された Data Store に関連付けられたタグを一覧表示します。

```
aws healthlake list-tags-for-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/  
  fhir/0725c83f4307f263e16fd56b6d8ebdbe" \  
  --region us-east-1
```

出力:

```
{  
  "tags": {  
    "key": "value",  
    "key1": "value1"  
  }  
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのリソースのタグ付け HealthLake」](#)を参照してください。 HealthLake

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

start-fhir-export-job

次のコード例は、start-fhir-export-job を使用する方法を示しています。

AWS CLI

FHIRエクスポートジョブを開始するには

次のstart-fhir-export-job例は、Amazon を使用してFHIRエクスポートジョブを開始する方法を示しています HealthLake。

```
aws healthlake start-fhir-export-job \  
  --output-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \  
  --datastore-id (Datastore ID) \  
  --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)
```

出力:

```
{  
  "DatastoreId": "(Datastore ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "9b9a51943afaedd0a8c0c26c49135a31"  
}
```

詳細については、「[Amazon HealthLake デベロッパーガイド](#)」のFHIR「[データストアからのファイルのエクスポート](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスStartFhirExportJob](#)」の「」を参照してください。

start-fhir-import-job

次のコード例は、start-fhir-import-job を使用する方法を示しています。

AWS CLI

FHIRインポートジョブを開始するには

次のstart-fhir-import-job例は、Amazon を使用してFHIRインポートジョブを開始する方法を示しています HealthLake。

```
aws healthlake start-fhir-import-job \  
  --input-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \  
  --datastore-id (Datastore ID) \  
  --data-access-role-arn "arn:aws:iam::(AWS Account ID):role/(Role Name)" \  
  --output-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \  
  --datastore-id (Datastore ID) \  
  --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)
```

```
--region us-east-1
```

出力:

```
{  
  "DatastoreId": "(Datastore ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "c145fbb27b192af392f8ce6e7838e34f"  
}
```

詳細については、「Amazon HealthLake デベロッパーガイド」の FHIR 「Data Store 'https://docs.aws.amazon.com/healthlake/latest/devguide/import-datastore.html へのファイルのインポート」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartFhirImportJob](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

Data Store にタグを追加するには

次の tag-resource 例は、データストアにタグを追加する方法を示しています。

```
aws healthlake tag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:691207106566:datastore/  
fhir/0725c83f4307f263e16fd56b6d8ebd8e" \  
  --tags '[{"Key": "key1", "Value": "value1"}]' \  
  --region us-east-1
```

このコマンドでは何も出力されません。

詳細については、「Amazon デベロッパーガイド」の「Data Store へのタグの追加」 <<https://docs.aws.amazon.com/healthlake/latest/devguide/add-a-tag.html>>」を参照してください。

HealthLake

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

データストアからタグを削除するには。

次のuntag-resource例は、データストアからタグを削除する方法を示しています。

```
aws healthlake untag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/fhir/  
b91723d65c6fdeb1d26543a49d2ed1fa" \  
  --tag-keys '["key1"]' \  
  --region us-east-1
```

このコマンドでは何も出力されません。

詳細については、「Amazon デベロッパーガイド」の「[データストアからのタグの削除](#)」を参照してください。 HealthLake

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

HealthOmics を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています HealthOmics。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

abort-multipart-read-set-upload

次のコード例は、`abort-multipart-read-set-upload` を使用する方法を示しています。

AWS CLI

マルチパートリードセットのアップロードを停止するには

次の`abort-multipart-read-set-upload`例では、HealthOmics シーケンスストアへのマルチパートリードセットのアップロードを停止します。

```
aws omics abort-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の [「シーケンスストアへの直接アップロード」](#) を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス [AbortMultipartReadSetUpload](#)」の「」を参照してください。

accept-share

次の例は、`accept-share` を使用する方法を説明しています。

AWS CLI

分析ストアデータの共有を受け入れるには

次の`accept-share`例では、HealthOmics 分析ストアデータの一部を受け入れます。

```
aws omics accept-share \  
  ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

出力:

```
{
```



```
"status": "ACTIVATING"
}
```

詳細については、「ユーザーガイド」の「[クロスアカウント共有](#)」を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptShare](#)」の「」を参照してください。

batch-delete-read-set

次のコード例は、batch-delete-read-set を使用する方法を示しています。

AWS CLI

複数のリードセットを削除するには

次のbatch-delete-read-set例では、2 つのリードセットを削除します。

```
aws omics batch-delete-read-set \
  --sequence-store-id 1234567890 \
  --ids 1234567890 0123456789
```

指定されたリードセットの削除にエラーがある場合、サービスはエラーリストを返します。

```
{
  "errors": [
    {
      "code": "",
      "id": "0123456789",
      "message": "The specified readset does not exist."
    }
  ]
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDeleteReadSet](#)」の「」を参照してください。

cancel-annotation-import-job

次の例は、cancel-annotation-import-job を使用する方法を説明しています。

AWS CLI

注釈インポートジョブをキャンセルするには

次のcancel-annotation-import-job例では、ID が の注釈インポートジョブをキャンセルします04f57618-xmpl-4fd0-9349-e5a85aefb997。

```
aws omics cancel-annotation-import-job \  
  --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelAnnotationImportJob](#)」の「」を参照してください。

cancel-run

次のコード例は、cancel-run を使用する方法を示しています。

AWS CLI

実行をキャンセルするには

次のcancel-run例では、ID で実行をキャンセルします1234567。

```
aws omics cancel-run \  
  --id 1234567
```

詳細については、「[Amazon Omics デベロッパーガイド](#)」の「Omics ワークフロー」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelRun](#)」の「」を参照してください。

cancel-variant-import-job

次の例は、cancel-variant-import-job を使用する方法を説明しています。

AWS CLI

バリエントインポートジョブをキャンセルするには

次のcancel-variant-import-job例では、ID が のバリエントインポートジョブをキャンセルします69cb65d6-xmpl-4a4a-9025-4565794b684e。

```
aws omics cancel-variant-import-job \  
  --job-id 69cb65d6-xmpl-4a4a-9025-4565794b684e
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelVariantImportJob](#)」の「」を参照してください。

complete-multipart-read-set-upload

次の例は、complete-multipart-read-set-upload を使用する方法を説明しています。

AWS CLI

すべてのコンポーネントをアップロードしたら、マルチパートアップロードを終了します。

次のcomplete-multipart-read-set-upload例では、すべてのコンポーネントがアップロードされると、シーケンスストアへのマルチパートアップロードを終了します。

```
aws omics complete-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --parts '["checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/  
Vobh5zBDb4W4=", "partNumber":1, "partSource":"SOURCE1"]'
```

出力:

```
{  
  "readSetId": "0000000001"  
  "readSetId": "0000000002"  
  "readSetId": "0000000003"  
}
```

詳細については、「ユーザーガイド」の[「シーケンスストアへの直接アップロード」](#)を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[CompleteMultipartReadSetUpload](#)」の「」を参照してください。

create-annotation-store-version

次のコード例は、create-annotation-store-version を使用する方法を示しています。

AWS CLI

アノテーションストアの新しいバージョンを作成するには

次のcreate-annotation-store-version例では、アノテーションストアの新しいバージョンを作成します。

```
aws omics create-annotation-store-version \  
  --name my_annotation_store \  
  --version-name my_version
```

出力:

```
{  
  "creationTime": "2023-07-21T17:15:49.251040+00:00",  
  "id": "3b93cdef69d2",  
  "name": "my_annotation_store",  
  "reference": {  
    "referenceArn": "arn:aws:omics:us-west-2:555555555555:referenceStore/6505293348/reference/5987565360"  
  },  
  "status": "CREATING",  
  "versionName": "my_version"  
}
```

詳細については、AWS HealthOmics 「ユーザーガイド」の[「注釈ストアの新しいバージョンの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAnnotationStoreVersion](#)」の「」を参照してください。

create-annotation-store

次のコード例は、create-annotation-store を使用する方法を示しています。

AWS CLI

例 1: VCF 注釈ストアを作成するには

次のcreate-annotation-store例では、VCFフォーマット注釈ストアを作成します。

```
aws omics create-annotation-store \  
  --name my_ann_store \  
  --store-format VCF \  
  --reference referenceArn=arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

出力:

```
{  
  "creationTime": "2022-11-23T22:48:39.226492Z",  
  "id": "0a91xmplc71f",  
  "name": "my_ann_store",  
  "reference": {  
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"  
  },  
  "status": "CREATING",  
  "storeFormat": "VCF"  
}
```

例 2: TSV 注釈ストアを作成するには

次のcreate-annotation-store例では、TSVフォーマット注釈ストアを作成します。

```
aws omics create-annotation-store \  
  --name tsv_ann_store \  
  --store-format TSV \  
  --reference referenceArn=arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890 \  
  --store-options file://tsv-store-options.json
```

tsv-store-options.json は注釈の形式オプションを設定します。

```
{
  "tsvStoreOptions": {
    "annotationType": "CHR_START_END_ZERO_BASE",
    "formatToHeader": {
      "CHR": "chromosome",
      "START": "start",
      "END": "end"
    },
    "schema": [
      {
        "chromosome": "STRING"
      },
      {
        "start": "LONG"
      },
      {
        "end": "LONG"
      },
      {
        "name": "STRING"
      }
    ]
  }
}
```

出力:

```
{
  "creationTime": "2022-11-30T01:28:08.525586Z",
  "id": "861cxmpl96b0",
  "name": "tsv_ann_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
  },
  "status": "CREATING",
  "storeFormat": "TSV",
  "storeOptions": {
    "tsvStoreOptions": {
      "annotationType": "CHR_START_END_ZERO_BASE",
      "formatToHeader": {
        "CHR": "chromosome",
        "END": "end",
```

```
        "START": "start"
      },
      "schema": [
        {
          "chromosome": "STRING"
        },
        {
          "start": "LONG"
        },
        {
          "end": "LONG"
        },
        {
          "name": "STRING"
        }
      ]
    }
  }
}
```

詳細については、Amazon [Omics デベロッパーガイドの「Omics Analytics」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAnnotationStore](#)」の「」を参照してください。

create-multipart-read-set-upload

次の例は、create-multipart-read-set-upload を使用する方法を説明しています。

AWS CLI

マルチパートリードセットのアップロードを開始するには。

次のcreate-multipart-read-set-upload例では、マルチパートリードセットのアップロードを開始します。

```
aws omics create-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --name HG00146 \  
  --source-file-type FASTQ \  
  --subject-id mySubject\  
  --sample-id mySample\  

```

```
--description "FASTQ for HG00146"\  
--generated-from "1000 Genomes"
```

出力:

```
{  
  "creationTime": "2022-07-13T23:25:20Z",  
  "description": "FASTQ for HG00146",  
  "generatedFrom": "1000 Genomes",  
  "name": "HG00146",  
  "sampleId": "mySample",  
  "sequenceStoreId": "0123456789",  
  "sourceFileType": "FASTQ",  
  "subjectId": "mySubject",  
  "uploadId": "1122334455"  
}
```

詳細については、「ユーザーガイド」の[「シーケンスストアへの直接アップロード」](#)を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateMultipartReadSetUpload](#)」の「」を参照してください。

create-reference-store

次の例は、create-reference-store を使用する方法を説明しています。

AWS CLI

リファレンスストアを作成するには

次のcreate-reference-store例では、リファレンスストアを作成しますmy-ref-store。

```
aws omics create-reference-store \  
  --name my-ref-store
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",  
  "creationTime": "2022-11-22T22:13:25.947Z",  
  "id": "1234567890",
```



```
"name": "my-ref-store"
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateReferenceStore](#)」の「」を参照してください。

create-run-group

次のコード例は、create-run-group を使用する方法を示しています。

AWS CLI

実行グループを作成するには

次のcreate-run-group例では、 という名前の実行グループを作成しますcram-converter。

```
aws omics create-run-group \
  --name cram-converter \
  --max-cpus 20 \
  --max-duration 600
```

出力:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
  "id": "1234567",
  "tags": {}
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics ワークフロー」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRunGroup](#)」の「」を参照してください。

create-sequence-store

次の例は、create-sequence-store を使用する方法を説明しています。

AWS CLI

シーケンスストアを作成するには

次のcreate-sequence-store例では、シーケンスストアを作成します。

```
aws omics create-sequence-store \  
  --name my-seq-store
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",  
  "creationTime": "2022-11-23T01:24:33.629Z",  
  "id": "1234567890",  
  "name": "my-seq-store"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSequenceStore](#)」の「」を参照してください。

create-share

次の例は、create-share を使用する方法を説明しています。

AWS CLI

HealthOmics 分析ストアの共有を作成するには

次のcreate-share例は、アカウント外のサブスクライバーが受け入れることができる HealthOmics 分析ストアの共有を作成する方法を示しています。

```
aws omics create-share \  
  --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/  
omics_dev_var_store" \  
  --principal-subscriber "123456789012" \  
  --name "my_Share-123"
```

出力:

```
{
  "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
  "name": "my_Share-123",
  "status": "PENDING"
}
```

詳細については、「ユーザーガイド」の「[クロスアカウント共有](#)」を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateShare](#)」の「」を参照してください。

create-variant-store

次の例は、create-variant-store を使用する方法を説明しています。

AWS CLI

バリエーションストアを作成するには

次のcreate-variant-store例では、という名前のバリエーションストアを作成しますmy_var_store。

```
aws omics create-variant-store \
  --name my_var_store \
  --reference referenceArn=arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

出力:

```
{
  "creationTime": "2022-11-23T22:09:07.534499Z",
  "id": "02dexplcfdd",
  "name": "my_var_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
  },
  "status": "CREATING"
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVariantStore](#)」の「」を参照してください。

create-workflow

次のコード例は、create-workflow を使用する方法を示しています。

AWS CLI

ワークフローを作成するには

次のcreate-workflow例では、WDLワークフローを作成します。

```
aws omics create-workflow \  
  --name cram-converter \  
  --engine WDL \  
  --definition-zip fileb://workflow-crambam.zip \  
  --parameter-template file://workflow-params.json
```

workflow-crambam.zip は、ワークフロー定義を含むZIPアーカイブです。は、ワークフローのランタイムパラメータworkflow-params.jsonを定義します。

```
{  
  "ref_fasta" : {  
    "description": "Reference genome fasta file",  
    "optional": false  
  },  
  "ref_fasta_index" : {  
    "description": "Index of the reference genome fasta file",  
    "optional": false  
  },  
  "ref_dict" : {  
    "description": "dictionary file for 'ref_fasta'",  
    "optional": false  
  },  
  "input_cram" : {  
    "description": "The Cram file to convert to BAM",  
    "optional": false  
  },  
}
```

```
"sample_name" : {
  "description": "The name of the input sample, used to name the output BAM",
  "optional": false
}
}
```

出力:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
  "id": "1234567",
  "status": "CREATING",
  "tags": {}
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateWorkflow](#)」の「」を参照してください。

delete-annotation-store-versions

次のコード例は、delete-annotation-store-versions を使用する方法を示しています。

AWS CLI

注釈ストアのバージョンを削除するには

次のdelete-annotation-store-versions例では、注釈ストアバージョンを削除します。

```
aws omics delete-annotation-store-versions \
  --name my_annotation_store \
  --versions my_version
```

出力:

```
{
  "errors": []
}
```

詳細については、AWS HealthOmics 「ユーザーガイド」の「[注釈ストアの新しいバージョンの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAnnotationStoreVersions](#)」の「」を参照してください。

delete-annotation-store

次の例は、delete-annotation-store を使用する方法を説明しています。

AWS CLI

注釈ストアを削除するには

次のdelete-annotation-store例では、 という名前の注釈ストアを削除しますmy_vcf_store。

```
aws omics delete-annotation-store \  
  --name my_vcf_store
```

出力:

```
{  
  "status": "DELETING"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAnnotationStore](#)」の「」を参照してください。

delete-reference-store

次のコード例は、delete-reference-store を使用する方法を示しています。

AWS CLI

リファレンスストアを削除するには

次のdelete-reference-store例では、ID が のリファレンスストアを削除します1234567890。

```
aws omics delete-reference-store \  
  --id 1234567890
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteReferenceStore](#)」の「」を参照してください。

delete-reference

次のコード例は、delete-reference を使用する方法を示しています。

AWS CLI

リファレンスを削除するには

次のdelete-reference例では、リファレンスを削除します。

```
aws omics delete-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteReference](#)」の「」を参照してください。

delete-run-group

次の例は、delete-run-group を使用する方法を説明しています。

AWS CLI

実行グループを削除するには

次のdelete-run-group例では、ID のラングループを削除します1234567。

```
aws omics delete-run-group \  
  --id 1234567
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRunGroup](#)」の「」を参照してください。

delete-run

次のコード例は、delete-run を使用する方法を示しています。

AWS CLI

ワークフロー実行を削除するには

次のdelete-run例では、ID で実行を削除します1234567。

```
aws omics delete-run \  
  --id 1234567
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRun](#)」の「」を参照してください。

delete-sequence-store

次の例は、delete-sequence-store を使用する方法を説明しています。

AWS CLI

シーケンスストアを削除するには

次のdelete-sequence-store例では、ID が のシーケンスストアを削除します1234567890。

```
aws omics delete-sequence-store \  
  --id 1234567890
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSequenceStore](#)」の「」を参照してください。

delete-share

次の例は、delete-share を使用する方法を説明しています。

AWS CLI

HealthOmics 分析データの共有を削除するには

次のdelete-share例では、分析データのクロスアカウント共有を削除します。

```
aws omics delete-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

出力:

```
{  
  "status": "DELETING"  
}
```

詳細については、「ユーザーガイド」の「[クロスアカウント共有](#)」を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteShare](#)」の「」を参照してください。

delete-variant-store

次のコード例は、delete-variant-store を使用する方法を示しています。

AWS CLI

バリエーションストアを削除するには

次のdelete-variant-store例では、という名前のバリエーションストアを削除しますmy_var_store。

```
aws omics delete-variant-store \  
  --name my_var_store
```

出力:

```
{
  "status": "DELETING"
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVariantStore](#)」の「」を参照してください。

delete-workflow

次の例は、delete-workflow を使用する方法を説明しています。

AWS CLI

ワークフローを削除するには

次のdelete-workflow例では、ID が のワークフローを削除します1234567。

```
aws omics delete-workflow \  
  --id 1234567
```

詳細については、「[Amazon Omics デベロッパーガイド](#)」の「Omics ワークフロー」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteWorkflow](#)」の「」を参照してください。

get-annotation-import-job

次の例は、get-annotation-import-job を使用する方法を説明しています。

AWS CLI

注釈インポートジョブを表示するには

次のget-annotation-import-job例では、注釈インポートジョブの詳細を取得します。

```
aws omics get-annotation-import-job \  
  --id 1234567
```

```
--job-id 984162c7-xmpl-4d23-ab47-286f7950bfbf
```

出力:

```
{
  "creationTime": "2022-11-30T01:40:11.017746Z",
  "destinationName": "tsv_ann_store",
  "id": "984162c7-xmpl-4d23-ab47-286f7950bfbf",
  "items": [
    {
      "jobStatus": "COMPLETED",
      "source": "s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz"
    }
  ],
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "runLeftNormalization": false,
  "status": "COMPLETED",
  "updateTime": "2022-11-30T01:42:39.134009Z"
}
```

詳細については、「Amazon [Omics](#) デベロッパーガイド」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAnnotationImportJob](#)」の「」を参照してください。

get-annotation-store-version

次のコード例は、get-annotation-store-version を使用する方法を示しています。

AWS CLI

注釈ストアバージョンのメタデータを取得するには

次のget-annotation-store-version例では、リクエストされたアノテーションストアバージョンのメタデータを取得します。

```
aws omics get-annotation-store-version \
  --name my_annotation_store \
  --version-name my_version
```

出力:

```
{
  "storeId": "4934045d1c6d",
  "id": "2a3f4a44aa7b",
  "status": "ACTIVE",
  "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/
my_annotation_store/version/my_version",
  "name": "my_annotation_store",
  "versionName": "my_version",
  "creationTime": "2023-07-21T17:15:49.251040+00:00",
  "updateTime": "2023-07-21T17:15:56.434223+00:00",
  "statusMessage": "",
  "versionSizeBytes": 0
}
```

詳細については、AWS HealthOmics 「ユーザーガイド」の「[注釈ストアの新しいバージョンの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAnnotationStoreVersion](#)」の「」を参照してください。

get-annotation-store

次の例は、get-annotation-store を使用する方法を説明しています。

AWS CLI

注釈ストアを表示するには

次のget-annotation-store例では、`my_ann_store` という名前の注釈ストアの詳細を取得します。

```
aws omics get-annotation-store \
  --name my_ann_store
```

出力:

```
{
  "creationTime": "2022-11-23T22:48:39.226492Z",
  "id": "0a91xmplc71f",
  "name": "my_ann_store",
}
```

```
"reference": {
  "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
},
"status": "CREATING",
"storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/my_ann_store",
"storeFormat": "VCF",
"storeSizeBytes": 0,
"tags": {}
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAnnotationStore](#)」の「」を参照してください。

get-read-set-activation-job

次の例は、get-read-set-activation-job を使用する方法を説明しています。

AWS CLI

リードセットアクティベーションジョブを表示するには

次のget-read-set-activation-job例では、リードセットアクティベーションジョブの詳細を取得します。

```
aws omics get-read-set-activation-job \
  --sequence-store-id 1234567890 \
  --id 1234567890
```

出力:

```
{
  "completionTime": "2022-12-06T22:33:42.828Z",
  "creationTime": "2022-12-06T22:32:45.213Z",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "sources": [
    {
      "readSetId": "1234567890",
```

```
        "status": "FINISHED",
        "statusMessage": "No activation needed as read set is already in
ACTIVATING or ACTIVE state."
    }
],
"status": "COMPLETED",
"statusMessage": "The job completed successfully."
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReadSetActivationJob](#)」の「」を参照してください。

get-read-set-export-job

次のコード例は、get-read-set-export-job を使用する方法を示しています。

AWS CLI

リードセットエクスポートジョブを表示するには

次のget-read-set-export-job例では、リードセットエクスポートジョブの詳細を取得します。

```
aws omics get-read-set-export-job \
  --sequence-store-id 1234567890 \
  --id 1234567890
```

出力:

```
{
  "completionTime": "2022-12-06T22:39:14.491Z",
  "creationTime": "2022-12-06T22:37:18.612Z",
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "COMPLETED",
  "statusMessage": "The job is submitted and will start soon."
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReadSetExportJob](#)」の「」を参照してください。

get-read-set-import-job

次のコード例は、get-read-set-import-job を使用する方法を示しています。

AWS CLI

リードセットのインポートジョブを表示するには

次のget-read-set-import-job例では、リードセットのインポートジョブの詳細を取得します。

```
aws omics get-read-set-import-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

出力:

```
{  
  "creationTime": "2022-11-23T01:36:38.158Z",  
  "id": "1234567890",  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "sequenceStoreId": "1234567890",  
  "sources": [  
    {  
      "name": "HG00100",  
      "referenceArn": "arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890",  
      "sampleId": "bam-sample",  
      "sourceFileType": "BAM",  
      "sourceFiles": {  
        "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/  
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam",  
        "source2": ""  
      },  
      "status": "IN_PROGRESS",  
      "statusMessage": "The source job is currently in progress."  
    }  
  ]  
}
```

```
    "subjectId": "bam-subject",
    "tags": {
      "aws:omics:sampleId": "bam-sample",
      "aws:omics:subjectId": "bam-subject"
    }
  },
  {
    "name": "HG00146",
    "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
    "sampleId": "fastq-sample",
    "sourceFileType": "FASTQ",
    "sourceFiles": {
      "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_1.filt.fastq.gz",
      "source2": "s3://omics-artifacts-01d6xmpl4e72dd32/
SRR233106_2.filt.fastq.gz"
    },
    "status": "IN_PROGRESS",
    "statusMessage": "The source job is currently in progress.",
    "subjectId": "fastq-subject",
    "tags": {
      "aws:omics:sampleId": "fastq-sample",
      "aws:omics:subjectId": "fastq-subject"
    }
  },
  {
    "name": "HG00096",
    "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890",
    "sampleId": "cram-sample",
    "sourceFileType": "CRAM",
    "sourceFiles": {
      "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/
HG00096.alt_bwamem_GRCh38DH.20150718.GBR.low_coverage.cram",
      "source2": ""
    },
    "status": "IN_PROGRESS",
    "statusMessage": "The source job is currently in progress.",
    "subjectId": "cram-subject",
    "tags": {
      "aws:omics:sampleId": "cram-sample",
      "aws:omics:subjectId": "cram-subject"
    }
  }
}
```



```
    }
  ],
  "status": "IN_PROGRESS",
  "statusMessage": "The job is currently in progress."
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReadSetImportJob](#)」の「」を参照してください。

get-read-set-metadata

次の例は、get-read-set-metadata を使用する方法を説明しています。

AWS CLI

リードセットを表示するには

次のget-read-set-metadata例では、リードセットのファイルの詳細を取得します。

```
aws omics get-read-set-metadata \
  --sequence-store-id 1234567890 \
  --id 1234567890
```

出力:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/
readSet/1234567890",
  "creationTime": "2022-11-23T21:55:00.515Z",
  "fileType": "FASTQ",
  "files": {
    "source1": {
      "contentLength": 310054739,
      "partSize": 104857600,
      "totalParts": 3
    },
    "source2": {
      "contentLength": 307846621,
      "partSize": 104857600,

```

```
        "totalParts": 3
      }
    },
    "id": "1234567890",
    "name": "HG00146",
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
    "sampleId": "fastq-sample",
    "sequenceInformation": {
      "alignment": "UNALIGNED",
      "totalBaseCount": 677717384,
      "totalReadCount": 8917334
    },
    "sequenceStoreId": "1234567890",
    "status": "ACTIVE",
    "subjectId": "fastq-subject"
  }
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReadSetMetadata](#)」の「」を参照してください。

get-read-set

次のコード例は、get-read-set を使用する方法を示しています。

AWS CLI

リードセットをダウンロードするには

次のget-read-set例では、リードセットのパート 3 をとしてダウンロードします1234567890.3.bam。

```
aws omics get-read-set \
  --sequence-store-id 1234567890 \
  --id 1234567890 \
  --part-number 3 1234567890.3.bam
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReadSet](#)」の「」を参照してください。

get-reference-import-job

次の例は、get-reference-import-job を使用する方法を説明しています。

AWS CLI

リファレンスインポートジョブを表示するには

次のget-reference-import-job例では、リファレンスインポートジョブの詳細を取得します。

```
aws omics get-reference-import-job \
  --reference-store-id 1234567890 \
  --id 1234567890
```

出力:

```
{
  "creationTime": "2022-11-22T22:25:41.124Z",
  "id": "1234567890",
  "referenceStoreId": "1234567890",
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "sources": [
    {
      "name": "assembly-38",
      "sourceFile": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.fasta",
      "status": "IN_PROGRESS",
      "statusMessage": "The source job is currently in progress."
    }
  ],
  "status": "IN_PROGRESS",
  "statusMessage": "The job is currently in progress."
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReferenceImportJob](#)」の「」を参照してください。

get-reference-metadata

次の例は、get-reference-metadata を使用する方法を説明しています。

AWS CLI

リファレンスを表示するには

次のget-reference-metadata例では、リファレンスの詳細を取得します。

```
aws omics get-reference-metadata \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/  
reference/1234567890",  
  "creationTime": "2022-11-22T22:27:09.033Z",  
  "files": {  
    "index": {  
      "contentLength": 160928,  
      "partSize": 104857600,  
      "totalParts": 1  
    },  
    "source": {  
      "contentLength": 3249912778,  
      "partSize": 104857600,  
      "totalParts": 31  
    }  
  },  
  "id": "1234567890",  
  "md5": "7ff134953dcca8c8997453bbb80b6b5e",  
  "name": "assembly-38",  
  "referenceStoreId": "1234567890",  
  "status": "ACTIVE",  
  "updateTime": "2022-11-22T22:27:09.033Z"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReferenceMetadata](#)」の「」を参照してください。

get-reference-store

次の例は、get-reference-store を使用する方法を説明しています。

AWS CLI

リファレンスストアを表示するには

次のget-reference-store例では、リファレンスストアの詳細を取得します。

```
aws omics get-reference-store \  
  --id 1234567890
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",  
  "creationTime": "2022-09-23T23:27:20.364Z",  
  "id": "1234567890",  
  "name": "my-rstore-0"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReferenceStore](#)」の「」を参照してください。

get-reference

次のコード例は、get-reference を使用する方法を示しています。

AWS CLI

ゲノムリファレンスをダウンロードするには

次のget-reference例では、ゲノムのパート 1 をとしてダウンロードしますhg38.1.fa。

```
aws omics get-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890 \  
  --part-number 1 hg38.1.fa
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReference](#)」の「」を参照してください。

get-run-group

次の例は、get-run-group を使用する方法を説明しています。

AWS CLI

実行グループを表示するには

次のget-run-group例では、実行グループの詳細を取得します。

```
aws omics get-run-group \  
  --id 1234567
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",  
  "creationTime": "2022-12-01T00:58:42.915219Z",  
  "id": "1234567",  
  "maxCpus": 20,  
  "maxDuration": 600,  
  "name": "cram-convert",  
  "tags": {}  
}
```

詳細については、「Amazon Omics [デベロッパーガイド](#)」の「Omics ワークフロー」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRunGroup](#)」の「」を参照してください。

get-run-task

次のコード例は、get-run-task を使用する方法を示しています。

AWS CLI

タスクを表示するには

次のget-run-task例では、ワークフロータスクの詳細を取得します。

```
aws omics get-run-task \  
  --id 1234567 \  
  --task-id 1234567
```

出力:

```
{  
  "cpus": 1,  
  "creationTime": "2022-11-30T23:13:00.718651Z",  
  "logStream": "arn:aws:logs:us-west-2:123456789012:log-group:/aws/omics/  
WorkflowLog:log-stream:run/1234567/task/1234567",  
  "memory": 15,  
  "name": "CramToBamTask",  
  "startTime": "2022-11-30T23:17:47.016Z",  
  "status": "COMPLETED",  
  "stopTime": "2022-11-30T23:18:21.503Z",  
  "taskId": "1234567"  
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRunTask](#)」の「」を参照してください。

get-run

次のコード例は、get-run を使用する方法を示しています。

AWS CLI

ワークフロー実行を表示するには

次の`get-run`例では、ワークフローの実行に関する詳細を取得します。

```
aws omics get-run \  
  --id 1234567
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",  
  "creationTime": "2022-11-30T22:58:22.615865Z",  
  "digest":  
  "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",  
  "id": "1234567",  
  "name": "cram-to-bam",  
  "outputUri": "s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/",  
  "parameters": {  
    "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.dict",  
    "ref_fasta_index": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta.fai",  
    "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta",  
    "sample_name": "NA12878",  
    "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram"  
  },  
  "resourceDigests": {  
    "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta.fai":  
"etag:f76371b113734a56cde236bc0372de0a",  
    "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.dict":  
"etag:3884c62eb0e53fa92459ed9bfff133ae6",  
    "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta":  
"etag:e307d81c605fb91b7720a08f00276842-388",  
    "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram":  
"etag:a9f52976381286c6143b5cc681671ec6"  
  },  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "startedBy": "arn:aws:iam::123456789012:user/laptop-2020",  
  "status": "STARTING",  
}
```



```
"tags": {},  
"workflowId": "1234567",  
"workflowType": "PRIVATE"  
}
```

詳細については、[「Amazon Omics デベロッパーガイド」](#)の「Omics ワークフロー」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRun](#)」の「」を参照してください。

get-sequence-store

次の例は、get-sequence-store を使用する方法を説明しています。

AWS CLI

シーケンスストアを表示するには

次のget-sequence-store例では、ID を持つシーケンスストアの詳細を取得します1234567890。

```
aws omics get-sequence-store \  
  --id 1234567890
```

出力:

```
{  
  "arn": "arn:aws:omics:us-east-1:123456789012:sequenceStore/1234567890",  
  "creationTime": "2022-11-23T19:55:48.376Z",  
  "id": "1234567890",  
  "name": "my-seq-store"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSequenceStore](#)」の「」を参照してください。

get-share

次のコード例は、get-share を使用する方法を示しています。

AWS CLI

HealthOmics 分析データの共有に関するメタデータを取得するには

次のget-share例では、分析データのクロスアカウント共有のメタデータを取得します。

```
aws omics get-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

出力:

```
{  
  "share": {  
    "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",  
    "name": "my_Share-123",  
    "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/  
omics_dev_var_store",  
    "principalSubscriber": "123456789012",  
    "ownerId": "555555555555",  
    "status": "PENDING"  
  }  
}
```

詳細については、「ユーザーガイド」の「[クロスアカウント共有](#)」を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[GetShare](#)」の「」を参照してください。

get-variant-import-job

次のコード例は、get-variant-import-job を使用する方法を示しています。

AWS CLI

バリエントインポートジョブを表示するには

次のget-variant-import-job例では、バリエーションインポートジョブの詳細を取得します。

```
aws omics get-variant-import-job \  
  --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

出力:

```
{  
  "creationTime": "2022-11-23T22:42:50.037812Z",  
  "destinationName": "my_var_store",  
  "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",  
  "items": [  
    {  
      "jobStatus": "IN_PROGRESS",  
      "source": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.known_indels.vcf.gz"  
    }  
  ],  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "runLeftNormalization": false,  
  "status": "IN_PROGRESS",  
  "updateTime": "2022-11-23T22:43:05.898309Z"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVariantImportJob](#)」の「」を参照してください。

get-variant-store

次のコード例は、get-variant-store を使用する方法を示しています。

AWS CLI

バリエーションストアを表示するには

次のget-variant-store例では、バリエーションストアの詳細を取得します。

```
aws omics get-variant-store \  
  --store-name my_var_store
```

```
--name my_var_store
```

出力:

```
{
  "creationTime": "2022-11-23T22:09:07.534499Z",
  "id": "02dexplcfdd",
  "name": "my_var_store",
  "reference": {
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
  },
  "status": "CREATING",
  "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/my_var_store",
  "storeSizeBytes": 0,
  "tags": {},
  "updateTime": "2022-11-23T22:09:24.931711Z"
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVariantStore](#)」の「」を参照してください。

get-workflow

次の例は、get-workflow を使用する方法を説明しています。

AWS CLI

ワークフローを表示するには

次のget-workflow例では、ID を持つワークフローの詳細を取得します1234567。

```
aws omics get-workflow \  
  --id 1234567
```

出力:

```
{
```

```
"arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
"creationTime": "2022-11-30T22:33:16.225368Z",
"digest":
"sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
"engine": "WDL",
"id": "1234567",
"main": "workflow-crambam.wdl",
"name": "cram-converter",
"parameterTemplate": {
  "ref_dict": {
    "description": "dictionary file for 'ref_fasta'"
  },
  "ref_fasta_index": {
    "description": "Index of the reference genome fasta file"
  },
  "ref_fasta": {
    "description": "Reference genome fasta file"
  },
  "input_cram": {
    "description": "The Cram file to convert to BAM"
  },
  "sample_name": {
    "description": "The name of the input sample, used to name the output
BAM"
  }
},
"status": "ACTIVE",
"statusMessage": "workflow-crambam.wdl\n      workflow CramToBamFlow\n
call CramToBamTask\n      call ValidateSamFile\n      task CramToBamTask\n      task
ValidateSamFile\n",
"tags": {},
"type": "PRIVATE"
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWorkflow](#)」の「」を参照してください。

list-annotation-import-jobs

次の例は、list-annotation-import-jobs を使用する方法を説明しています。

AWS CLI

注釈インポートジョブのリストを取得するには

以下は、注釈インポートジョブのリスト `list-annotation-import-jobs` を取得します。

```
aws omics list-annotation-import-jobs
```

出力:

```
{
  "annotationImportJobs": [
    {
      "creationTime": "2022-11-30T01:39:41.478294Z",
      "destinationName": "gff_ann_store",
      "id": "18a9e792-xmpl-4869-a105-e5b602900444",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "runLeftNormalization": false,
      "status": "COMPLETED",
      "updateTime": "2022-11-30T01:47:09.145178Z"
    },
    {
      "creationTime": "2022-11-30T00:45:58.007838Z",
      "destinationName": "my_ann_store",
      "id": "4e9eafc8-xmpl-431e-a0b2-3bda27cb600a",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "runLeftNormalization": false,
      "status": "FAILED",
      "updateTime": "2022-11-30T00:47:01.706325Z"
    }
  ]
}
```

詳細については、「Amazon [Omics デベロッパガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAnnotationImportJobs](#)」の「」を参照してください。

list-annotation-store-versions

次のコード例は、list-annotation-store-versions を使用する方法を示しています。

AWS CLI

注釈ストアのすべてのバージョンを一覧表示するには。

次のlist-annotation-store-versions例では、注釈ストアが存在するすべてのバージョンを一覧表示します。

```
aws omics list-annotation-store-versions \  
  --name my_annotation_store
```

出力:

```
{  
  "annotationStoreVersions": [  
    {  
      "storeId": "4934045d1c6d",  
      "id": "2a3f4a44aa7b",  
      "status": "CREATING",  
      "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/  
my_annotation_store/version/my_version_2",  
      "name": "my_annotation_store",  
      "versionName": "my_version_2",  
      "creationTime": "2023-07-21T17:20:59.380043+00:00",  
      "versionSizeBytes": 0  
    },  
    {  
      "storeId": "4934045d1c6d",  
      "id": "4934045d1c6d",  
      "status": "ACTIVE",  
      "versionArn": "arn:aws:omics:us-west-2:555555555555:annotationStore/  
my_annotation_store/version/my_version_1",  
      "name": "my_annotation_store",  
      "versionName": "my_version_1",  
      "creationTime": "2023-07-21T17:15:49.251040+00:00",  
      "updateTime": "2023-07-21T17:15:56.434223+00:00",  
      "statusMessage": "",  
      "versionSizeBytes": 0  
    }  
  ]  
}
```

```
}
```

詳細については、AWS HealthOmics 「ユーザーガイド」の「[注釈ストアの新しいバージョンの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAnnotationStoreVersions](#)」の「」を参照してください。

list-annotation-stores

次の例は、list-annotation-stores を使用する方法を説明しています。

AWS CLI

注釈ストアのリストを取得するには

次のlist-annotation-stores例では、注釈ストアのリストを取得します。

```
aws omics list-annotation-stores
```

出力:

```
{
  "annotationStores": [
    {
      "creationTime": "2022-11-23T22:48:39.226492Z",
      "id": "0a91xmplc71f",
      "name": "my_ann_store",
      "reference": {
        "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"
      },
      "status": "ACTIVE",
      "statusMessage": "",
      "storeArn": "arn:aws:omics:us-west-2:123456789012:annotationStore/my_ann_store",
      "storeFormat": "VCF",
      "storeSizeBytes": 0,
      "updateTime": "2022-11-23T22:53:27.372840Z"
    }
  ]
}
```


詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAnnotationStores](#)」の「」を参照してください。

list-multipart-read-set-uploads

次の例は、list-multipart-read-set-uploads を使用する方法を説明しています。

AWS CLI

すべてのマルチパートリードセットのアップロードとそのステータスを一覧表示するには。

次のlist-multipart-read-set-uploads例では、すべてのマルチパートリードセットのアップロードとそのステータスを一覧表示します。

```
aws omics list-multipart-read-set-uploads \  
  --sequence-store-id 0123456789
```

出力:

```
{  
  "uploads":  
    [  
      {  
        "sequenceStoreId": "0123456789",  
        "uploadId": "8749584421",  
        "sourceFileType": "FASTQ",  
        "subjectId": "mySubject",  
        "sampleId": "mySample",  
        "generatedFrom": "1000 Genomes",  
        "name": "HG00146",  
        "description": "FASTQ for HG00146",  
        "creationTime": "2023-11-29T19:22:51.349298+00:00"  
      },  
      {  
        "sequenceStoreId": "0123456789",  
        "uploadId": "5290538638",  
        "sourceFileType": "BAM",  
        "subjectId": "mySubject",  
        "sampleId": "mySample",  
        "generatedFrom": "1000 Genomes",
```

```
    "referenceArn": "arn:aws:omics:us-west-2:845448930428:referenceStore/8168613728/reference/2190697383",
    "name": "HG00146",
    "description": "BAM for HG00146",
    "creationTime": "2023-11-29T19:23:33.116516+00:00"
  },
  {
    "sequenceStoreId": "0123456789",
    "uploadId": "4174220862",
    "sourceFileType": "BAM",
    "subjectId": "mySubject",
    "sampleId": "mySample",
    "generatedFrom": "1000 Genomes",
    "referenceArn": "arn:aws:omics:us-west-2:845448930428:referenceStore/8168613728/reference/2190697383",
    "name": "HG00147",
    "description": "BAM for HG00147",
    "creationTime": "2023-11-29T19:23:47.007866+00:00"
  }
]
```

詳細については、「ユーザーガイド」の[「シーケンスストアへの直接アップロード」](#)を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMultipartReadSetUploads](#)」の「」を参照してください。

list-read-set-activation-jobs

次のコード例は、list-read-set-activation-jobs を使用する方法を示しています。

AWS CLI

リードセットアクティベーションジョブのリストを取得するには

次のlist-read-set-activation-jobs例では、ID が のシーケンスストアのアクティベーションジョブのリストを取得します1234567890。

```
aws omics list-read-set-activation-jobs \
  --sequence-store-id 1234567890
```

出力:

```
{
  "activationJobs": [
    {
      "completionTime": "2022-12-06T22:33:42.828Z",
      "creationTime": "2022-12-06T22:32:45.213Z",
      "id": "1234567890",
      "sequenceStoreId": "1234567890",
      "status": "COMPLETED"
    },
    {
      "creationTime": "2022-12-06T22:35:10.100Z",
      "id": "1234567890",
      "sequenceStoreId": "1234567890",
      "status": "IN_PROGRESS"
    }
  ]
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReadSetActivationJobs](#)」の「」を参照してください。

list-read-set-export-jobs

次のコード例は、list-read-set-export-jobs を使用する方法を示しています。

AWS CLI

リードセットエクスポートジョブのリストを取得するには

次のlist-read-set-export-jobs例では、ID が のシーケンスストアのエクスポートジョブのリストを取得します1234567890。

```
aws omics list-read-set-export-jobs \
  --sequence-store-id 1234567890
```

出力:

```
{
  "exportJobs": [
```

```
{
  "completionTime": "2022-12-06T22:39:14.491Z",
  "creationTime": "2022-12-06T22:37:18.612Z",
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "COMPLETED"
},
{
  "creationTime": "2022-12-06T22:38:04.871Z",
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "IN_PROGRESS"
}
]
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReadSetExportJobs](#)」の「」を参照してください。

list-read-set-import-jobs

次のコード例は、list-read-set-import-jobs を使用する方法を示しています。

AWS CLI

リードセットインポートジョブのリストを取得するには

次のlist-read-set-import-jobs例では、ID が のシーケンスストアのインポートジョブのリストを取得します1234567890。

```
aws omics list-read-set-import-jobs \
  --sequence-store-id 1234567890
```

出力:

```
{
  "importJobs": [
    {
```

```
    "completionTime": "2022-11-29T18:17:49.244Z",
    "creationTime": "2022-11-29T17:32:47.700Z",
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
    "sequenceStoreId": "1234567890",
    "status": "COMPLETED"
  },
  {
    "completionTime": "2022-11-23T22:01:34.090Z",
    "creationTime": "2022-11-23T21:52:43.289Z",
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
    "sequenceStoreId": "1234567890",
    "status": "COMPLETED_WITH_FAILURES"
  }
]
}
```

詳細については、「Amazon [Omics デベロッパークガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReadSetImportJobs](#)」の「」を参照してください。

list-read-set-upload-parts

次のコード例は、list-read-set-upload-parts を使用する方法を示しています。

AWS CLI

シーケンスストアのリクエストされたマルチパートアップロード内のすべてのパートを一覧表示するには。

次のlist-read-set-upload-parts例では、シーケンスストアのリクエストされたマルチパートアップロード内のすべてのパートを一覧表示します。

```
aws omics list-read-set-upload-parts \
  --sequence-store-id 0123456789 \
  --upload-id 1122334455 \
  --part-source SOURCE1
```

出力:

```
{
  "parts": [
    {
      "partNumber": 1,
      "partSize": 94371840,
      "file": "SOURCE1",
      "checksum":
"984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
      "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
    }
    {
      "partNumber": 2,
      "partSize": 10471840,
      "file": "SOURCE1",
      "checksum":
"984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635",
      "lastUpdatedTime": "2023-02-02T20:14:47.533000+00:00"
    }
  ]
}
```

詳細については、「ユーザーガイド」の[「シーケンスストアへの直接アップロード」](#)を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReadSetUploadParts](#)」の「」を参照してください。

list-read-sets

次のコード例は、list-read-sets を使用する方法を示しています。

AWS CLI

リードセットのリストを取得するには

次のlist-read-sets例では、ID が のシーケンスストアのリードセットのリストを取得します1234567890。

```
aws omics list-read-sets \
```

```
--sequence-store-id 1234567890
```

出力:

```
{
  "readSets": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890/readSet/1234567890",
      "creationTime": "2022-11-23T21:55:00.515Z",
      "fileType": "FASTQ",
      "id": "1234567890",
      "name": "HG00146",
      "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890",
      "sampleId": "fastq-sample",
      "sequenceStoreId": "1234567890",
      "status": "ACTIVE",
      "subjectId": "fastq-subject"
    }
  ]
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReadSets](#)」の「」を参照してください。

list-reference-import-jobs

次の例は、list-reference-import-jobs を使用する方法を説明しています。

AWS CLI

リファレンスインポートジョブのリストを取得するには

次のlist-reference-import-jobs例では、ID が のリファレンスストアのリファレンスインポートジョブのリストを取得します1234567890。

```
aws omics list-reference-import-jobs \
  --reference-store-id 1234567890
```

出力:

```
{
  "importJobs": [
    {
      "completionTime": "2022-11-23T19:54:58.204Z",
      "creationTime": "2022-11-23T19:53:20.729Z",
      "id": "1234567890",
      "referenceStoreId": "1234567890",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "status": "COMPLETED"
    },
    {
      "creationTime": "2022-11-23T20:34:03.250Z",
      "id": "1234567890",
      "referenceStoreId": "1234567890",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "status": "IN_PROGRESS"
    }
  ]
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReferenceImportJobs](#)」の「」を参照してください。

list-reference-stores

次の例は、list-reference-stores を使用する方法を説明しています。

AWS CLI

リファレンスストアのリストを取得するには

次のlist-reference-stores例では、リファレンスストアのリストを取得します。

```
aws omics list-reference-stores
```


出力:

```
{
  "referenceStores": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
      "creationTime": "2022-11-22T22:13:25.947Z",
      "id": "1234567890",
      "name": "my-ref-store"
    }
  ]
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReferenceStores](#)」の「」を参照してください。

list-references

次の例は、list-references を使用する方法を説明しています。

AWS CLI

リファレンスのリストを取得するには

次のlist-references例では、ID が のリファレンスストアのゲノムリファレンスのリストを取得します1234567890。

```
aws omics list-references \
  --reference-store-id 1234567890
```

出力:

```
{
  "references": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/
reference/1234567890",
      "creationTime": "2022-11-22T22:27:09.033Z",
```

```
    "id": "1234567890",
    "md5": "7ff134953dcca8c8997453bbb80b6b5e",
    "name": "assembly-38",
    "referenceStoreId": "1234567890",
    "status": "ACTIVE",
    "updateTime": "2022-11-22T22:27:09.033Z"
  }
]
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReferences](#)」の「」を参照してください。

list-run-groups

次のコード例は、list-run-groups を使用する方法を示しています。

AWS CLI

実行グループのリストを取得するには

次のlist-run-groups例では、実行グループのリストを取得します。

```
aws omics list-run-groups
```

出力:

```
{
  "items": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
      "creationTime": "2022-12-01T00:58:42.915219Z",
      "id": "1234567",
      "maxCpus": 20,
      "maxDuration": 600,
      "name": "cram-convert"
    }
  ]
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRunGroups](#)」の「」を参照してください。

list-run-tasks

次の例は、list-run-tasks を使用する方法を説明しています。

AWS CLI

タスクのリストを取得するには

次のlist-run-tasks例では、ワークフロー実行のタスクのリストを取得します。

```
aws omics list-run-tasks \  
  --id 1234567
```

出力:

```
{  
  "items": [  
    {  
      "cpus": 1,  
      "creationTime": "2022-11-30T23:13:00.718651Z",  
      "memory": 15,  
      "name": "CramToBamTask",  
      "startTime": "2022-11-30T23:17:47.016Z",  
      "status": "COMPLETED",  
      "stopTime": "2022-11-30T23:18:21.503Z",  
      "taskId": "1234567"  
    },  
    {  
      "cpus": 1,  
      "creationTime": "2022-11-30T23:18:32.315606Z",  
      "memory": 4,  
      "name": "ValidateSamFile",  
      "startTime": "2022-11-30T23:23:40.165Z",  
      "status": "COMPLETED",  
      "stopTime": "2022-11-30T23:24:14.766Z",  
      "taskId": "1234567"  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRunTasks](#)」の「」を参照してください。

list-runs

次の例は、list-runs を使用する方法を説明しています。

AWS CLI

ワークフロー実行のリストを取得するには

次のlist-runs例では、ワークフロー実行のリストを取得します。

```
aws omics list-runs
```

出力:

```
{  
  "items": [  
    {  
      "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",  
      "creationTime": "2022-12-02T23:20:01.202074Z",  
      "id": "1234567",  
      "name": "cram-to-bam",  
      "priority": 1,  
      "startTime": "2022-12-02T23:29:18.115Z",  
      "status": "COMPLETED",  
      "stopTime": "2022-12-02T23:57:54.428812Z",  
      "storageCapacity": 10,  
      "workflowId": "1234567"  
    },  
    {  
      "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",  
      "creationTime": "2022-12-03T00:16:57.180066Z",  
      "id": "1234567",  
      "name": "cram-to-bam",  
      "priority": 1,  
      "startTime": "2022-12-03T00:16:57.180066Z",  
      "status": "COMPLETED",  
      "stopTime": "2022-12-03T00:16:57.180066Z",  
      "storageCapacity": 10,  
      "workflowId": "1234567"  
    }  
  ]  
}
```

```
    "name": "cram-to-bam",
    "priority": 1,
    "startTime": "2022-12-03T00:26:50.233Z",
    "status": "FAILED",
    "stopTime": "2022-12-03T00:37:21.451340Z",
    "storageCapacity": 10,
    "workflowId": "1234567"
  },
  {
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "creationTime": "2022-12-05T17:57:08.444817Z",
    "id": "1234567",
    "name": "cram-to-bam",
    "status": "STARTING",
    "workflowId": "1234567"
  }
]
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRuns](#)」の「」を参照してください。

list-sequence-stores

次のコード例は、list-sequence-stores を使用する方法を示しています。

AWS CLI

シーケンスストアのリストを取得するには

次のlist-sequence-stores例では、シーケンスストアのリストを取得します。

```
aws omics list-sequence-stores
```

出力:

```
{
  "sequenceStores": [
    {
```

```
    "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
    "creationTime": "2022-11-23T01:24:33.629Z",
    "id": "1234567890",
    "name": "my-seq-store"
  }
]
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSequenceStores](#)」の「」を参照してください。

list-shares

次のコード例は、list-shares を使用する方法を示しています。

AWS CLI

HealthOmics 分析データの利用可能な共有を一覧表示するには

次のlist-shares例では、リソース所有者用に作成されたすべての共有を一覧表示します。

```
aws omics list-shares \
  --resource-owner SELF
```

出力:

```
{
  "shares": [
    {
      "shareId": "595c1cbd-a008-4eca-a887-954d30c91c6e",
      "name": "myShare",
      "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_1",
      "principalSubscriber": "123456789012",
      "ownerId": "555555555555",
      "status": "PENDING"
    }
  ]
}
```

```

    "name": "myShare3456",
    "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_2",
    "principalSubscriber": "123456789012",
    "ownerId": "555555555555",
    "status": "ACTIVE"
  },
  {
    "shareId": "203152f5-eef9-459d-a4e0-a691668d44ef",
    "name": "myShare4",
    "resourceArn": "arn:aws:omics:us-west-2:555555555555:variantStore/
store_3",
    "principalSubscriber": "123456789012",
    "ownerId": "555555555555",
    "status": "ACTIVE"
  }
]
}

```

詳細については、「ユーザーガイド」の[「クロスアカウント共有」](#)を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[ListShares](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

タグのリストを取得するには

次のlist-tags-for-resource例では、ID が のワークフローのタグのリストを取得します1234567。

```

aws omics list-tags-for-resource \
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567

```

出力:

```
{
```

```
"tags": {
  "department": "analytics"
}
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Amazon Omics でのリソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-variant-import-jobs

次の例は、list-variant-import-jobs を使用する方法を説明しています。

AWS CLI

バリエーションインポートジョブのリストを取得するには

次のlist-variant-import-jobs例では、バリエーションインポートジョブのリストを取得します。

```
aws omics list-variant-import-jobs
```

出力:

```
{
  "variantImportJobs": [
    {
      "creationTime": "2022-11-23T22:47:02.514002Z",
      "destinationName": "my_var_store",
      "id": "69cb65d6-xmpl-4a4a-9025-4565794b684e",
      "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W801XMPL7QZ",
      "runLeftNormalization": false,
      "status": "COMPLETED",
      "updateTime": "2022-11-23T22:49:17.976597Z"
    },
    {
      "creationTime": "2022-11-23T22:42:50.037812Z",
      "destinationName": "my_var_store",
      "id": "edd7b8ce-xmpl-47e2-bc99-258cac95a508",
```



```
        "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-  
serviceRole-W801XMPL7QZ",  
        "runLeftNormalization": false,  
        "status": "COMPLETED",  
        "updateTime": "2022-11-23T22:45:26.009880Z"  
    }  
]  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVariantImportJobs](#)」の「」を参照してください。

list-variant-stores

次の例は、list-variant-stores を使用する方法を説明しています。

AWS CLI

バリエントストアのリストを取得するには

次のlist-variant-stores例では、バリエントストアのリストを取得します。

```
aws omics list-variant-stores
```

出力:

```
{  
  "variantStores": [  
    {  
      "creationTime": "2022-11-23T22:09:07.534499Z",  
      "id": "02dexmplcfdd",  
      "name": "my_var_store",  
      "reference": {  
        "referenceArn": "arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890"  
      },  
      "status": "CREATING",  
      "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/  
my_var_store",  
    }  
  ]  
}
```

```
        "storeSizeBytes": 0,
        "updateTime": "2022-11-23T22:09:24.931711Z"
    },
    {
        "creationTime": "2022-09-23T23:00:09.140265Z",
        "id": "8777xmpl1a24",
        "name": "myvstore0",
        "status": "ACTIVE",
        "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
myvstore0",
        "storeSizeBytes": 0,
        "updateTime": "2022-09-23T23:03:26.013220Z"
    }
]
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVariantStores](#)」の「」を参照してください。

list-workflows

次の例は、list-workflows を使用する方法を説明しています。

AWS CLI

ワークフローのリストを取得するには

次のlist-workflows例では、ワークフローのリストを取得します。

```
aws omics list-workflows
```

出力:

```
{
  "items": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
      "creationTime": "2022-09-23T23:08:22.041227Z",
      "digest": "nSCNo/qMWFxmplXpUdokXJnwgne0axyyc2Y0xVxrJTE=",
      "id": "1234567",
```

```
    "name": "my-wkflow-0",
    "status": "ACTIVE",
    "type": "PRIVATE"
  },
  {
    "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
    "creationTime": "2022-11-30T22:33:16.225368Z",
    "digest":
"sha256:c54bxmpl1742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",
    "id": "1234567",
    "name": "cram-converter",
    "status": "ACTIVE",
    "type": "PRIVATE"
  }
]
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListWorkflows](#)」の「」を参照してください。

start-annotation-import-job

次の例は、start-annotation-import-job を使用する方法を説明しています。

AWS CLI

注釈をインポートするには

次のstart-annotation-import-job例では、Amazon S3 から注釈をインポートします。

```
aws omics start-annotation-import-job \
  --destination-name tsv_ann_store \
  --no-run-left-normalization \
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
  --items source=s3://omics-artifacts-01d6xmpl4e72dd32/targetedregions.bed.gz
```

出力:

```
{
```

```
"jobId": "984162c7-xmpl-4d23-ab47-286f7950bfbf"
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartAnnotationImportJob](#)」の「」を参照してください。

start-read-set-activation-job

次の例は、start-read-set-activation-job を使用する方法を説明しています。

AWS CLI

アーカイブされたリードセットをアクティブ化するには

次のstart-read-set-activation-job例では、2つのリードセットをアクティブ化します。

```
aws omics start-read-set-activation-job \
  --sequence-store-id 1234567890 \
  --sources readSetId=1234567890 readSetId=1234567890
```

出力:

```
{
  "creationTime": "2022-12-06T22:35:10.100Z",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "SUBMITTED"
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartReadSetActivationJob](#)」の「」を参照してください。

start-read-set-export-job

次のコード例は、start-read-set-export-job を使用する方法を示しています。

AWS CLI

リードセットをエクスポートするには

次のstart-read-set-export-job例では、2つのリードセットをAmazon S3にエクスポートします。

```
aws omics start-read-set-export-job \  
  --sequence-store-id 1234567890 \  
  --sources readSetId=1234567890 readSetId=1234567890 \  
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ  
\  
  --destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/
```

出力:

```
{  
  "creationTime": "2022-12-06T22:37:18.612Z",  
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",  
  "id": "1234567890",  
  "sequenceStoreId": "1234567890",  
  "status": "SUBMITTED"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartReadSetExportJob](#)」の「」を参照してください。

start-read-set-import-job

次のコード例は、start-read-set-import-job を使用する方法を示しています。

AWS CLI

リードセットをインポートするには

次のstart-read-set-import-job例では、リードセットをインポートします。

```
aws omics start-read-set-import-job \  
  --sequence-store-id 1234567890 \  
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ
```

```
--role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ \  
--sources file://readset-sources.json
```

readset-sources.json は、次のコンテンツを含むJSONドキュメントです。

```
[  
  {  
    "sourceFiles":  
    {  
      "source1": "s3://omics-artifacts-01d6xmpl4e72dd32/  
HG00100.chrom20.ILLUMINA.bwa.GBR.low_coverage.20101123.bam"  
    },  
    "sourceFileType": "BAM",  
    "subjectId": "bam-subject",  
    "sampleId": "bam-sample",  
    "referenceArn": "arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890",  
    "name": "HG00100"  
  }  
]
```

出力:

```
{  
  "creationTime": "2022-11-23T01:36:38.158Z",  
  "id": "1234567890",  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "sequenceStoreId": "1234567890",  
  "status": "SUBMITTED"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartReadSetImportJob](#)」の「」を参照してください。

start-reference-import-job

次の例は、start-reference-import-job を使用する方法を説明しています。

AWS CLI

リファレンスゲノムをインポートするには

次のstart-reference-import-job例では、Amazon S3 からリファレンスゲノムをインポートします。

```
aws omics start-reference-import-job \  
  --reference-store-id 1234567890 \  
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ \  
  --sources sourceFile=s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta,name=assembly-38
```

出力:

```
{  
  "creationTime": "2022-11-22T22:25:41.124Z",  
  "id": "1234567890",  
  "referenceStoreId": "1234567890",  
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
W801XMPL7QZ",  
  "status": "SUBMITTED"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartReferenceImportJob](#)」の「」を参照してください。

start-run

次の例は、start-run を使用する方法を説明しています。

AWS CLI

ワークフローを実行するには

次のstart-run例では、ID でワークフローを実行します1234567。

```
aws omics start-run \  
  --workflow-id 1234567
```

```
--workflow-id 1234567 \  
--role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-  
WS01XMPL7QZ \  
--name 'cram-to-bam' \  
--output-uri s3://omics-artifacts-01d6xmpl4e72dd32/workflow-output/ \  
--run-group-id 1234567 \  
--priority 1 \  
--storage-capacity 10 \  
--log-level ALL \  
--parameters file://workflow-inputs.json
```

workflow-inputs.json は、次の内容のJSONドキュメントです。

```
{  
  "sample_name": "NA12878",  
  "input_cram": "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram",  
  "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.dict",  
  "ref_fasta": "s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta",  
  "ref_fasta_index": "omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.fasta.fai"  
}
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",  
  "id": "1234567",  
  "status": "PENDING",  
  "tags": {}  
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

Amazon Omics からソースファイルをロードするには

サービス固有の `awscli` を使用して、Amazon Omics ストレージからソースファイルをロードすることもできます。次の Workflow-inputs.json ファイルの例では、リードセットおよびリファレンスゲノムソースURLsに Amazon Omics を使用します。


```
{
  "sample_name": "NA12878",
  "input_cram": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/
readSet/1234567890/source1",
  "ref_dict": "s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.dict",
  "ref_fasta": "omics://123456789012.storage.us-west-2.amazonaws.com/1234567890/
reference/1234567890",
  "ref_fasta_index": "omics://123456789012.storage.us-
west-2.amazonaws.com/1234567890/reference/1234567890/index"
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartRun](#)」の「」を参照してください。

start-variant-import-job

次の例は、start-variant-import-job を使用する方法を説明しています。

AWS CLI

バリエーションファイルをインポートするには

次のstart-variant-import-job例では、VCFフォーマットバリエーションファイルをインポートします。

```
aws omics start-variant-import-job \
  --destination-name my_var_store \
  --no-run-left-normalization \
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
  --items source=s3://omics-artifacts-01d6xmpl4e72dd32/
Homo_sapiens_assembly38.known_indels.vcf.gz
```

出力:

```
{
  "jobId": "edd7b8ce-xmpl-47e2-bc99-258cac95a508"
```

```
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartVariantImportJob](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、ID が のワークフローにdepartmentタグを追加します1234567。

```
aws omics tag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tags department=analytics
```

詳細については、「[Amazon Omics デベロッパーガイド](#)」の「[Amazon Omics でのリソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースからタグを削除するには

次のuntag-resource例では、ワークフローからdepartmentタグを削除します。

```
aws omics untag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tag-keys department
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-annotation-store

次の例は、update-annotation-store を使用する方法を説明しています。

AWS CLI

注釈ストアを更新するには

次のupdate-annotation-store例では、 という名前の注釈ストアの説明を更新しますmy_vcf_store。

```
aws omics update-annotation-store \  
  --name my_vcf_store \  
  --description "VCF annotation store"
```

出力:

```
{  
  "creationTime": "2022-12-05T18:00:56.101860Z",  
  "description": "VCF annotation store",  
  "id": "bd6axmpl2444",  
  "name": "my_vcf_store",  
  "reference": {  
    "referenceArn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890/reference/1234567890"  
  },  
  "status": "ACTIVE",  
  "storeFormat": "VCF",  
  "updateTime": "2022-12-05T18:13:16.100051Z"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAnnotationStore](#)」の「」を参照してください。

update-run-group

次の例は、update-run-group を使用する方法を説明しています。

AWS CLI

実行グループを更新するには

次のupdate-run-group例では、ID を使用して実行グループの設定を更新します1234567。

```
aws omics update-run-group \  
  --id 1234567 \  
  --max-cpus 10
```

出力:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",  
  "creationTime": "2022-12-01T00:58:42.915219Z",  
  "id": "1234567",  
  "maxCpus": 10,  
  "maxDuration": 600,  
  "name": "cram-convert",  
  "tags": {}  
}
```

詳細については、[「Amazon Omics デベロッパーガイド」の「Omics ワークフロー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRunGroup](#)」の「」を参照してください。

update-variant-store

次のコード例は、update-variant-store を使用する方法を示しています。

AWS CLI

バリエーションストアを更新するには

次のupdate-variant-store例では、 という名前のバリエーションストアの説明を更新しますmy_var_store。

```
aws omics update-variant-store \  
  --name my_var_store \  
  --description "variant store"
```

出力:

```
{  
  "creationTime": "2022-11-23T22:09:07.534499Z",  
  "description": "variant store",  
  "id": "02dexplcfd",  
  "name": "my_var_store",  
  "reference": {  
    "referenceArn": "arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890"  
  },  
  "status": "ACTIVE",  
  "updateTime": "2022-12-05T18:23:37.686402Z"  
}
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Analytics」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVariantStore](#)」の「」を参照してください。

update-workflow

次のコード例は、update-workflow を使用する方法を示しています。

AWS CLI

ワークフローを更新するには

次のupdate-workflow例では、ID でワークフローの説明を更新します1234567。

```
aws omics update-workflow \  
  --id 1234567 \  
  --description "copy workflow"
```

詳細については、「Amazon [Omics デベロッパーガイド](#)」の「Omics Storage」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateWorkflow](#)」の「」を参照してください。

upload-read-set-part

次の例は、upload-read-set-part を使用する方法を説明しています。

AWS CLI

リードセットパートをアップロードするには。

次のupload-read-set-part例では、リードセットの指定された部分をアップロードします。

```
aws omics upload-read-set-part \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --part-source SOURCE1 \  
  --part-number 1 \  
  --payload /path/to/file/read_1_part_1.fastq.gz
```

出力:

```
{  
  "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"  
}
```

詳細については、「ユーザーガイド」の「[シーケンスストアへの直接アップロード](#)」を参照してください。AWS HealthOmics

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadReadSetPart](#)」の「」を参照してください。

IAM を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますIAM。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-client-id-to-open-id-connect-provider

次の例は、`add-client-id-to-open-id-connect-provider` を使用する方法を説明しています。

AWS CLI

Open-ID Connect () プロバイダーにクライアント ID (オーディエンスOIDC) を追加するには

次の`add-client-id-to-open-id-connect-provider`コマンドは、 という名前のOIDCプロバイダー`my-application-ID`にクライアント ID を追加します`server.example.com`。

```
aws iam add-client-id-to-open-id-connect-provider \  
  --client-id my-application-ID \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com
```

このコマンドでは何も出力されません。

OIDC プロバイダーを作成するには、`create-open-id-connect-provider` コマンドを使用します。

詳細については、AWS IAM 「ユーザーガイド」の[OpenID Connect \(OIDC\) ID プロバイダーの作成](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddClientIdToOpenIdConnectProvider](#)」の「」を参照してください。

add-role-to-instance-profile

次の例は、`add-role-to-instance-profile` を使用する方法を説明しています。

AWS CLI

ロールをインスタンスプロファイルに追加するには

次の `add-role-to-instance-profile` コマンドは、`S3Access` という名前のロールを `Webserver` という名前のインスタンスプロファイルに追加します。

```
aws iam add-role-to-instance-profile \  
  --role-name S3Access \  
  --instance-profile-name Webserver
```

このコマンドでは何も出力されません。

インスタンスプロファイルを作成するには、`create-instance-profile` コマンドを使用します。

詳細については、AWS IAM 「ユーザーガイド」のIAM「[ロールを使用して Amazon EC2 インスタンスで実行されているアプリケーションにアクセス許可を付与する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddRoleToInstanceProfile](#)」の「」を参照してください。

add-user-to-group

次のコード例は、`add-user-to-group` を使用する方法を示しています。

AWS CLI

ユーザーをIAMグループに追加するには

次の`add-user-to-group`コマンドは、`Bob` という名前のIAMユーザーを `Admins` という名前のIAMグループに追加します。

```
aws iam add-user-to-group \  
  --user-name Bob \  
  --group-name Admins
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[ユーザーグループ内のIAMユーザーの追加と削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddUserToGroup](#)」の「」を参照してください。

attach-group-policy

次の例は、attach-group-policy を使用する方法を説明しています。

AWS CLI

マネージドポリシーをIAMグループにアタッチするには

次のattach-group-policyコマンドは、 という名前の AWS マネージドポリシーReadOnlyAccessを という名前のIAMグループにアタッチしますFinance。

```
aws iam attach-group-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --group-name Finance
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[マネージドポリシーとインラインポリシー](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachGroupPolicy](#)」の「」を参照してください。

attach-role-policy

次の例は、attach-role-policy を使用する方法を説明しています。

AWS CLI

マネージドポリシーをIAMロールにアタッチするには

次のattach-role-policyコマンドは、 という名前の AWS マネージドポリシーを という名前のIAMロールReadOnlyAccessにアタッチしますReadOnlyRole。

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --role-name ReadOnlyRole
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[マネージドポリシーとインラインポリシー](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachRolePolicy](#)」の「」を参照してください。

attach-user-policy

次の例は、attach-user-policy を使用する方法を説明しています。

AWS CLI

マネージドポリシーをIAMユーザーにアタッチするには

次のattach-user-policyコマンドは、 という名前の AWS マネージドポリシーAdministratorAccessを という名前のIAMユーザーにアタッチしますAlice。

```
aws iam attach-user-policy \  
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \  
  --user-name Alice
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[マネージドポリシーとインラインポリシー](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachUserPolicy](#)」の「」を参照してください。

change-password

次の例は、change-password を使用する方法を説明しています。

AWS CLI

IAMユーザーのパスワードを変更するには

IAM ユーザーのパスワードを変更するには、 --cli-input-jsonパラメータを使用して、古いパスワードと新しいパスワードを含むJSONファイルを渡すことをお勧めします。この方法を使

用すると、英数字以外の文字を含む強力なパスワードを使用できます。英数字以外の文字を含むパスワードをコマンドラインパラメータとして渡す場合、そのパスワードの使用が難しい場合があります。--cli-input-json パラメータを使用するには、まず次の例のように、change-password コマンドと --generate-cli-skeleton パラメータを組み合わせて使用します。

```
aws iam change-password \  
  --generate-cli-skeleton > change-password.json
```

前のコマンドは change-password.json というJSONファイルを作成し、古いパスワードと新しいパスワードを入力するために使用します。例えば、ファイルは次のようになります。

```
{  
  "OldPassword": "3s0K_;xh4~8XXI",  
  "NewPassword": "]35d/{pB9Fo9wJ"  
}
```

次に、パスワードを変更するには、change-password コマンドを再度使用します。今回は --cli-input-json パラメータを渡してJSONファイルを指定します。次のchange-password コマンドは、change-password.json というJSONファイルを持つ --cli-input-json パラメータを使用します。

```
aws iam change-password \  
  --cli-input-json file://change-password.json
```

このコマンドでは何も出力されません。

このコマンドはIAMユーザーのみが呼び出すことができます。このコマンドが AWS アカウント (ルート) 認証情報を使用して呼び出された場合、コマンドはInvalidUserTypeエラーを返します。

詳細については、AWS IAM 「[ユーザーガイド](#)」のIAM 「[ユーザーが自分のパスワードを変更する方法](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスChangePassword](#)」の「」を参照してください。

create-access-key

次のコード例は、create-access-key を使用する方法を示しています。

AWS CLI

IAMユーザーのアクセスキーを作成するには

次のcreate-access-keyコマンドは、という名前のIAMユーザーのアクセスキー (アクセスキー ID とシークレットアクセスキー) を作成しますBob。

```
aws iam create-access-key \  
  --user-name Bob
```

出力:

```
{  
  "AccessKey": {  
    "UserName": "Bob",  
    "Status": "Active",  
    "CreateDate": "2015-03-09T18:39:23.411Z",  
    "SecretAccessKey": "wJa1rXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"  
  }  
}
```

シークレットアクセスキーを安全な場所に保管します。紛失した場合は回復できないため、新しいアクセスキーを作成する必要があります。

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「ユーザーのアクセスキーの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAccessKey](#)」の「」を参照してください。

create-account-alias

次のコード例は、create-account-alias を使用する方法を示しています。

AWS CLI

アカウントエイリアスを作成するには

次のcreate-account-aliasコマンドは、examplecorp AWS アカウントのエイリアスを作成します。

```
aws iam create-account-alias \  
  --account-alias examplecorp
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「アカウント ID とそのエイリアス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAccountAlias](#)」の「」を参照してください。

create-group

次の例は、create-group を使用する方法を説明しています。

AWS CLI

IAMグループを作成するには

次のcreate-groupコマンドは、 という名前のIAMグループを作成しますAdmins。

```
aws iam create-group \  
  --group-name Admins
```

出力:

```
{  
  "Group": {  
    "Path": "/",  
    "CreateDate": "2015-03-09T20:30:24.940Z",  
    "GroupId": "AIDGPMS9R04H3FEXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:group/Admins",  
    "GroupName": "Admins"  
  }  
}
```

詳細については、「ユーザーガイド」の[IAM 「ユーザーグループの作成」](#)を参照してください。

AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateGroup](#)」の「」を参照してください。

create-instance-profile

次のコード例は、create-instance-profile を使用する方法を示しています。

AWS CLI

インスタンスプロファイルを作成するには

次の create-instance-profile コマンドは、Webserver という名前のインスタンスプロファイルを作成します。

```
aws iam create-instance-profile \  
  --instance-profile-name Webserver
```

出力:

```
{  
  "InstanceProfile": {  
    "InstanceProfileId": "AIPAJMBC7DLSPEXAMPLE",  
    "Roles": [],  
    "CreateDate": "2015-03-09T20:33:19.626Z",  
    "InstanceProfileName": "Webserver",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"  
  }  
}
```

インスタンスプロファイルにロールを追加するには、add-role-to-instance-profile コマンドを使用します。

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[ロールを使用して Amazon EC2 インスタンスで実行されているアプリケーションにアクセス許可を付与する](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス CreateInstanceProfile](#)」の「」を参照してください。

create-login-profile

次の例は、create-login-profile を使用する方法を説明しています。

AWS CLI

IAMユーザーのパスワードを作成するには

IAM ユーザーのパスワードを作成するには、`--cli-input-json`パラメータを使用してパスワードを含むJSONファイルを渡すことをお勧めします。この方法を使用すると、英数字以外の文字を含む強力なパスワードを作成できます。英数字以外の文字を含むパスワードをコマンドラインパラメータとして渡す場合、そのパスワードの作成が難しい場合があります。

`--cli-input-json` パラメータを使用するには、まず次の例のように、`create-login-profile` コマンドと `--generate-cli-skeleton` パラメータを組み合わせて使用します。

```
aws iam create-login-profile \  
  --generate-cli-skeleton > create-login-profile.json
```

前のコマンドでは、`create-login-profile.json` という名前のJSONファイルを作成し、これを使用して後続の`create-login-profile`コマンドの情報を入力できます。例:

```
{  
  "UserName": "Bob",  
  "Password": "&1-3a6u:RA0djs",  
  "PasswordResetRequired": true  
}
```

次に、IAMユーザーのパスワードを作成するには、`create-login-profile` コマンドを再度使用します。今回は `--cli-input-json`パラメータを渡してJSONファイルを指定します。次の`create-login-profile`コマンドは、`create-login-profile.json` というJSONファイルを持つ `--cli-input-json`パラメータを使用します。

```
aws iam create-login-profile \  
  --cli-input-json file://create-login-profile.json
```

出力:

```
{  
  "LoginProfile": {  
    "UserName": "Bob",  
    "CreateDate": "2015-03-10T20:55:40.274Z",  
    "PasswordResetRequired": true  
  }  
}
```

新しいパスワードがアカウントのパスワードポリシーに違反している場合、コマンドは `PasswordPolicyViolation` エラーを返します。

既にパスワードを持っているユーザーのパスワードを変更するには、`update-login-profile` を使用します。アカウントのパスワードポリシーを設定するには、`update-account-password-policy` コマンドを使用します。

アカウントパスワードポリシーで許可されている場合、IAMユーザーは `change-password` コマンドを使用して独自のパスワードを変更できます。

詳細については、「[ユーザーガイド](#)」のIAM「[ユーザーのパスワードの管理](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンスCreateLoginProfile](#)」の「」を参照してください。

create-open-id-connect-provider

次のコード例は、`create-open-id-connect-provider` を使用する方法を示しています。

AWS CLI

OpenID Connect (OIDC) プロバイダーを作成するには

OpenID Connect (OIDC) プロバイダーを作成するには、`--cli-input-json` パラメータを使用して、必要なパラメータを含むJSONファイルを渡すことをお勧めします。OIDC プロバイダーを作成するときは、プロバイダーURLの を渡す必要があります、 はで始まるURL必要があります `https://`。コロン (:) とフォワードスラッシュ (/) 文字は一部のコマンドライン環境で特別な意味を持つため、 をコマンドラインパラメータURLとして渡すのは難しい場合があります。この `--cli-input-json` パラメータを使用すると、この制限を回避できます。

`--cli-input-json` パラメータを使用するには、まず次の例のように、`create-open-id-connect-provider` コマンドと `--generate-cli-skeleton` パラメータを組み合わせて使用します。

```
aws iam create-open-id-connect-provider \  
  --generate-cli-skeleton > create-open-id-connect-provider.json
```

前のコマンドでは、`create-open-id-connect-provider.json` という名前のJSONファイルを作成します。このファイルを使用して、後続の`create-open-id-connect-provider`コマンドの情報を入力できます。例:

```
{
```



```
"Url": "https://server.example.com",
"ClientIDList": [
  "example-application-ID"
],
"ThumbprintList": [
  "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
]
}
```

次に、OpenID Connect (OIDC) プロバイダーを作成するには、`create-open-id-connect-provider` コマンドを再度使用します。今回は `--cli-input-json` パラメータを渡して JSON ファイルを指定します。次の `create-open-id-connect-provider` コマンドは、`--cli-input-json` パラメータを `create-open-id-connect-provider.json` という JSON ファイルとともに使用します。

```
aws iam create-open-id-connect-provider \
  --cli-input-json file://create-open-id-connect-provider.json
```

出力:

```
{
  "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

OIDC プロバイダーの詳細については、AWS IAM 「ユーザーガイド」の [OpenID Connect \(OIDC\) ID プロバイダーの作成](#) を参照してください。

OIDC プロバイダーのサムプリントの取得の詳細については、AWS IAM ユーザーガイドの [OpenID Connect Identity Provider のサムプリントの取得](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateOpenIdConnectProvider](#)」の「」を参照してください。

create-policy-version

次の例は、`create-policy-version` を使用する方法を説明しています。

AWS CLI

新しいバージョンの管理ポリシーを作成するには

この例では、ARN である IAM ポリシーの新しい v2 バージョンを作成し `arn:aws:iam::123456789012:policy/MyPolicy`、それをデフォルトバージョンにします。

```
aws iam create-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --policy-document file://NewPolicyVersion.json \  
  --set-as-default
```

出力:

```
{  
  "PolicyVersion": {  
    "CreateDate": "2015-06-16T18:56:03.721Z",  
    "VersionId": "v2",  
    "IsDefaultVersion": true  
  }  
}
```

詳細については、AWS IAM 「[ユーザーガイド](#)」の「[バージョンニング IAM ポリシー](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス CreatePolicyVersion](#)」の「」を参照してください。

create-policy

次のコード例は、`create-policy` を使用する方法を示しています。

AWS CLI

例 1: カスタマー管理ポリシーを作成するには

次のコマンドは、`my-policy` という名前でカスタマー管理ポリシーを作成します。

```
aws iam create-policy \  
  --policy-name my-policy \  
  --policy-document file://policy
```

ファイルは、という名前の Amazon S3 バケット内のフォルダへの読み取り専用アクセスを許可する現在の `shared` フォルダ内の JSON ドキュメント `policy` です `my-bucket`。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket/shared/*"
      ]
    }
  ]
}
```

出力:

```
{
  "Policy": {
    "PolicyName": "my-policy",
    "CreateDate": "2015-06-01T19:31:18.620Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::0123456789012:policy/my-policy",
    "UpdateDate": "2015-06-01T19:31:18.620Z"
  }
}
```

文字列パラメータの入力としてファイルを使用する方法の詳細については、AWS CLI「ユーザーガイド」の「[のパラメータ値を指定する AWS CLI](#)」を参照してください。

例 2: 説明を含むカスタマー管理ポリシーを作成するには

次のコマンドは、イミュータブルな説明を使用して my-policy という名前のカスタマー管理ポリシーを作成します。

```
aws iam create-policy \
```

```
--policy-name my-policy \  
--policy-document file://policy.json \  
--description "This policy grants access to all Put, Get, and List actions for  
my-bucket"
```

ファイルは、という名前の Amazon S3 バケットのすべての Put、List、Get アクションへのアクセスを許可する現在のフォルダ内の JSON ドキュメント `policy.json` です `my-bucket`。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:ListBucket*",  
        "s3:PutBucket*",  
        "s3:GetBucket*"  
      ],  
      "Resource": [  
        "arn:aws:s3:::my-bucket"  
      ]  
    }  
  ]  
}
```

出力:

```
{  
  "Policy": {  
    "PolicyName": "my-policy",  
    "PolicyId": "ANPAWGSUGIDPEXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:policy/my-policy",  
    "Path": "/",  
    "DefaultVersionId": "v1",  
    "AttachmentCount": 0,  
    "PermissionsBoundaryUsageCount": 0,  
    "IsAttachable": true,  
    "CreateDate": "2023-05-24T22:38:47+00:00",  
    "UpdateDate": "2023-05-24T22:38:47+00:00"  
  }  
}
```

Identity ベースのポリシーの詳細については、AWS IAM 「ユーザーガイド」の [「アイデンティティベースのポリシーとリソースベースのポリシー」](#) を参照してください。

例 3: タグを使用してカスタマー管理ポリシーを作成するには

次のコマンドは、タグを使用して my-policy という名前のカスタマー管理ポリシーを作成します。この例では、--tags パラメータフラグを次の JSON 形式のタグで使用します。 '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}' あるいは、--tags フラグを次の短縮形式のタグとともに使用することもできます: 'Key=Department,Value=Accounting Key=Location,Value=Seattle'。

```
aws iam create-policy \  
  --policy-name my-policy \  
  --policy-document file://policy.json \  
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",  
  "Value": "Seattle"}'
```

ファイルは、 という名前の Amazon S3 バケットのすべての Put、List、Get アクションへのアクセスを許可する現在のフォルダ内の JSON ドキュメント policy.json です my-bucket。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:ListBucket*",  
        "s3:PutBucket*",  
        "s3:GetBucket*"  
      ],  
      "Resource": [  
        "arn:aws:s3:::my-bucket"  
      ]  
    }  
  ]  
}
```

出力:

```
{  
  "Policy": {
```

```
"PolicyName": "my-policy",
"PolicyId": "ANPAWGSUGIDPEXAMPLE",
"Arn": "arn:aws:iam::12345678012:policy/my-policy",
"Path": "/",
"DefaultVersionId": "v1",
"AttachmentCount": 0,
"PermissionsBoundaryUsageCount": 0,
"IsAttachable": true,
"CreateDate": "2023-05-24T23:16:39+00:00",
"UpdateDate": "2023-05-24T23:16:39+00:00",
"Tags": [
  {
    "Key": "Department",
    "Value": "Accounting"
  },
  {
    "Key": "Location",
    "Value": "Seattle"
  }
]
}
```

タグ付けポリシーの詳細については、AWS IAM「ユーザーガイド」の[「カスタマーマネージドポリシーのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreatePolicy](#)」の「」を参照してください。

create-role

次のコード例は、create-role を使用する方法を示しています。

AWS CLI

例 1: IAMロールを作成するには

次の create-role コマンドは、Test-Role という名前のロールを作成し、それに信頼ポリシーをアタッチします。

```
aws iam create-role \
  --role-name Test-Role \
```

```
--assume-role-policy-document file://Test-Role-Trust-Policy.json
```

出力:

```
{
  "Role": {
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2013-06-07T20:43:32.821Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/Test-Role"
  }
}
```

信頼ポリシーは、Test-Role-Trust-Policy.json ファイル内のJSONドキュメントとして定義されます。(ファイル名と拡張子には意味はありません。)信頼ポリシーはプリンシパルを指定する必要があります。

アクセス許可ポリシーをロールにアタッチするには、put-role-policy コマンドを使用します。

詳細については、「ユーザーガイド」の[IAM「ロールの作成」](#)を参照してください。AWS IAM

例 2: 指定された最大セッション時間を持つIAMロールを作成するには

次の create-role コマンドは、Test-Role という名前のロールを作成し、最大セッション時間を 7,200 秒 (2 時間) に設定します。

```
aws iam create-role \
  --role-name Test-Role \
  --assume-role-policy-document file://Test-Role-Trust-Policy.json \
  --max-session-duration 7200
```

出力:

```
{
  "Role": {
    "Path": "/",
    "RoleName": "Test-Role",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::12345678012:role/Test-Role",
  }
}
```

```

    "CreateDate": "2023-05-24T23:50:25+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "Statement1",
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::12345678012:root"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
  }
}

```

詳細については、AWS IAM「[ユーザーガイド](#)」の「[ロールの最大セッション期間の変更 \(AWS API\)](#)」を参照してください。

例 3: タグを使用してIAMロールを作成するには

次のコマンドは、タグTest-Role付きのIAMロールを作成します。この例では、`--tags`パラメータフラグをJSON形式のタグで使用します'`{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`。あるいは、`--tags`フラグを次の短縮形式のタグとともに使用することもできます:
'`Key=Department,Value=Accounting Key=Location,Value=Seattle`'。

```

aws iam create-role \
  --role-name Test-Role \
  --assume-role-policy-document file://Test-Role-Trust-Policy.json \
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",  
"Value": "Seattle"}'

```

出力:

```

{
  "Role": {
    "Path": "/",
    "RoleName": "Test-Role",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:role/Test-Role",

```



```
"CreateDate": "2023-05-25T23:29:41+00:00",
"AssumeRolePolicyDocument": {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ],
  "Tags": [
    {
      "Key": "Department",
      "Value": "Accounting"
    },
    {
      "Key": "Location",
      "Value": "Seattle"
    }
  ]
}
```

詳細については、AWS IAM「ユーザーガイド」の[IAM「ロールのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateRole](#)」の「」を参照してください。

create-saml-provider

次の例は、create-saml-provider を使用する方法を説明しています。

AWS CLI

SAMLプロバイダーを作成するには

この例では、IAMという名前の新しいSAMLプロバイダーを作成しますMySAMLProvider。ファイルにあるSAMLメタデータドキュメントで説明されていますSAMLMetaData.xml。

```
aws iam create-saml-provider \  
  --saml-metadata-document file://SAMLMetaData.xml \  
  --name MySAMLProvider
```

出力:

```
{  
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"  
}
```

詳細については、「ユーザーガイド」の [IAM SAML「アイデンティティプロバイダーの作成」](#) を参照してください。AWS IAM

- API 詳細については、「コマンドリファレンス」の [「CreateSAMLProvider」](#) を参照してください。AWS CLI

create-service-linked-role

次の例は、create-service-linked-role を使用する方法を説明しています。

AWS CLI

サービスにリンクされたロールを作成するには

次のcreate-service-linked-role例では、指定されたサービスの AWS サービスリンクロールを作成し、指定された説明をアタッチします。

```
aws iam create-service-linked-role \  
  --aws-service-name lex.amazonaws.com \  
  --description "My service-linked role to support Lex"
```

出力:

```
{  
  "Role": {  
    "Path": "/aws-service-role/lex.amazonaws.com/",  
    "RoleName": "AWSServiceRoleForLexBots",  
    "RoleId": "ARO1234567890EXAMPLE",  
    "Arn": "arn:aws:iam::1234567890:role/aws-service-role/lex.amazonaws.com/  
AWSServiceRoleForLexBots",
```

```
"CreateDate": "2019-04-17T20:34:14+00:00",
"AssumeRolePolicyDocument": {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "lex.amazonaws.com"
        ]
      }
    }
  ]
}
```

詳細については、AWS IAM「ユーザーガイド」の[「サービスにリンクされたロールの使用」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateServiceLinkedRole](#)」の「」を参照してください。

create-service-specific-credential

次の例は、create-service-specific-credential を使用する方法を説明しています。

AWS CLI

ユーザーのサービス固有の認証情報のセットを作成する

次のcreate-service-specific-credential例では、設定されたサービスにのみアクセスするために使用できるユーザー名とパスワードを作成します。

```
aws iam create-service-specific-credential \  
  --user-name sofia \  
  --service-name codecommit.amazonaws.com
```

出力:

```
{
  "ServiceSpecificCredential": {
    "CreateDate": "2019-04-18T20:45:36+00:00",
    "ServiceName": "codecommit.amazonaws.com",
    "ServiceUserName": "sofia-at-123456789012",
    "ServicePassword": "k1zPZM6uVxMQ3oxqgoY1NuJPyRTZ1vREs76zTQE3eJk=",
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
    "UserName": "sofia",
    "Status": "Active"
  }
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[HTTPSへの接続用の Git 認証情報の作成 CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateServiceSpecificCredential](#)」の「」を参照してください。

create-user

次のコード例は、create-user を使用する方法を示しています。

AWS CLI

例 1: IAM ユーザーを作成するには

次のcreate-userコマンドは、現在のアカウントに という名前BobのIAMユーザーを作成します。

```
aws iam create-user \
  --user-name Bob
```

出力:

```
{
  "User": {
    "UserName": "Bob",
    "Path": "/",
    "CreateDate": "2023-06-08T03:20:41.270Z",
    "UserId": "AIDAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/Bob"
  }
}
```

```
}
```

詳細については、[「ユーザーガイド」の AWS 「アカウントでの IAM ユーザーの作成」](#) を参照してください。AWS IAM

例 2: 指定されたパスで IAM ユーザーを作成するには

次の create-user コマンドは、指定されたパス Bob に という名前の IAM ユーザーを作成します。

```
aws iam create-user \  
  --user-name Bob \  
  --path /division_abc/subdivision_xyz/
```

出力:

```
{  
  "User": {  
    "Path": "/division_abc/subdivision_xyz/",  
    "UserName": "Bob",  
    "UserId": "AIDAIOSFODNN7EXAMPLE",  
    "Arn": "arn:aws:iam::12345678012:user/division_abc/subdivision_xyz/Bob",  
    "CreateDate": "2023-05-24T18:20:17+00:00"  
  }  
}
```

詳細については、AWS IAM [「ユーザーガイド IAM」の「識別子」](#) を参照してください。

例 3: タグを使用して IAM ユーザーを作成するには

次の create-user コマンドは、タグ Bob で という名前の IAM ユーザーを作成します。

この例では、--tags パラメータフラグを JSON 形式のタグで使用します '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'。あるいは、--tags フラグを次の短縮形式のタグとともに使用することもできます: 'Key=Department,Value=Accounting Key=Location,Value=Seattle'。

```
aws iam create-user \  
  --user-name Bob \  
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",  
  "Value": "Seattle"}'
```

出力:

```
{
  "User": {
    "Path": "/",
    "UserName": "Bob",
    "UserId": "AIDAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::12345678012:user/Bob",
    "CreateDate": "2023-05-25T17:14:21+00:00",
    "Tags": [
      {
        "Key": "Department",
        "Value": "Accounting"
      },
      {
        "Key": "Location",
        "Value": "Seattle"
      }
    ]
  }
}
```

詳細については、AWS IAM「ユーザーガイド」の[IAM「ユーザーのタグ付け」](#)を参照してください。

例 3: アクセス許可の境界が設定されたIAMユーザーを作成するには

次のcreate-userコマンドは、AmazonS3FullAccess のアクセス許可境界Bobを持つ という名前のIAMユーザーを作成します。

```
aws iam create-user \
  --user-name Bob \
  --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess
```

出力:

```
{
  "User": {
    "Path": "/",
    "UserName": "Bob",
    "UserId": "AIDAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::12345678012:user/Bob",
    "CreateDate": "2023-05-24T17:50:53+00:00",
    "PermissionsBoundary": {
```

```
    "PermissionsBoundaryType": "Policy",
    "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
  }
}
```

詳細については、AWS IAM「ユーザーガイド」のIAM「[エンティティのアクセス許可の境界](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateUser](#)」の「」を参照してください。

create-virtual-mfa-device

次のコード例は、create-virtual-mfa-device を使用する方法を示しています。

AWS CLI

仮想MFAデバイスを作成するには

この例では、という新しい仮想MFAデバイスを作成しますBobsMFADevice。QRCode.png というブートストラップ情報を含むファイルを作成し、C:/ディレクトリに配置します。この例で使用されているブートストラップメソッドは QRCodePNG です。

```
aws iam create-virtual-mfa-device \
  --virtual-mfa-device-name BobsMFADevice \
  --outfile C:/QRCode.png \
  --bootstrap-method QRCodePNG
```

出力:

```
{
  "VirtualMFADevice": {
    "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
  }
}
```

詳細については、AWS IAM「ユーザーガイド」の「[での多要素認証 \(MFA\) の使用 AWS](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateVirtualMfaDevice](#)」の「」を参照してください。

deactivate-mfa-device

次のコード例は、`deactivate-mfa-device` を使用する方法を示しています。

AWS CLI

MFAデバイスを非アクティブ化するには

このコマンドARNarn:aws:iam::210987654321:mfa/BobsMFADeviceは、ユーザーに関連付けられているを使用して仮想MFAデバイスを非アクティブ化しますBob。

```
aws iam deactivate-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の「[での多要素認証 \(MFA\) の使用 AWS](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeactivateMfaDevice](#)」の「」を参照してください。

decode-authorization-message

次の例は、`decode-authorization-message` を使用する方法を説明しています。

AWS CLI

認証失敗メッセージをデコードするには

次の`decode-authorization-message`例では、必要なアクセス許可なしでインスタンスを起動しようとする、EC2コンソールから返されるメッセージをデコードします。

```
aws sts decode-authorization-message \  
  --encoded-message LxzA8VEjEvu-s0TTt3PgYCXik9Yak0qsrFJGRZR98xNcyWAxwRq14xIvd-  
npzbgTevuufCTbjeBAaDARg9cbTK1rJbg3awM33o-Vy3ebPErE2-  
mWR9hVYdvX-0zKgV0WF9pWjZaJSMqxB-aLXo-I_8TTvBq88x8IFPbMArNdpu0IjxDjzf22PF3S0E3XvIQ-  
_PE00aUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8lsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX  
Xx9AYAAIr6bhcis7C__bZh4dLAAWooHFGKgoJcWGwgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

出力は、任意のJSONテキストプロセッサで解析できるJSONテキストの 1 行文字列としてフォーマットされます。


```
{
  "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":false,\"matchedStatements\":"
  "\":{\\"items\":[],\"failures\":{\\"items\":[],\"context\":{\\"principal\":"
  "\":{\\"id\":"AIDAV3ZUEFP6J7GY706L0\",\"name\":"chain-user\",\"arn\":"
  "\":{\\"arn:aws:iam:403299380220:user/chain-user\"},\"action\":"ec2:RunInstances\",
  "\":{\\"resource\":"arn:aws:ec2:us-east-2:403299380220:instance/*\",\"conditions\":"
  "\":{\\"items\":[{\\"key\":"ec2:InstanceMarketType\",\"values\":"{\\"items\":[{\\"value\":"
  "\":{\\"on-demand\"}]}}],{\\"key\":"aws:Resource\",\"values\":"{\\"items\":[{\\"value\":"
  "\":{\\"instance/*\"}]}}],{\\"key\":"aws:Account\",\"values\":"{\\"items\":[{\\"value\":"
  "\":{\\"403299380220\"}]}}],{\\"key\":"ec2:AvailabilityZone\",\"values\":"{\\"items\":"
  "\":{\\"value\":"us-east-2b\"}]}}],{\\"key\":"ec2:efsOptimized\",\"values\":"{\\"items\":"
  "\":{\\"value\":"false\"}]}}],{\\"key\":"ec2:IsLaunchTemplateResource\",\"values\":"
  "\":{\\"items\":[{\\"value\":"false\"}]}}],{\\"key\":"ec2:InstanceType\",\"values\":"
  "\":{\\"items\":[{\\"value\":"t2.micro\"}]}}],{\\"key\":"ec2:RootDeviceType\",
  "\":{\\"values\":"{\\"items\":[{\\"value\":"efs\"}]}}],{\\"key\":"aws:Region\",\"values\":"
  "\":{\\"items\":[{\\"value\":"us-east-2\"}]}}],{\\"key\":"aws:Service\",\"values\":"
  "\":{\\"items\":[{\\"value\":"ec2\"}]}}],{\\"key\":"ec2:InstanceID\",\"values\":"
  "\":{\\"items\":[{\\"value\":"*\"}]}}],{\\"key\":"aws:Type\",\"values\":"{\\"items\":"
  "\":{\\"value\":"instance\"}]}}],{\\"key\":"ec2:Tenancy\",\"values\":"{\\"items\":"
  "\":{\\"value\":"default\"}]}}],{\\"key\":"ec2:Region\",\"values\":"{\\"items\":"[{\\"value\":"
  "\":{\\"us-east-2\"}]}}],{\\"key\":"aws:ARN\",\"values\":"{\\"items\":"[{\\"value\":"
  "\":{\\"arn:aws:ec2:us-east-2:403299380220:instance/*\"}]}}]}\""}
}
```

詳細については、AWS re:Post [のEC2「インスタンス起動中にUnauthorizedOperation」エラーを受け取った後に認証失敗メッセージをデコードする方法](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DecodeAuthorizationMessage](#)」の「」を参照してください。

delete-access-key

次の例は、delete-access-key を使用する方法を説明しています。

AWS CLI

IAMユーザーのアクセスキーを削除するには

次のdelete-access-keyコマンドは、 という名前のIAMユーザーの指定されたアクセスキー (アクセスキー ID とシークレットアクセスキー) を削除しますBob。

```
aws iam delete-access-key \
```

```
--access-key-id AKIDPMS9R04H3FEXAMPLE \  
--user-name Bob
```

このコマンドでは何も出力されません。

IAM ユーザー用に定義されたアクセスキーを一覧表示するには、`list-access-keys` コマンドを使用します。

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「ユーザーのアクセスキーの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccessKey](#)」の「」を参照してください。

delete-account-alias

次のコード例は、`delete-account-alias` を使用する方法を示しています。

AWS CLI

アカウントエイリアスを削除するには

次の `delete-account-alias` コマンドは、現在のアカウントのエイリアス `mycompany` を削除します。

```
aws iam delete-account-alias \  
--account-alias mycompany
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「アカウント ID とそのエイリアス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccountAlias](#)」の「」を参照してください。

delete-account-password-policy

次のコード例は、`delete-account-password-policy` を使用する方法を示しています。

AWS CLI

現在のアカウントのパスワードポリシーを削除するには

次の `delete-account-password-policy` コマンドは、現在のアカウントのパスワードポリシーを削除します。

```
aws iam delete-account-password-policy
```

このコマンドでは何も出力されません。

詳細については、[「ユーザーガイド」のIAM「ユーザーのアカウントパスワードポリシーの設定」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteAccountPasswordPolicy](#)」の「」を参照してください。

delete-group-policy

次の例は、`delete-group-policy` を使用する方法を説明しています。

AWS CLI

IAMグループからポリシーを削除するには

次の `delete-group-policy` コマンドは、Admins という名前のグループから ExamplePolicy という名前のポリシーを削除します。

```
aws iam delete-group-policy \  
  --group-name Admins \  
  --policy-name ExamplePolicy
```

このコマンドでは何も出力されません。

グループにアタッチされているポリシーを表示するには、`list-group-policies` コマンドを使用します。

詳細については、「ユーザーガイド」の[IAM「ポリシーの管理」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteGroupPolicy](#)」の「」を参照してください。

delete-group

次のコード例は、`delete-group` を使用する方法を示しています。

AWS CLI

IAMグループを削除するには

次のdelete-groupコマンドは、 という名前のIAMグループを削除しますMyTestGroup。

```
aws iam delete-group \  
  --group-name MyTestGroup
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」のIAM「[ユーザーグループの削除](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteGroup](#)」の「」を参照してください。

delete-instance-profile

次の例は、delete-instance-profile を使用する方法を説明しています。

AWS CLI

インスタンスプロファイルを削除するには

次の delete-instance-profile コマンドは、ExampleInstanceProfile という名前のインスタンスプロファイルを削除します。

```
aws iam delete-instance-profile \  
  --instance-profile-name ExampleInstanceProfile
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[インスタンスプロファイルの使用](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteInstanceProfile](#)」の「」を参照してください。

delete-login-profile

次の例は、delete-login-profile を使用する方法を説明しています。

AWS CLI

IAMユーザーのパスワードを削除するには

次のdelete-login-profileコマンドは、 という名前のIAMユーザーのパスワードを削除しますBob。

```
aws iam delete-login-profile \  
  --user-name Bob
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」のIAM「[ユーザーのパスワードの管理](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンスDeleteLoginProfile](#)」の「」を参照してください。

delete-open-id-connect-provider

次のコード例は、delete-open-id-connect-provider を使用する方法を示しています。

AWS CLI

IAM OpenID Connect ID プロバイダーを削除するには

この例では、IAMOIDCプロバイダー に接続するプロバイダーを削除しますexample.oidcprovider.com。

```
aws iam delete-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
  example.oidcprovider.com
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[OpenID Connect \(OIDC\) ID プロバイダーの作成](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDeleteOpenIdConnectProvider](#)」の「」を参照してください。

delete-policy-version

次の例は、delete-policy-version を使用する方法を説明しています。

AWS CLI

管理ポリシーのバージョンを削除するには

この例では、として識別されたバージョンを、ARNであるポリシーv2から削除しますarn:aws:iam::123456789012:policy/MySamplePolicy。

```
aws iam delete-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicyVersion](#)」の「」を参照してください。

delete-policy

次のコード例は、delete-policy を使用する方法を示しています。

AWS CLI

IAMポリシーを削除するには

この例では、ARN であるポリシーを削除しますarn:aws:iam::123456789012:policy/MySamplePolicy。

```
aws iam delete-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicy](#)」の「」を参照してください。

delete-role-permissions-boundary

次の例は、delete-role-permissions-boundary を使用する方法を説明しています。

AWS CLI

IAMロールからアクセス許可の境界を削除するには

次のdelete-role-permissions-boundary例では、指定されたIAMロールのアクセス許可の境界を削除します。アクセス許可の境界をロールに適用するには、put-role-permissions-boundary コマンドを使用します。

```
aws iam delete-role-permissions-boundary \  
  --role-name lambda-application-role
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRolePermissionsBoundary](#)」の「」を参照してください。

delete-role-policy

次のコード例は、delete-role-policy を使用する方法を示しています。

AWS CLI

IAMロールからポリシーを削除するには

次の delete-role-policy コマンドは、Test-Role という名前のロールから ExamplePolicy という名前のポリシーを削除します。

```
aws iam delete-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[「ロールの変更」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteRolePolicy](#)」の「」を参照してください。

delete-role

次の例は、delete-role を使用する方法を説明しています。

AWS CLI

IAMロールを削除するには

次の delete-role コマンドは、Test-Role という名前のロールを削除します。

```
aws iam delete-role \  
  --role-name Test-Role
```

このコマンドでは何も出力されません。

ロールを削除する前に、インスタンスプロファイルからロールを削除し (remove-role-from-instance-profile)、管理ポリシーをデタッチして (detach-role-policy)、ロールにアタッチされているインラインポリシーを削除する (delete-role-policy) 必要があります。

詳細については、「ユーザーガイド」の[IAM「ロールの作成」](#)と[「インスタンスプロファイルの使用」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteRole](#)」の「」を参照してください。

delete-saml-provider

次のコード例は、delete-saml-provider を使用する方法を示しています。

AWS CLI

SAMLプロバイダーを削除するには

この例では、ARN である IAM SAML 2.0 プロバイダーを削除します。arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider。

```
aws iam delete-saml-provider \  
  --arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```



```
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の [IAM SAML「アイデンティティプロバイダーの作成」](#) を参照してください。AWS IAM

- API 詳細については、「コマンドリファレンス」の [「DeleteSAMLProvider」](#) を参照してください。AWS CLI

delete-server-certificate

次の例は、delete-server-certificate を使用する方法を説明しています。

AWS CLI

AWS アカウントからサーバー証明書を削除するには

次のdelete-server-certificateコマンドは、指定されたサーバー証明書を AWS アカウントから削除します。

```
aws iam delete-server-certificate \  
  --server-certificate-name myUpdatedServerCertificate
```

このコマンドでは何も出力されません。

AWS アカウントで使用できるサーバー証明書を一覧表示するには、list-server-certificates コマンドを使用します。

詳細については、AWS IAM「ユーザーガイド」の [「でのサーバー証明書の管理IAM」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DeleteServerCertificate](#)」の「」を参照してください。

delete-service-linked-role

次の例は、delete-service-linked-role を使用する方法を説明しています。

AWS CLI

サービスにリンクされたロールを削除するには

次の `delete-service-linked-role` の例では、不要になったサービスにリンクされたロールのうち、指定されたものを削除します。削除は非同期で実行されます。`get-service-linked-role-deletion-status` コマンドを使用して、削除のステータスをチェックし、削除がいつ完了したかを確認できます。

```
aws iam delete-service-linked-role \  
  --role-name AWSServiceRoleForLexBots
```

出力:

```
{  
  "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/  
  AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"  
}
```

詳細については、AWS IAM 「ユーザーガイド」の [「サービスにリンクされたロールの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteServiceLinkedRole](#)」の「」を参照してください。

delete-service-specific-credential

次の例は、`delete-service-specific-credential` を使用方法を説明しています。

AWS CLI

例 1: リクエストするユーザーのサービス固有の認証情報を削除する

次の `delete-service-specific-credential` 例では、リクエストを行うユーザーの指定されたサービス固有の認証情報を削除します。`service-specific-credential-id` は、認証情報を作成するときに提供され、`list-service-specific-credentials` コマンドを使用して取得できます。

```
aws iam delete-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

このコマンドでは何も出力されません。

例 2: 指定されたユーザーのサービス固有の認証情報を削除する

次のdelete-service-specific-credential例では、指定されたユーザーの指定されたサービス固有の認証情報を削除します。service-specific-credential-id は、認証情報を作成するときに提供され、list-service-specific-credentials コマンドを使用して取得できます。

```
aws iam delete-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit ユーザーガイドの「[HTTPSへの接続用の Git 認証情報の作成 CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteServiceSpecificCredential](#)」の「」を参照してください。

delete-signing-certificate

次の例は、delete-signing-certificate を使用する方法を説明しています。

AWS CLI

IAMユーザーの署名証明書を削除するには

次のdelete-signing-certificateコマンドは、という名前のIAMユーザーの指定された署名証明書を削除しますBob。

```
aws iam delete-signing-certificate \  
  --user-name Bob \  
  --certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE
```

このコマンドでは何も出力されません。

署名証明書の ID を取得するには、list-signing-certificates コマンドを使用します。

詳細については、「Amazon ユーザーガイド」の「[署名証明書の管理](#)」を参照してください。

EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSigningCertificate](#)」の「」を参照してください。

delete-ssh-public-key

次の例は、delete-ssh-public-key を使用する方法を説明しています。

AWS CLI

IAM ユーザーに添付されたSSHパブリックキーを削除するには

次のdelete-ssh-public-keyコマンドは、IAMユーザー にアタッチされた指定されたSSHパブリックキーを削除しますsofia。

```
aws iam delete-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[SSHキーと SSH を使用する CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSshPublicKey](#)」の「」を参照してください。

delete-user-permissions-boundary

次のコード例は、delete-user-permissions-boundary を使用する方法を示しています。

AWS CLI

IAM ユーザーからアクセス許可の境界を削除するには

次のdelete-user-permissions-boundary例では、という名前のIAMユーザーにアタッチされたアクセス許可の境界を削除しますintern。アクセス許可の境界をユーザーに適用するには、put-user-permissions-boundary コマンドを使用します。

```
aws iam delete-user-permissions-boundary \  
  --user-name intern
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUserPermissionsBoundary](#)」の「」を参照してください。

delete-user-policy

次の例は、delete-user-policy を使用する方法を説明しています。

AWS CLI

IAM ユーザーからポリシーを削除するには

次のdelete-user-policyコマンドは、 という名前のIAMユーザーから指定されたポリシーを削除しますBob。

```
aws iam delete-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy
```

このコマンドでは何も出力されません。

IAM ユーザーのポリシーのリストを取得するには、 list-user-policies コマンドを使用します。

詳細については、「ユーザーガイド」の[AWS 「アカウントでのIAMユーザーの作成」](#)を参照してください。 AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUserPolicy](#)」の「」を参照してください。

delete-user

次の例は、delete-user を使用する方法を説明しています。

AWS CLI

IAM ユーザーを削除するには

次のdelete-userコマンドは、現在のアカウントBobから という名前のIAMユーザーを削除します。

```
aws iam delete-user \  
  --user-name Bob
```

```
--user-name Bob
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の [IAM「ユーザーの削除」](#) を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [DeleteUser](#)」の「」を参照してください。

delete-virtual-mfa-device

次のコード例は、delete-virtual-mfa-device を使用する方法を示しています。

AWS CLI

仮想MFAデバイスを削除するには

次のdelete-virtual-mfa-deviceコマンドは、現在のアカウントから指定されたMFAデバイスを削除します。

```
aws iam delete-virtual-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の [MFA「デバイスの非アクティブ化」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DeleteVirtualMfaDevice](#)」の「」を参照してください。

detach-group-policy

次のコード例は、detach-group-policy を使用する方法を示しています。

AWS CLI

グループからポリシーをデタッチするには

この例では、というグループARNarn:aws:iam::123456789012:policy/TesterAccessPolicyから を含む マネージドポリシーを削除しますTesters。

```
aws iam detach-group-policy \  
  --group-name Testers \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の [IAM「ユーザーグループの管理」](#) を参照してください。

AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [DetachGroupPolicy](#)」の「」を参照してください。

detach-role-policy

次のコード例は、detach-role-policy を使用する方法を示しています。

AWS CLI

ロールからポリシーをデタッチするには

この例では、というロールARNarn:aws:iam::123456789012:policy/FederatedTesterAccessPolicyから を含む マネージドポリシーを削除しますFedTesterRole。

```
aws iam detach-role-policy \  
  --role-name FedTesterRole \  
  --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の [「ロールの変更」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DetachRolePolicy](#)」の「」を参照してください。

detach-user-policy

次のコード例は、detach-user-policy を使用する方法を示しています。

AWS CLI

ユーザーからポリシーをデタッチするには

この例では、ユーザー `ARNarn:aws:iam::123456789012:policy/TesterPolicy` からを含む管理ポリシーを削除しますBob。

```
aws iam detach-user-policy \  
  --user-name Bob \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

このコマンドでは何も出力されません。

詳細については、「[ユーザーガイド](#)」のIAM「[ユーザーのアクセス許可の変更](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンスDetachUserPolicy](#)」の「」を参照してください。

enable-mfa-device

次の例は、`enable-mfa-device` を使用する方法を説明しています。

AWS CLI

MFA デバイスを有効にするには

`create-virtual-mfa-device` コマンドを使用して新しい仮想MFAデバイスを作成したら、MFAそのデバイスをユーザーに割り当てることができます。次の`enable-mfa-device`例では、シリアル番号を持つMFAデバイスをユーザー `arn:aws:iam::210987654321:mfa/BobsMFADevice` に割り当てますBob。コマンドは、仮想デバイスからの最初の 2 つのコードを順番に含める AWS ことでMFA、デバイスを と同期します。

```
aws iam enable-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \  
  --authentication-code1 123456 \  
  --authentication-code2 789012
```

このコマンドでは何も出力されません。

詳細については、AWS IAM ユーザーガイドの「[仮想多要素認証 \(MFA\) デバイスを有効にする](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスEnableMfaDevice](#)」の「」を参照してください。

generate-credential-report

次のコード例は、generate-credential-report を使用する方法を示しています。

AWS CLI

認証情報レポートを生成するには

次の例では、AWS アカウントの認証情報レポートを生成しようとしています。

```
aws iam generate-credential-report
```

出力:

```
{
  "State": "STARTED",
  "Description": "No report exists. Starting a new report generation task"
}
```

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「アカウントの認証情報レポートの取得」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateCredentialReport](#)」の「」を参照してください。

generate-organizations-access-report

次のコード例は、generate-organizations-access-report を使用する方法を示しています。

AWS CLI

例 1: 組織内のルートへのアクセスレポートを生成するには

次のgenerate-organizations-access-report例では、バックグラウンドジョブを開始して、組織内の指定されたルートへのアクセスレポートを作成します。get-organizations-access-report コマンドを実行して、レポートの作成後にレポートを表示できます。

```
aws iam generate-organizations-access-report \
  --entity-path o-4fxmpl1t198/r-c3xb
```

出力:

```
{
  "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"
}
```

例 2: 組織内のアカウントのアクセスレポートを生成するには

次のgenerate-organizations-access-report例では、バックグラウンドジョブを開始して、組織 123456789012 のアカウント ID のアクセスレポートを作成します。o-4fxmplt198。get-organizations-access-report コマンドを実行して、レポートの作成後にレポートを表示できます。

```
aws iam generate-organizations-access-report \
  --entity-path o-4fxmplt198/r-c3xb/123456789012
```

出力:

```
{
  "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"
}
```

例 3: 組織内の組織単位のアカウントのアクセスレポートを生成するには

次のgenerate-organizations-access-report例では、バックグラウンドジョブを開始して、組織 ou-c3xb-lmu7j2yg の組織単位234567890123にアカウント ID のアクセスレポートを作成します。o-4fxmplt198。get-organizations-access-report コマンドを実行して、レポートの作成後にレポートを表示できます。

```
aws iam generate-organizations-access-report \
  --entity-path o-4fxmplt198/r-c3xb/ou-c3xb-lmu7j2yg/234567890123
```

出力:

```
{
  "JobId": "2eb6c2e6-0xmp-ec04-1425-c937916a64af"
}
```

組織内のルートと組織単位の詳細については、コマンドorganizations list-rootsとorganizations list-organizational-units-for-parent コマンドを使用します。

詳細については、AWS IAM「ユーザーガイド」の「[最終アクセス情報 AWS を使用した でのアクセス許可の改良](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [GenerateOrganizationsAccessReport](#)」の「」を参照してください。

generate-service-last-accessed-details

次のコード例は、generate-service-last-accessed-details を使用する方法を示しています。

AWS CLI

例 1: カスタムポリシーのサービスアクセスレポートを生成するには

次のgenerate-service-last-accessed-details例では、`intern-boundary` という名前のカスタムポリシーを使用して、IAMユーザーやその他のエンティティがアクセスするサービスを一覧表示するレポートを生成するバックグラウンドジョブを開始します。generate-service-last-accessed-details コマンドを実行すると、レポートを作成した後に表示できます。

```
aws iam generate-service-last-accessed-details \  
  --arn arn:aws:iam::123456789012:policy/intern-boundary
```

出力:

```
{  
  "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"  
}
```

例 2: AWS マネージド AdministratorAccess ポリシーのサービスアクセスレポートを生成するには

次のgenerate-service-last-accessed-details例では、バックグラウンドジョブを開始して、AWS マネージドAdministratorAccessポリシーを使用してIAMユーザーやその他のエンティティがアクセスするサービスを一覧表示するレポートを生成します。generate-service-last-accessed-details コマンドを実行すると、レポートを作成した後に表示できます。

```
aws iam generate-service-last-accessed-details \  
  --arn arn:aws:iam::aws:policy/AdministratorAccess
```

出力:

```
{
  "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"
}
```

詳細については、AWS IAM「ユーザーガイド」の[「最終アクセス情報 AWS の使用におけるアクセス許可の改良」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GenerateServiceLastAccessedDetails](#)」の「」を参照してください。

get-access-key-last-used

次のコード例は、get-access-key-last-used を使用方法を示しています。

AWS CLI

指定されたアクセスキーの最後の使用時の情報を取得するには

次の例では、アクセスキー ABCDEXAMPLE が最後に使用されたときに関する情報を取得します。

```
aws iam get-access-key-last-used \
  --access-key-id ABCDEXAMPLE
```

出力:

```
{
  "UserName": "Bob",
  "AccessKeyLastUsed": {
    "Region": "us-east-1",
    "ServiceName": "iam",
    "LastUsedDate": "2015-06-16T22:45:00Z"
  }
}
```

詳細については、AWS IAM「ユーザーガイド」の[IAM「ユーザーのアクセスキーの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetAccessKeyLastUsed](#)」の「」を参照してください。

get-account-authorization-details

次のコード例は、get-account-authorization-details を使用する方法を示しています。

AWS CLI

AWS アカウントのIAMユーザー、グループ、ロール、ポリシーを一覧表示するには

次のget-account-authorization-detailsコマンドは、AWS アカウント内のすべてのIAMユーザー、グループ、ロール、ポリシーに関する情報を返します。

```
aws iam get-account-authorization-details
```

出力:

```
{
  "RoleDetailList": [
    {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      },
      "RoleId": "ARO1234567890EXAMPLE",
      "CreateDate": "2014-07-30T17:09:20Z",
      "InstanceProfileList": [
        {
          "InstanceProfileId": "AIPA1234567890EXAMPLE",
          "Roles": [
            {
              "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Sid": "",
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": "ec2.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
},
"RoleId": "AROA1234567890EXAMPLE",
"CreateDate": "2014-07-30T17:09:20Z",
"RoleName": "EC2role",
"Path": "/",
"Arn": "arn:aws:iam::123456789012:role/EC2role"
}
],
"CreateDate": "2014-07-30T17:09:20Z",
"InstanceProfileName": "EC2role",
"Path": "/",
"Arn": "arn:aws:iam::123456789012:instance-profile/EC2role"
}
],
"RoleName": "EC2role",
"Path": "/",
"AttachedManagedPolicies": [
    {
        "PolicyName": "AmazonS3FullAccess",
        "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
    },
    {
        "PolicyName": "AmazonDynamoDBFullAccess",
        "PolicyArn": "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess"
    }
],
"RoleLastUsed": {
    "Region": "us-west-2",
    "LastUsedDate": "2019-11-13T17:30:00Z"
},
"RolePolicyList": [],
"Arn": "arn:aws:iam::123456789012:role/EC2role"
}
],
"GroupDetailList": [
    {
        "GroupId": "AIDA1234567890EXAMPLE",
```

```
    "AttachedManagedPolicies": {
      "PolicyName": "AdministratorAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
    },
    "GroupName": "Admins",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:group/Admins",
    "CreateDate": "2013-10-14T18:32:24Z",
    "GroupPolicyList": []
  },
  {
    "GroupId": "AIDA1234567890EXAMPLE",
    "AttachedManagedPolicies": {
      "PolicyName": "PowerUserAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
    },
    "GroupName": "Dev",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:group/Dev",
    "CreateDate": "2013-10-14T18:33:55Z",
    "GroupPolicyList": []
  },
  {
    "GroupId": "AIDA1234567890EXAMPLE",
    "AttachedManagedPolicies": [],
    "GroupName": "Finance",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:group/Finance",
    "CreateDate": "2013-10-14T18:57:48Z",
    "GroupPolicyList": [
      {
        "PolicyName": "policygen-201310141157",
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "aws-portal:*",
              "Sid": "Stmnt1381777017000",
              "Resource": "*",
              "Effect": "Allow"
            }
          ]
        }
      }
    ]
  }
}
```

```
    ]
  }
],
"UserDetailList": [
  {
    "UserName": "Alice",
    "GroupList": [
      "Admins"
    ],
    "CreateDate": "2013-10-14T18:32:24Z",
    "UserId": "AIDA1234567890EXAMPLE",
    "UserPolicyList": [],
    "Path": "/",
    "AttachedManagedPolicies": [],
    "Arn": "arn:aws:iam::123456789012:user/Alice"
  },
  {
    "UserName": "Bob",
    "GroupList": [
      "Admins"
    ],
    "CreateDate": "2013-10-14T18:32:25Z",
    "UserId": "AIDA1234567890EXAMPLE",
    "UserPolicyList": [
      {
        "PolicyName": "DenyBillingAndIAMPolicy",
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": {
            "Effect": "Deny",
            "Action": [
              "aws-portal:*",
              "iam:*"
            ],
            "Resource": "*"
          }
        }
      }
    ],
    "Path": "/",
    "AttachedManagedPolicies": [],
    "Arn": "arn:aws:iam::123456789012:user/Bob"
  },
  {
```



```
    "UserName": "Charlie",
    "GroupList": [
      "Dev"
    ],
    "CreateDate": "2013-10-14T18:33:56Z",
    "UserId": "AIDA1234567890EXAMPLE",
    "UserPolicyList": [],
    "Path": "/",
    "AttachedManagedPolicies": [],
    "Arn": "arn:aws:iam::123456789012:user/Charlie"
  }
],
"Policies": [
  {
    "PolicyName": "create-update-delete-set-managed-policies",
    "CreateDate": "2015-02-06T19:58:34Z",
    "AttachmentCount": 1,
    "IsAttachable": true,
    "PolicyId": "ANPA1234567890EXAMPLE",
    "DefaultVersionId": "v1",
    "PolicyVersionList": [
      {
        "CreateDate": "2015-02-06T19:58:34Z",
        "VersionId": "v1",
        "Document": {
          "Version": "2012-10-17",
          "Statement": {
            "Effect": "Allow",
            "Action": [
              "iam:CreatePolicy",
              "iam:CreatePolicyVersion",
              "iam>DeletePolicy",
              "iam>DeletePolicyVersion",
              "iam:GetPolicy",
              "iam:GetPolicyVersion",
              "iam>ListPolicies",
              "iam>ListPolicyVersions",
              "iam:SetDefaultPolicyVersion"
            ],
            "Resource": "*"
          }
        }
      },
      {
        "IsDefaultVersion": true
      }
    ]
  }
]
```

```

    ],
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/create-update-delete-set-
managed-policies",
    "UpdateDate": "2015-02-06T19:58:34Z"
  },
  {
    "PolicyName": "S3-read-only-specific-bucket",
    "CreateDate": "2015-01-21T21:39:41Z",
    "AttachmentCount": 1,
    "IsAttachable": true,
    "PolicyId": "ANPA1234567890EXAMPLE",
    "DefaultVersionId": "v1",
    "PolicyVersionList": [
      {
        "CreateDate": "2015-01-21T21:39:41Z",
        "VersionId": "v1",
        "Document": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Action": [
                "s3:Get*",
                "s3:List*"
              ],
              "Resource": [
                "arn:aws:s3:::example-bucket",
                "arn:aws:s3:::example-bucket/*"
              ]
            }
          ]
        },
        "IsDefaultVersion": true
      }
    ],
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/S3-read-only-specific-bucket",
    "UpdateDate": "2015-01-21T23:39:41Z"
  },
  {
    "PolicyName": "AmazonEC2FullAccess",
    "CreateDate": "2015-02-06T18:40:15Z",
    "AttachmentCount": 1,

```

```
"IsAttachable": true,
"PolicyId": "ANPA1234567890EXAMPLE",
"DefaultVersionId": "v1",
"PolicyVersionList": [
  {
    "CreateDate": "2014-10-30T20:59:46Z",
    "VersionId": "v1",
    "Document": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "ec2:*",
          "Effect": "Allow",
          "Resource": "*"
        },
        {
          "Effect": "Allow",
          "Action": "elasticloadbalancing:*",
          "Resource": "*"
        },
        {
          "Effect": "Allow",
          "Action": "cloudwatch:*",
          "Resource": "*"
        },
        {
          "Effect": "Allow",
          "Action": "autoscaling:*",
          "Resource": "*"
        }
      ]
    },
    "IsDefaultVersion": true
  }
],
"Path": "/",
"Arn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess",
"UpdateDate": "2015-02-06T18:40:15Z"
}
],
"Marker": "EXAMPLEkakv9BCuUNFDtxWSyfetYwEx2ADc8dnzfvERF5S6YmvXKx41t6gCl/
eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
"IsTruncated": true
```

```
}
```

詳細については、AWS IAM 「[ユーザーガイドAWS](#)」の[セキュリティ監査ガイドライン](#)を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetAccountAuthorizationDetails](#)」の「」を参照してください。

get-account-password-policy

次のコード例は、get-account-password-policy を使用する方法を示しています。

AWS CLI

現在のアカウントのパスワードポリシーを表示するには

次の get-account-password-policy コマンドは、現在のアカウントのパスワードポリシーに関する詳細を表示します。

```
aws iam get-account-password-policy
```

出力:

```
{
  "PasswordPolicy": {
    "AllowUsersToChangePassword": false,
    "RequireLowercaseCharacters": false,
    "RequireUppercaseCharacters": false,
    "MinimumPasswordLength": 8,
    "RequireNumbers": true,
    "RequireSymbols": true
  }
}
```

アカウントのためにパスワードポリシーが定義されていない場合、コマンドは NoSuchEntity エラーを返します。

詳細については、「[ユーザーガイド](#)」のIAM「[ユーザーのアカウントパスワードポリシーの設定](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「[コマンドリファレンスGetAccountPasswordPolicy](#)」の「」を参照してください。

get-account-summary

次の例は、get-account-summary を使用する方法を説明しています。

AWS CLI

現在のアカウントのIAMエンティティの使用状況とIAMクォータに関する情報を取得するには

次のget-account-summaryコマンドは、アカウントの現在のIAMエンティティ使用量と現在のIAMエンティティクォータに関する情報を返します。

```
aws iam get-account-summary
```

出力:

```
{
  "SummaryMap": {
    "UsersQuota": 5000,
    "GroupsQuota": 100,
    "InstanceProfiles": 6,
    "SigningCertificatesPerUserQuota": 2,
    "AccountAccessKeysPresent": 0,
    "RolesQuota": 250,
    "RolePolicySizeQuota": 10240,
    "AccountSigningCertificatesPresent": 0,
    "Users": 27,
    "ServerCertificatesQuota": 20,
    "ServerCertificates": 0,
    "AssumeRolePolicySizeQuota": 2048,
    "Groups": 7,
    "MFADevicesInUse": 1,
    "Roles": 3,
    "AccountMFAEnabled": 1,
    "MFADevices": 3,
    "GroupsPerUserQuota": 10,
    "GroupPolicySizeQuota": 5120,
    "InstanceProfilesQuota": 100,
    "AccessKeysPerUserQuota": 2,
    "Providers": 0,
    "UserPolicySizeQuota": 2048
  }
}
```

エンティティの制限の詳細については、AWS IAM「ユーザーガイド」の「[IAMおよび AWS STS クォータ](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetAccountSummary](#)」の「」を参照してください。

get-context-keys-for-custom-policy

次のコード例は、`get-context-keys-for-custom-policy` を使用する方法を示しています。

AWS CLI

例 1: コマンドラインでパラメータとして提供される 1 つ以上のカスタムJSONポリシーによって参照されるコンテキストキーを一覧表示するには

次の `get-context-keys-for-custom-policy` コマンドは、指定された各ポリシーを解析し、それらのポリシーが使用するコンテキストキーを一覧表示します。このコマンドを使用して、ポリシーシミュレータコマンド `simulate-custom-policy` と `simulate-custom-policy` を正常に使用するために指定する必要があるコンテキストキー値を特定します。`get-context-keys-for-custom-policy` コマンドを使用して、IAMユーザーまたはロールに関連付けられたすべてのポリシーで使用されるコンテキストキーのリストを取得することもできます。`file://` で始まるパラメータ値は、ファイルを読み取り、ファイル名自体ではなくその内容をパラメータの値として使用するようにコマンドに指示します。

```
aws iam get-context-keys-for-custom-policy \
  --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"arn:aws:dynamodb:us-
west-2:123456789012:table/${aws:username}","Condition":{"DateGreaterThan":
{"aws:CurrentTime":"2015-08-16T12:00:00Z"}}}]'
```

出力:

```
{
  "ContextKeyNames": [
    "aws:username",
    "aws:CurrentTime"
  ]
}
```

例 2: ファイル入力として提供される 1 つ以上のカスタムJSONポリシーによって参照されるコンテキストキーを一覧表示するには

次の `get-context-keys-for-custom-policy` コマンドは前の例と同じですが、ポリシーがパラメータとしてではなくファイルで提供される点が異なります。コマンドはJSON構造JSONのリストではなく文字列のリストを想定しているため、ファイルを1つに折りたたむことはできませんが、次のように構造化する必要があります。

```
[
  "Policy1",
  "Policy2"
]
```

そのため、例えば、前の例のポリシーを含むファイルは次のようになっている必要があります。ポリシー文字列内に埋め込まれている各二重引用符の前に「バックスラッシュ」を付けてエスケープする必要があります。

```
[ {"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Action": "dynamodb:*", "Resource": "arn:aws:dynamodb:us-west-2:128716708097:table/${aws:username}", "Condition": {"DateGreaterThan": {"aws:CurrentTime": "2015-08-16T12:00:00Z"}}}] } ]
```

その後、このファイルを次のコマンドに送信できます。

```
aws iam get-context-keys-for-custom-policy \
  --policy-input-list file://policyfile.json
```

出力:

```
{
  "ContextKeyNames": [
    "aws:username",
    "aws:CurrentTime"
  ]
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[ポリシーシミュレーターの使用 \(AWS CLI および AWS API\)](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetContextKeysForCustomPolicy](#)」の「」を参照してください。

get-context-keys-for-principal-policy

次のコード例は、get-context-keys-for-principal-policy を使用方法を示しています。

AWS CLI

IAMプリンシパルに関連付けられたすべてのポリシーによって参照されるコンテキストキーを一覧表示するには

次の get-context-keys-for-principal-policy コマンドは、ユーザー saanvi とそのユーザーがメンバーとなっているグループにアタッチされているすべてのポリシーを取得します。次に、このコマンドは各ポリシーを解析し、それらのポリシーで使用されているコンテキストキーを一覧表示します。このコマンドを使用して、simulate-custom-policy および simulate-principal-policy コマンドを正常に使用するために指定する必要があるコンテキストキー値を特定します。get-context-keys-for-custom-policy コマンドを使用して、任意のJSONポリシーで使用されるコンテキストキーのリストを取得することもできます。

```
aws iam get-context-keys-for-principal-policy \  
  --policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

出力:

```
{  
  "ContextKeyNames": [  
    "aws:username",  
    "aws:CurrentTime"  
  ]  
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[ポリシーシミュレーターの使用 \(AWS CLI および AWS API\)](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetContextKeysForPrincipalPolicy](#)」の「」を参照してください。

get-credential-report

次の例は、get-credential-report を使用方法を説明しています。

AWS CLI

認証情報レポートを取得するには

この例では、返されたレポートを開き、それをテキスト行の配列としてパイプラインに出力します。

```
aws iam get-credential-report
```

出力:

```
{
  "GeneratedTime": "2015-06-17T19:11:50Z",
  "ReportFormat": "text/csv"
}
```

詳細については、AWS IAM「ユーザーガイド」の[AWS「アカウントの認証情報レポートの取得」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetCredentialReport](#)」の「」を参照してください。

get-group-policy

次の例は、get-group-policy を使用する方法を説明しています。

AWS CLI

IAMグループにアタッチされたポリシーに関する情報を取得するには

次の get-group-policy コマンドは、Test-Group という名前のグループにアタッチされている指定されたポリシーに関する情報を取得します。

```
aws iam get-group-policy \
  --group-name Test-Group \
  --policy-name S3-ReadOnly-Policy
```

出力:

```
{
  "GroupName": "Test-Group",
  "PolicyDocument": {
```

```
    "Statement": [
      {
        "Action": [
          "s3:Get*",
          "s3:List*"
        ],
        "Resource": "*",
        "Effect": "Allow"
      }
    ],
    "PolicyName": "S3-ReadOnly-Policy"
  }
}
```

詳細については、「ユーザーガイド」の [IAM「ポリシーの管理」](#) を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [GetGroupPolicy](#)」の「」を参照してください。

get-group

次の例は、get-group を使用する方法を説明しています。

AWS CLI

IAMグループを取得するには

この例では、IAMグループに関する詳細を返しますAdmins。

```
aws iam get-group \
  --group-name Admins
```

出力:

```
{
  "Group": {
    "Path": "/",
    "CreateDate": "2015-06-16T19:41:48Z",
    "GroupId": "AIDGPM9R04H3FEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/Admins",
    "GroupName": "Admins"
  }
}
```

```
  },  
  "Users": []  
}
```

詳細については、「ユーザーガイド」の [IAM「アイデンティティ \(ユーザー、ユーザーグループ、ロール\)」](#) を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [GetGroup](#)」の「」を参照してください。

get-instance-profile

次の例は、get-instance-profile を使用する方法を説明しています。

AWS CLI

インスタンスプロファイルの情報を取得するには

次の get-instance-profile コマンドは、ExampleInstanceProfile という名前のインスタンスプロファイルに関する情報を取得します。

```
aws iam get-instance-profile \  
  --instance-profile-name ExampleInstanceProfile
```

出力:

```
{  
  "InstanceProfile": {  
    "InstanceProfileId": "AID2MAB8DPLSRHEXAMPLE",  
    "Roles": [  
      {  
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",  
        "RoleId": "AIDGPMS9R04H3FEXAMPLE",  
        "CreateDate": "2013-01-09T06:33:26Z",  
        "RoleName": "Test-Role",  
        "Path": "/",  
        "Arn": "arn:aws:iam::336924118301:role/Test-Role"  
      }  
    ],  
    "CreateDate": "2013-06-12T23:52:02Z",  
    "InstanceProfileName": "ExampleInstanceProfile",  
    "Path": "/",  
  }  
}
```

```
    "Arn": "arn:aws:iam::336924118301:instance-profile/ExampleInstanceProfile"
  }
}
```

詳細については、「ユーザーガイド」の「[インスタンスプロファイルの使用](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceProfile](#)」の「」を参照してください。

get-login-profile

次のコード例は、get-login-profile を使用する方法を示しています。

AWS CLI

IAMユーザーのパスワード情報を取得するには

次のget-login-profileコマンドは、という名前のIAMユーザーのパスワードに関する情報を取得しますBob。

```
aws iam get-login-profile \
  --user-name Bob
```

出力:

```
{
  "LoginProfile": {
    "UserName": "Bob",
    "CreateDate": "2012-09-21T23:03:39Z"
  }
}
```

get-login-profile コマンドを使用して、IAMユーザーがパスワードを持っていることを確認できます。ユーザーのためにパスワードが定義されていない場合、コマンドは NoSuchEntity エラーを返します。

このコマンドを使用してパスワードを表示することはできません。パスワードを忘れた場合は、ユーザーのパスワードをリセットできます (update-login-profile)。または、ユーザーのログインプロファイルを削除し (delete-login-profile)、新しいログインプロファイルを作成することもできます (create-login-profile)。

詳細については、「[ユーザーガイド](#)」のIAM「[ユーザーのパスワードの管理](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンスGetLoginProfile](#)」の「」を参照してください。

get-mfa-device

次のコード例は、get-mfa-device を使用する方法を示しています。

AWS CLI

FIDO セキュリティキーに関する情報を取得するには

次のget-mfa-deviceコマンド例では、指定されたFIDOセキュリティキーに関する情報を取得します。

```
aws iam get-mfa-device \  
  --serial-number arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-  
EXAMPLEBN5FHTECLFG7EXAMPLE
```

出力:

```
{  
  "UserName": "alice",  
  "SerialNumber": "arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-  
EXAMPLEBN5FHTECLFG7EXAMPLE",  
  "EnableDate": "2023-09-19T01:49:18+00:00",  
  "Certifications": {  
    "FIDO": "L1"  
  }  
}
```

詳細については、AWS IAM ユーザーガイドの「[での多要素認証 \(MFA\) の使用 AWS](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetMfaDevice](#)」の「」を参照してください。

get-open-id-connect-provider

次の例は、get-open-id-connect-provider を使用する方法を説明しています。

AWS CLI

指定した OpenID Connect プロバイダーに関する情報を返すには

この例では、ARNである OpenID Connect プロバイダーの詳細を返します
arn:aws:iam::123456789012:oidc-provider/server.example.com。

```
aws iam get-open-id-connect-provider \
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

出力:

```
{
  "Url": "server.example.com"
  "CreateDate": "2015-06-16T19:41:48Z",
  "ThumbprintList": [
    "12345abcdefghijk67890lmnopqrst987example"
  ],
  "ClientIDList": [
    "example-application-ID"
  ]
}
```

詳細については、AWS IAM 「ユーザーガイド」の[OpenID Connect \(OIDC\) ID プロバイダーの作成](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOpenIdConnectProvider](#)」の「」を参照してください。

get-organizations-access-report

次のコード例は、get-organizations-access-report を使用する方法を示しています。

AWS CLI

アクセスレポートを取得するには

次のget-organizations-access-report例では、AWS Organizations エンティティの以前に生成されたアクセスレポートを表示します。レポートを生成するには、generate-organizations-access-report コマンドを使用します。

```
aws iam get-organizations-access-report \  
--job-id a8b6c06f-aaa4-8xmp-28bc-81da71836359
```

出力:

```
{  
  "JobStatus": "COMPLETED",  
  "JobCreationDate": "2019-09-30T06:53:36.187Z",  
  "JobCompletionDate": "2019-09-30T06:53:37.547Z",  
  "NumberOfServicesAccessible": 188,  
  "NumberOfServicesNotAccessed": 171,  
  "AccessDetails": [  
    {  
      "ServiceName": "Alexa for Business",  
      "ServiceNamespace": "a4b",  
      "TotalAuthenticatedEntities": 0  
    },  
    ...  
  ]  
}
```

詳細については、AWS IAM「ユーザーガイド」の「[最終アクセス情報 AWS を使用した のアクセス許可の改良](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetOrganizationsAccessReport](#)」の「」を参照してください。

get-policy-version

次のコード例は、get-policy-version を使用する方法を示しています。

AWS CLI

指定された管理ポリシーの指定されたバージョンに関する情報を取得するには

この例では、ARNであるポリシーの v2 バージョンのポリシードキュメントを返します
arn:aws:iam::123456789012:policy/MyManagedPolicy。

```
aws iam get-policy-version \  
--policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
--version-id v2
```

出力:

```
{
  "PolicyVersion": {
    "Document": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": "iam:*",
          "Resource": "*"
        }
      ]
    },
    "VersionId": "v2",
    "IsDefaultVersion": true,
    "CreateDate": "2023-04-11T00:22:54+00:00"
  }
}
```

詳細については、「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicyVersion](#)」の「」を参照してください。

get-policy

次のコード例は、get-policy を使用する方法を示しています。

AWS CLI

指定した管理ポリシーに関する情報を取得するには

この例では、ARN がである マネージドポリシーの詳細を返しますarn:aws:iam::123456789012:policy/MySamplePolicy。

```
aws iam get-policy \
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

出力:

```
{
  "Policy": {
```



```
"PolicyName": "MySamplePolicy",
"CreateDate": "2015-06-17T19:23:32Z",
"AttachmentCount": 0,
"IsAttachable": true,
"PolicyId": "Z27SI6FQMGNO2EXAMPLE1",
"DefaultVersionId": "v1",
"Path": "/",
"Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",
"UpdateDate": "2015-06-17T19:23:32Z"
}
}
```

詳細については、AWS IAM「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetPolicy](#)」の「」を参照してください。

get-role-policy

次のコード例は、get-role-policy を使用する方法を示しています。

AWS CLI

IAMロールにアタッチされたポリシーに関する情報を取得するには

次の get-role-policy コマンドは、Test-Role という名前のロールにアタッチされている指定されたポリシーに関する情報を取得します。

```
aws iam get-role-policy \
  --role-name Test-Role \
  --policy-name ExamplePolicy
```

出力:

```
{
  "RoleName": "Test-Role",
  "PolicyDocument": {
    "Statement": [
      {
        "Action": [
          "s3:ListBucket",
          "s3:Put*"
        ]
      }
    ]
  }
}
```

```

        "s3:Get*",
        "s3:*MultipartUpload*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "1"
  }
]
}
"PolicyName": "ExamplePolicy"
}

```

詳細については、「ユーザーガイド」の [IAM「ロールの作成」](#) を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [GetRolePolicy](#)」の「」を参照してください。

get-role

次のコード例は、get-role を使用する方法を示しています。

AWS CLI

IAMロールに関する情報を取得するには

次の get-role コマンドは、Test-Role という名前のロールに関する情報を取得します。

```
aws iam get-role \
  --role-name Test-Role
```

出力:

```
{
  "Role": {
    "Description": "Test Role",
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "MaxSessionDuration": 3600,
    "RoleId": "ARO0A1234567890EXAMPLE",
    "CreateDate": "2019-11-13T16:45:56Z",
    "RoleName": "Test-Role",
    "Path": "/",
    "RoleLastUsed": {
      "Region": "us-east-1",

```

```
        "LastUsedDate": "2019-11-13T17:14:00Z"
      },
      "Arn": "arn:aws:iam::123456789012:role/Test-Role"
    }
  }
}
```

このコマンドは、ロールにアタッチされている信頼ポリシーを表示します。ロールにアタッチされているアクセス許可ポリシーを一覧表示するには、`list-role-policies` コマンドを使用します。

詳細については、「ユーザーガイド」の [IAM「ロールの作成」](#) を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [GetRole](#)」の「」を参照してください。

get-saml-provider

次のコード例は、`get-saml-provider` を使用する方法を示しています。

AWS CLI

SAMLプロバイダーメタドキュメントを取得するには

この例では、ARMである SAML 2.0 プロバイダーの詳細を取得します。arn:aws:iam::123456789012:saml-provider/SAMLADFS。レスポンスには、ID プロバイダーから取得したメタデータドキュメントと、プロバイダーエンティティの作成 AWS SAML 日と有効期限が含まれます。

```
aws iam get-saml-provider \
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

出力:

```
{
  "SAMLMetadataDocument": "...SAMLMetadataDocument-XML...",
  "CreateDate": "2017-03-06T22:29:46+00:00",
  "ValidUntil": "2117-03-06T22:29:46.433000+00:00",
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    }
  ],
}
```

```

    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}

```

詳細については、「ユーザーガイド」の[IAM SAML「アイデンティティプロバイダーの作成」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[GetSamlProvider](#)」の「」を参照してください。

get-server-certificate

次の例は、get-server-certificate を使用する方法を説明しています。

AWS CLI

AWS アカウントのサーバー証明書の詳細を取得するには

次のget-server-certificate コマンドは、AWS アカウントで指定されたサーバー証明書に関する詳細をすべて取得します。

```

aws iam get-server-certificate \
  --server-certificate-name myUpdatedServerCertificate

```

出力:

```

{
  "ServerCertificate": {
    "ServerCertificateMetadata": {
      "Path": "/",
      "ServerCertificateName": "myUpdatedServerCertificate",
      "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:server-certificate/myUpdatedServerCertificate",
      "UploadDate": "2019-04-22T21:13:44+00:00",
      "Expiration": "2019-10-15T22:23:16+00:00"
    },
    "CertificateBody": "-----BEGIN CERTIFICATE-----
      MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
      VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6

```

```

b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZlZlbnR1eWVhZAd
BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMVCVVMxMzA0NTIxWjCBiDELMAKGA1UEBh
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZlZlbnR1eWVhZAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
"CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCcAfICQD6md
7oRw0uX0jANBgkqhkiG9w0BAQQUFADCBiDELMAKGA1UEBhMVCVVMxMzA0NTIxWjCBiDELMAKGA1UEBh
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQ
TC01BTSBDb25zb2x1MRIwEAYDVQsQQDEw1UZlZlbnR1eWVhZAdBgkqhkiG9w0BCQ
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBh
MVCVVMxMzA0NTIxWjCBiDELMAKGA1UEBhMgYDVQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZB
WF6b24xFDASBgNVBA5TC01BTSBDb2d5zb2x1MRIwEAYDVQQDEw1UZlZlbnR1eWVh
HzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySwTc2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gjpEIbb30hjZnzcvcQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVvxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0F1kbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEWEg5vb25lQGFTsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
}
}

```

AWS アカウントで使用できるサーバー証明書を一覧表示するには、`list-server-certificates` コマンドを使用します。

詳細については、AWS IAM 「ユーザーガイド」の「[でのサーバー証明書の管理IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServerCertificate](#)」の「」を参照してください。

get-service-last-accessed-details-with-entities

次のコード例は、`get-service-last-accessed-details-with-entities` を使用する方法を示しています。

AWS CLI

サービスの詳細を含むサービスアクセスレポートを取得するには

次の`get-service-last-accessed-details-with-entities`例では、指定されたサービスにアクセスしたIAMユーザーやその他のエンティティに関する詳細を含むレポートを取得します。レポートを生成するには、`generate-service-last-accessed-details` コマンドを使用します。名前空間でアクセスされるサービスのリストを取得するには、`get-service-last-accessed-details` を使用します。

```
aws iam get-service-last-accessed-details-with-entities \  
  --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \  
  --service-namespace lambda
```

出力:

```
{  
  "JobStatus": "COMPLETED",  
  "JobCreationDate": "2019-10-01T03:55:41.756Z",  
  "JobCompletionDate": "2019-10-01T03:55:42.533Z",  
  "EntityDetailsList": [  
    {  
      "EntityInfo": {  
        "Arn": "arn:aws:iam::123456789012:user/admin",  
        "Name": "admin",  
        "Type": "USER",  
        "Id": "AIDAI02XMPLENQEXAMPLE",  
        "Path": "/"  
      },  
      "LastAuthenticated": "2019-09-30T23:02:00Z"  
    },  
    {  
      "EntityInfo": {  
        "Arn": "arn:aws:iam::123456789012:user/developer",  
        "Name": "developer",  
        "Type": "USER",  
        "Id": "AIDAIBEYXMPL2YEXAMPLE",  
        "Path": "/"  
      },  
      "LastAuthenticated": "2019-09-16T19:34:00Z"  
    }  
  ]  
}
```

```
}
```

詳細については、AWS IAM「ユーザーガイド」の[「最終アクセス情報 AWS を使用した でのアクセス許可の改良」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetServiceLastAccessedDetailsWithEntities](#)」の「」を参照してください。

get-service-last-accessed-details

次のコード例は、get-service-last-accessed-details を使用する方法を示しています。

AWS CLI

サービスアクセスレポートを取得するには

次のget-service-last-accessed-details例では、IAMエンティティによってアクセスされたサービスを一覧表示する、以前に生成されたレポートを取得します。レポートを生成するには、generate-service-last-accessed-details コマンドを使用します。

```
aws iam get-service-last-accessed-details \  
  --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

出力:

```
{  
  "JobStatus": "COMPLETED",  
  "JobCreationDate": "2019-10-01T03:50:35.929Z",  
  "ServicesLastAccessed": [  
    ...  
    {  
      "ServiceName": "AWS Lambda",  
      "LastAuthenticated": "2019-09-30T23:02:00Z",  
      "ServiceNamespace": "lambda",  
      "LastAuthenticatedEntity": "arn:aws:iam::123456789012:user/admin",  
      "TotalAuthenticatedEntities": 6  
    },  
  ],  
}
```

詳細については、AWS IAM「ユーザーガイド」の[「最終アクセス情報 AWS を使用した でのアクセス許可の改良」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceLastAccessedDetails](#)」の「」を参照してください。

get-service-linked-role-deletion-status

次の例は、get-service-linked-role-deletion-status を使用する方法を説明しています。

AWS CLI

サービスにリンクされたロールの削除リクエストのステータスを確認するには

次の get-service-linked-role-deletion-status の例では、サービスにリンクされたロールを削除するという以前のリクエストのステータスが表示されます。削除オペレーションは非同期で実行されます。リクエストを実行すると、このコマンドのパラメータとして指定した DeletionTaskId の値を取得します。

```
aws iam get-service-linked-role-deletion-status \  
  --deletion-task-id task/aws-service-role/lex.amazonaws.com/  
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

出力:

```
{  
  "Status": "SUCCEEDED"  
}
```

詳細については、AWS IAM 「ユーザーガイド」の「[サービスにリンクされたロールの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceLinkedRoleDeletionStatus](#)」の「」を参照してください。

get-ssh-public-key

次のコード例は、get-ssh-public-key を使用する方法を示しています。

AWS CLI

例 1: エンSSHコードされた形式でIAMユーザーにアタッチされたSSHパブリックキーを取得するには

次のget-ssh-public-keyコマンドは、指定されたSSHパブリックキーをIAMユーザー から取得しますsofia。出力はSSHエンコード中です。

```
aws iam get-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE \  
  --encoding SSH
```

出力:

```
{  
  "SSHPublicKey": {  
    "UserName": "sofia",  
    "SSHPublicKeyId": "APKA123456789EXAMPLE",  
    "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",  
    "SSHPublicKeyBody": "ssh-rsa <<long encoded SSH string>>",  
    "Status": "Inactive",  
    "UploadDate": "2019-04-18T17:04:49+00:00"  
  }  
}
```

例 2: エンPEMコードされた形式でIAMユーザーにアタッチされたSSHパブリックキーを取得するには

次のget-ssh-public-keyコマンドは、指定されたSSHパブリックキーをIAMユーザー から取得しますsofia。出力はPEMエンコード中です。

```
aws iam get-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE \  
  --encoding PEM
```

出力:

```
{  
  "SSHPublicKey": {  
    "UserName": "sofia",  
    "SSHPublicKeyId": "APKA123456789EXAMPLE",  
    "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",  
    "SSHPublicKeyBody": ""-----BEGIN PUBLIC KEY-----\n<<long encoded PEM string>>\n-----END PUBLIC KEY-----\n""  
  }  
}
```

```
    "Status": "Inactive",
    "UploadDate": "2019-04-18T17:04:49+00:00"
  }
}
```

詳細については、AWS IAM「ユーザーガイド」の「[SSHキーと SSH を使用する CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetSshPublicKey](#)」の「」を参照してください。

get-user-policy

次のコード例は、get-user-policy を使用する方法を示しています。

AWS CLI

IAMユーザーのポリシーの詳細を一覧表示するには

次のget-user-policyコマンドは、という名前のIAMユーザーにアタッチされている指定されたポリシーの詳細を一覧表示しますBob。

```
aws iam get-user-policy \
  --user-name Bob \
  --policy-name ExamplePolicy
```

出力:

```
{
  "UserName": "Bob",
  "PolicyName": "ExamplePolicy",
  "PolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": "*",
        "Resource": "*",
        "Effect": "Allow"
      }
    ]
  }
}
```

```
}
```

IAM ユーザーのポリシーのリストを取得するには、`list-user-policies` コマンドを使用します。

詳細については、AWS IAM 「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUserPolicy](#)」の「」を参照してください。

get-user

次のコード例は、`get-user` を使用する方法を示しています。

AWS CLI

IAM ユーザーに関する情報を取得するには

次の`get-user`コマンドは、`Paulo` という名前のIAMユーザーに関する情報を取得します。

```
aws iam get-user \
  --user-name Paulo
```

出力:

```
{
  "User": {
    "UserName": "Paulo",
    "Path": "/",
    "CreateDate": "2019-09-21T23:03:13Z",
    "UserId": "AIDA123456789EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/Paulo"
  }
}
```

詳細については、「ユーザーガイド」のIAM「[ユーザーの管理](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUser](#)」の「」を参照してください。

list-access-keys

次の例は、list-access-keys を使用する方法を説明しています。

AWS CLI

IAMユーザーのアクセスキーIDsを一覧表示するには

次のlist-access-keysコマンドは、 という名前IDsのIAMユーザーのアクセスキーを一覧表示しますBob。

```
aws iam list-access-keys \  
  --user-name Bob
```

出力:

```
{  
  "AccessKeyMetadata": [  
    {  
      "UserName": "Bob",  
      "Status": "Active",  
      "CreateDate": "2013-06-04T18:17:34Z",  
      "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"  
    },  
    {  
      "UserName": "Bob",  
      "Status": "Inactive",  
      "CreateDate": "2013-06-06T20:42:26Z",  
      "AccessKeyId": "AKIAI44QH8DHBEXAMPLE"  
    }  
  ]  
}
```

IAMユーザーのシークレットアクセスキーを一覧表示することはできません。シークレットアクセスキーを紛失した場合は、create-access-keys コマンドを使用して新しいアクセスキーを作成する必要があります。

詳細については、AWS IAM「ユーザーガイド」の[IAM「ユーザーのアクセスキーの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListAccessKeys](#)」の「」を参照してください。

list-account-aliases

次のコード例は、list-account-aliases を使用する方法を示しています。

AWS CLI

アカウントエイリアスを一覧表示するには

次の list-account-aliases コマンドは、現在のアカウントのエイリアスを一覧表示します。

```
aws iam list-account-aliases
```

出力:

```
{
  "AccountAliases": [
    "mycompany"
  ]
}
```

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「アカウント ID とそのエイリアス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccountAliases](#)」の「」を参照してください。

list-attached-group-policies

次の例は、list-attached-group-policies を使用する方法を説明しています。

AWS CLI

指定されたグループにアタッチされているすべての管理ポリシーを一覧表示するには

この例では、AWS アカウントAdminsで という名前ARNsのIAMグループにアタッチされているマネージドポリシーの名前と を返します。

```
aws iam list-attached-group-policies \
  --group-name Admins
```

出力:

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "AdministratorAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
    },
    {
      "PolicyName": "SecurityAudit",
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
    }
  ],
  "IsTruncated": false
}
```

詳細については、「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAttachedGroupPolicies](#)」の「」を参照してください。

list-attached-role-policies

次の例は、list-attached-role-policies を使用する方法を説明しています。

AWS CLI

指定された IAM ロールにアタッチされている管理ポリシーを一覧表示するには

このコマンドは、AWS アカウントSecurityAuditRoleで という名前ARNsのIAMロールにアタッチされた マネージドポリシーの名前と を返します。

```
aws iam list-attached-role-policies \
  --role-name SecurityAuditRole
```

出力:

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "SecurityAudit",
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
    }
  ]
}
```

```
  ],
  "IsTruncated": false
}
```

詳細については、「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAttachedRolePolicies](#)」の「」を参照してください。

list-attached-user-policies

次の例は、list-attached-user-policies を使用する方法を説明しています。

AWS CLI

指定されたユーザーにアタッチされているすべての管理ポリシーを一覧表示するには

このコマンドは、AWS アカウントBobで という名前ARNsのIAMユーザーの名前と マネージドポリシーを返します。

```
aws iam list-attached-user-policies \
  --user-name Bob
```

出力:

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "AdministratorAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
    },
    {
      "PolicyName": "SecurityAudit",
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
    }
  ],
  "IsTruncated": false
}
```

詳細については、AWS IAM 「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAttachedUserPolicies](#)」の「」を参照してください。

list-entities-for-policy

次のコード例は、list-entities-for-policy を使用する方法を示しています。

AWS CLI

指定された管理ポリシーがアタッチされているすべてのユーザー、グループ、ロールを一覧表示するには

この例では、ポリシーがarn:aws:iam::123456789012:policy/TestPolicyアタッチされているIAMグループ、ロール、ユーザーのリストを返します。

```
aws iam list-entities-for-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

出力:

```
{  
  "PolicyGroups": [  
    {  
      "GroupName": "Admins",  
      "GroupId": "AGPACKCEVSQ6C2EXAMPLE"  
    }  
  ],  
  "PolicyUsers": [  
    {  
      "UserName": "Alice",  
      "UserId": "AIDACKCEVSQ6C2EXAMPLE"  
    }  
  ],  
  "PolicyRoles": [  
    {  
      "RoleName": "DevRole",  
      "RoleId": "AROADBQP57FF2AEXAMPLE"  
    }  
  ],  
  "IsTruncated": false  
}
```


詳細については、AWS IAM 「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEntitiesForPolicy](#)」の「」を参照してください。

list-group-policies

次のコード例は、list-group-policies を使用する方法を示しています。

AWS CLI

指定されたグループにアタッチされているすべてのインラインポリシーを一覧表示するには

次のlist-group-policiesコマンドは、現在のアカウントAdminsで という名前のIAMグループにアタッチされているインラインポリシーの名前を一覧表示します。

```
aws iam list-group-policies \  
  --group-name Admins
```

出力:

```
{  
  "PolicyNames": [  
    "AdminRoot",  
    "ExamplePolicy"  
  ]  
}
```

詳細については、「ユーザーガイド」の[IAM「ポリシーの管理](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGroupPolicies](#)」の「」を参照してください。

list-groups-for-user

次のコード例は、list-groups-for-user を使用する方法を示しています。

AWS CLI

IAMユーザーが属するグループを一覧表示するには

次のlist-groups-for-userコマンドは、 という名前のIAMユーザーがBob属するグループを表示します。

```
aws iam list-groups-for-user \  
--user-name Bob
```

出力:

```
{  
  "Groups": [  
    {  
      "Path": "/",  
      "CreateDate": "2013-05-06T01:18:08Z",  
      "GroupId": "AKIAIOSFODNN7EXAMPLE",  
      "Arn": "arn:aws:iam::123456789012:group/Admin",  
      "GroupName": "Admin"  
    },  
    {  
      "Path": "/",  
      "CreateDate": "2013-05-06T01:37:28Z",  
      "GroupId": "AKIAI44QH8DHBEXAMPLE",  
      "Arn": "arn:aws:iam::123456789012:group/s3-Users",  
      "GroupName": "s3-Users"  
    }  
  ]  
}
```

詳細については、「ユーザーガイド」のIAM「[ユーザーグループの管理](#)」を参照してください。
AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンスListGroupsForUser](#)」の「」を参照してください。

list-groups

次のコード例は、list-groups を使用する方法を示しています。

AWS CLI

現在のアカウントのIAMグループを一覧表示するには

次のlist-groupsコマンドは、現在のアカウントのIAMグループを一覧表示します。

```
aws iam list-groups
```

出力:

```
{
  "Groups": [
    {
      "Path": "/",
      "CreateDate": "2013-06-04T20:27:27.972Z",
      "GroupId": "AIDACKCEVSQ6C2EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:group/Admins",
      "GroupName": "Admins"
    },
    {
      "Path": "/",
      "CreateDate": "2013-04-16T20:30:42Z",
      "GroupId": "AIDGPMS9R04H3FEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:group/S3-Admins",
      "GroupName": "S3-Admins"
    }
  ]
}
```

詳細については、「ユーザーガイド」の [IAM「ユーザーグループの管理」](#) を参照してください。

AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [ListGroups](#)」の「」を参照してください。

list-instance-profile-tags

次の例は、list-instance-profile-tags を使用する方法を説明しています。

AWS CLI

インスタンスプロファイルにアタッチされたタグを一覧表示するには

次のlist-instance-profile-tagsコマンドは、指定されたインスタンスプロファイルに関連付けられたタグのリストを取得します。

```
aws iam list-instance-profile-tags \
```

```
--instance-profile-name deployment-role
```

出力:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

詳細については、AWS IAM「ユーザーガイド」の[IAM「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListInstanceProfileTags](#)」の「」を参照してください。

list-instance-profiles-for-role

次のコード例は、list-instance-profiles-for-role を使用する方法を示しています。

AWS CLI

IAMロールのインスタンスプロファイルを一覧表示するには

次の list-instance-profiles-for-role コマンドは、ロール Test-Role に関連付けられているインスタンスプロファイルを一覧表示します。

```
aws iam list-instance-profiles-for-role \  
  --role-name Test-Role
```

出力:

```
{
  "InstanceProfiles": [
    {
```

```
    "InstanceProfileId": "AIDGPMS9R04H3FEXAMPLE",
    "Roles": [
      {
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
        "RoleId": "AIDACKCEVSQ6C2EXAMPLE",
        "CreateDate": "2013-06-07T20:42:15Z",
        "RoleName": "Test-Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/Test-Role"
      }
    ],
    "CreateDate": "2013-06-07T21:05:24Z",
    "InstanceProfileName": "ExampleInstanceProfile",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:instance-profile/
ExampleInstanceProfile"
  }
]
}
```

詳細については、「ユーザーガイド」の[「インスタンスプロファイルの使用」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInstanceProfilesForRole](#)」の「」を参照してください。

list-instance-profiles

次の例は、list-instance-profiles を使用する方法を説明しています。

AWS CLI

アカウントのインスタンスプロファイルを一覧表示するには

次の list-instance-profiles コマンドは、現在のアカウントに関連付けられているインスタンスプロファイルを一覧表示します。

```
aws iam list-instance-profiles
```

出力:

```
{
  "InstanceProfiles": [
```

```
{
  "Path": "/",
  "InstanceProfileName": "example-dev-role",
  "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
  "Arn": "arn:aws:iam::123456789012:instance-profile/example-dev-role",
  "CreateDate": "2023-09-21T18:17:41+00:00",
  "Roles": [
    {
      "Path": "/",
      "RoleName": "example-dev-role",
      "RoleId": "AR0AJ520TH4H7LEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:role/example-dev-role",
      "CreateDate": "2023-09-21T18:17:40+00:00",
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      }
    }
  ]
},
{
  "Path": "/",
  "InstanceProfileName": "example-s3-role",
  "InstanceProfileId": "AIPAJVJVNRIQFREXAMPLE",
  "Arn": "arn:aws:iam::123456789012:instance-profile/example-s3-role",
  "CreateDate": "2023-09-21T18:18:50+00:00",
  "Roles": [
    {
      "Path": "/",
      "RoleName": "example-s3-role",
      "RoleId": "AR0AINUBC507XLEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:role/example-s3-role",
      "CreateDate": "2023-09-21T18:18:49+00:00",
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
}
```

詳細については、「ユーザーガイド」の[「インスタンスプロファイルの使用」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInstanceProfiles](#)」の「」を参照してください。

list-mfa-device-tags

次のコード例は、list-mfa-device-tags を使用する方法を示しています。

AWS CLI

MFA デバイスに添付されたタグを一覧表示するには

次のlist-mfa-device-tagsコマンドは、指定されたMFAデバイスに関連付けられたタグのリストを取得します。

```
aws iam list-mfa-device-tags \
  --serial-number arn:aws:iam::123456789012:mfa/alice
```

出力:

```
{
  "Tags": [
    {
      "Key": "DeptID",
```

```
        "Value": "123456"
      },
      {
        "Key": "Department",
        "Value": "Accounting"
      }
    ]
  }
}
```

詳細については、AWS IAM「ユーザーガイド」の[IAM「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListMfaDeviceTags](#)」の「」を参照してください。

list-mfa-devices

次の例は、list-mfa-devices を使用する方法を説明しています。

AWS CLI

指定されたユーザーのすべてのMFAデバイスを一覧表示するには

この例では、IAMユーザー に割り当てられたMFAデバイスに関する詳細を返しますBob。

```
aws iam list-mfa-devices \
  --user-name Bob
```

出力:

```
{
  "MFADevices": [
    {
      "UserName": "Bob",
      "SerialNumber": "arn:aws:iam::123456789012:mfa/Bob",
      "EnableDate": "2019-10-28T20:37:09+00:00"
    },
    {
      "UserName": "Bob",
      "SerialNumber": "GAKT12345678",
      "EnableDate": "2023-02-18T21:44:42+00:00"
    }
  ],
}
```



```
{
  "UserName": "Bob",
  "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey1-7XNL7NFNLZ123456789EXAMPLE",
  "EnableDate": "2023-09-19T02:25:35+00:00"
},
{
  "UserName": "Bob",
  "SerialNumber": "arn:aws:iam::123456789012:u2f/user/Bob/
fidosecuritykey2-VDRQTDBBN5123456789EXAMPLE",
  "EnableDate": "2023-09-19T01:49:18+00:00"
}
]
```

詳細については、AWS IAM「ユーザーガイド」の「[での多要素認証 \(MFA\) の使用 AWS](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListMfaDevices](#)」の「」を参照してください。

list-open-id-connect-provider-tags

次の例は、list-open-id-connect-provider-tags を使用する方法を説明しています。

AWS CLI

OpenID Connect (OIDC) 互換 ID プロバイダーにアタッチされたタグを一覧表示するには

次のlist-open-id-connect-provider-tagsコマンドは、指定された OIDC ID プロバイダーに関連付けられているタグのリストを取得します。

```
aws iam list-open-id-connect-provider-tags \
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

出力:

```
{
  "Tags": [
    {
      "Key": "DeptID",
```

```
        "Value": "123456"
      },
      {
        "Key": "Department",
        "Value": "Accounting"
      }
    ]
  }
}
```

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOpenIdConnectProviderTags](#)」の「」を参照してください。

list-open-id-connect-providers

次の例は、list-open-id-connect-providers を使用する方法を説明しています。

AWS CLI

AWS アカウントの OpenID Connect プロバイダーに関する情報を一覧表示するには

この例では、現在の AWS アカウントで定義されている ARNS すべての OpenID Connect プロバイダーのリストを返します。

```
aws iam list-open-id-connect-providers
```

出力:

```
{
  "OpenIDConnectProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
    }
  ]
}
```

詳細については、AWS IAM 「ユーザーガイド」の[OpenID Connect \(OIDC\) ID プロバイダーの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListOpenIdConnectProviders](#)」の「」を参照してください。

list-policies-granting-service-access

次のコード例は、list-policies-granting-service-access を使用方法を示しています。

AWS CLI

指定されたサービスへのプリンシパルアクセスを許可するポリシーを一覧表示するには

次のlist-policies-granting-service-access例では、IAMユーザーに AWS CodeCommit サービスsofiaへのアクセスを許可するポリシーのリストを取得します。

```
aws iam list-policies-granting-service-access \
  --arn arn:aws:iam::123456789012:user/sofia \
  --service-namespaces codecommit
```

出力:

```
{
  "PoliciesGrantingServiceAccess": [
    {
      "ServiceNamespace": "codecommit",
      "Policies": [
        {
          "PolicyName": "Grant-Sofia-Access-To-CodeCommit",
          "PolicyType": "INLINE",
          "EntityType": "USER",
          "EntityName": "sofia"
        }
      ]
    }
  ],
  "IsTruncated": false
}
```

詳細については、AWS IAM 「ユーザーガイド」の「[IAMで使用する CodeCommit: Git 認証情報、SSHキー、および AWS アクセスキー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListPoliciesGrantingServiceAccess](#)」の「」を参照してください。

list-policies

次のコード例は、list-policies を使用する方法を示しています。

AWS CLI

AWS アカウントで利用可能な管理ポリシーを一覧表示するには

この例では、現在の AWS アカウントで利用可能な最初の 2 つの管理ポリシーのコレクションを返します。

```
aws iam list-policies \  
  --max-items 3
```

出力:

```
{  
  "Policies": [  
    {  
      "PolicyName": "AWSCloudTrailAccessPolicy",  
      "PolicyId": "ANPAXQE2B5PJ7YEXAMPLE",  
      "Arn": "arn:aws:iam::123456789012:policy/AWSCloudTrailAccessPolicy",  
      "Path": "/",  
      "DefaultVersionId": "v1",  
      "AttachmentCount": 0,  
      "PermissionsBoundaryUsageCount": 0,  
      "IsAttachable": true,  
      "CreateDate": "2019-09-04T17:43:42+00:00",  
      "UpdateDate": "2019-09-04T17:43:42+00:00"  
    },  
    {  
      "PolicyName": "AdministratorAccess",  
      "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",  
      "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",  
      "Path": "/",  
      "DefaultVersionId": "v1",  
      "AttachmentCount": 6,  
      "PermissionsBoundaryUsageCount": 0,  
      "IsAttachable": true,  
    }  
  ]  
}
```

```
    "CreateDate": "2015-02-06T18:39:46+00:00",
    "UpdateDate": "2015-02-06T18:39:46+00:00"
  },
  {
    "PolicyName": "PowerUserAccess",
    "PolicyId": "ANPAJYRXTHIB4FOVS3ZXS",
    "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
    "Path": "/",
    "DefaultVersionId": "v5",
    "AttachmentCount": 1,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2015-02-06T18:39:47+00:00",
    "UpdateDate": "2023-07-06T22:04:00+00:00"
  }
],
"NextToken": "EXAMPLErZXIi0iBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQi0iA4fQ=="
}
```

詳細については、AWS IAM「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListPolicies](#)」の「」を参照してください。

list-policy-tags

次のコード例は、list-policy-tags を使用する方法を示しています。

AWS CLI

マネージドポリシーにアタッチされたタグを一覧表示するには

次のlist-policy-tagsコマンドは、指定された管理ポリシーに関連付けられたタグのリストを取得します。

```
aws iam list-policy-tags \
  --policy-arn arn:aws:iam::123456789012:policy/billing-access
```

出力:

```
{
```

```
"Tags": [
  {
    "Key": "DeptID",
    "Value": "123456"
  },
  {
    "Key": "Department",
    "Value": "Accounting"
  }
]
```

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicyTags](#)」の「」を参照してください。

list-policy-versions

次の例は、list-policy-versions を使用する方法を説明しています。

AWS CLI

指定された管理ポリシーのバージョンに関する情報を一覧表示するには

この例では、ARNであるポリシーの使用可能なバージョンのリストを返しますarn:aws:iam::123456789012:policy/MySamplePolicy。

```
aws iam list-policy-versions \
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

出力:

```
{
  "IsTruncated": false,
  "Versions": [
    {
      "VersionId": "v2",
      "IsDefaultVersion": true,
      "CreateDate": "2015-06-02T23:19:44Z"
    },
    {
```

```
    "VersionId": "v1",
    "IsDefaultVersion": false,
    "CreateDate": "2015-06-02T22:30:47Z"
  }
]
```

詳細については、AWS IAM「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListPolicyVersions](#)」の「」を参照してください。

list-role-policies

次のコード例は、list-role-policies を使用する方法を示しています。

AWS CLI

IAMロールにアタッチされたポリシーを一覧表示するには

次のlist-role-policiesコマンドは、指定されたIAMロールのアクセス許可ポリシーの名前を一覧表示します。

```
aws iam list-role-policies \
  --role-name Test-Role
```

出力:

```
{
  "PolicyNames": [
    "ExamplePolicy"
  ]
}
```

ロールにアタッチされている信頼ポリシーを表示するには、get-role コマンドを使用します。アクセス許可ポリシーの詳細を表示するには、get-role-policy コマンドを使用します。

詳細については、「ユーザーガイド」のIAM「[ロールの作成](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[ListRolePolicies](#)」の「」を参照してください。

list-role-tags

次のコード例は、list-role-tags を使用する方法を示しています。

AWS CLI

ロールにアタッチされたタグを一覧表示するには

次の list-role-tags コマンドは、指定されたロールに関連付けられているタグのリストを取得します。

```
aws iam list-role-tags \  
  --role-name production-role
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    },  
    {  
      "Key": "DeptID",  
      "Value": "12345"  
    }  
  ],  
  "IsTruncated": false  
}
```

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRoleTags](#)」の「」を参照してください。

list-roles

次のコード例は、list-roles を使用する方法を示しています。

AWS CLI

現在のアカウントのIAMロールを一覧表示するには

次のlist-rolesコマンドは、現在のアカウントのIAMロールを一覧表示します。

```
aws iam list-roles
```

出力:

```
{
  "Roles": [
    {
      "Path": "/",
      "RoleName": "ExampleRole",
      "RoleId": "AR0AJ520TH4H7LEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
      "CreateDate": "2017-09-12T19:23:36+00:00",
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      },
      "MaxSessionDuration": 3600
    },
    {
      "Path": "/example_path/",
      "RoleName": "ExampleRoleWithPath",
      "RoleId": "AR0AI4QRP7UFT7EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:role/example_path/ExampleRoleWithPath",
      "CreateDate": "2023-09-21T20:29:38+00:00",
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
```

```
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ],
  "MaxSessionDuration": 3600
}
}
```

詳細については、「ユーザーガイド」のIAM「[ロールの作成](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンスListRoles](#)」の「」を参照してください。

list-saml-provider-tags

次の例は、list-saml-provider-tags を使用する方法を説明しています。

AWS CLI

SAMLプロバイダーにアタッチされたタグを一覧表示するには

次のlist-saml-provider-tagsコマンドは、指定されたSAMLプロバイダーに関連付けられているタグのリストを取得します。

```
aws iam list-saml-provider-tags \
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS
```

出力:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

```
}
```

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListSamlProviderTags](#)」の「」を参照してください。

list-saml-providers

次の例は、list-saml-providers を使用方法を説明しています。

AWS CLI

AWS アカウントのSAMLプロバイダーを一覧表示するには

この例では、現在の AWS アカウントで作成された SAML 2.0 プロバイダーのリストを取得します。

```
aws iam list-saml-providers
```

出力:

```
{
  "SAMLProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
      "ValidUntil": "2015-06-05T22:45:14Z",
      "CreateDate": "2015-06-05T22:45:14Z"
    }
  ]
}
```

詳細については、「ユーザーガイド」の [IAMSAML 「アイデンティティプロバイダーの作成」](#) を参照してください。AWS IAM

- API 詳細については、「コマンドリファレンス」の [「ListSAMLProviders」](#) を参照してください。AWS CLI

list-server-certificate-tags

次のコード例は、list-server-certificate-tags を使用方法を示しています。

AWS CLI

サーバー証明書にアタッチされたタグを一覧表示するには

次のlist-server-certificate-tagsコマンドは、指定されたサーバー証明書に関連付けられたタグのリストを取得します。

```
aws iam list-server-certificate-tags \  
  --server-certificate-name ExampleCertificate
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

詳細については、AWS IAM「ユーザーガイド」の[IAM「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListServerCertificateTags](#)」の「」を参照してください。

list-server-certificates

次のコード例は、list-server-certificates を使用する方法を示しています。

AWS CLI

AWS アカウントのサーバー証明書を一覧表示するには

次のlist-server-certificatesコマンドは、AWS アカウントに保存され、使用可能なすべてのサーバー証明書を一覧表示します。

aws iam list-server-certificates

出力:

```
{
  "ServerCertificateMetadataList": [
    {
      "Path": "/",
      "ServerCertificateName": "myUpdatedServerCertificate",
      "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:server-certificate/myUpdatedServerCertificate",
      "UploadDate": "2019-04-22T21:13:44+00:00",
      "Expiration": "2019-10-15T22:23:16+00:00"
    },
    {
      "Path": "/cloudfront/",
      "ServerCertificateName": "MyTestCert",
      "ServerCertificateId": "ASCAEXAMPLE456EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:server-certificate/Org1/Org2/MyTestCert",
      "UploadDate": "2015-04-21T18:14:16+00:00",
      "Expiration": "2018-01-14T17:52:36+00:00"
    }
  ]
}
```

詳細については、AWS IAM「ユーザーガイド」の「[でのサーバー証明書の管理IAM](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListServerCertificates](#)」の「」を参照してください。

list-service-specific-credential

次の例は、list-service-specific-credential を使用する方法を説明しています。

AWS CLI

例 1: ユーザーのサービス固有の認証情報を一覧表示する

次のlist-service-specific-credentials例では、指定されたユーザーに割り当てられたすべてのサービス固有の認証情報を表示します。パスワードはレスポンスに含まれません。

```
aws iam list-service-specific-credentials \  
  --user-name sofia
```

出力:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",  
    "ServiceName": "codecommit.amazonaws.com",  
    "ServiceUserName": "sofia-at-123456789012",  
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",  
    "UserName": "sofia",  
    "Status": "Active"  
  }  
}
```

例 2: 指定されたサービスにフィルタリングされたユーザーのサービス固有の認証情報を一覧表示する

次のlist-service-specific-credentials例では、リクエストを行うユーザーに割り当てられたサービス固有の認証情報を表示します。リストは、指定されたサービスの認証情報のみを含むようにフィルタリングされます。パスワードはレスポンスに含まれません。

```
aws iam list-service-specific-credentials \  
  --service-name codecommit.amazonaws.com
```

出力:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",  
    "ServiceName": "codecommit.amazonaws.com",  
    "ServiceUserName": "sofia-at-123456789012",  
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",  
    "UserName": "sofia",  
    "Status": "Active"  
  }  
}
```

詳細については、AWS CodeCommit ユーザーガイドの「[HTTPSへの接続用の Git 認証情報の作成 CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServiceSpecificCredential](#)」の「」を参照してください。

list-service-specific-credentials

次のコード例は、list-service-specific-credentials を使用する方法を示しています。

AWS CLI

認証情報のリストを取得するには

次のlist-service-specific-credentials例では、という名前のユーザーの AWS CodeCommit リポジトリHTTPSへのアクセス用に生成された認証情報を一覧表示します developer。

```
aws iam list-service-specific-credentials \  
  --user-name developer \  
  --service-name codecommit.amazonaws.com
```

出力:

```
{  
  "ServiceSpecificCredentials": [  
    {  
      "UserName": "developer",  
      "Status": "Inactive",  
      "ServiceUserName": "developer-at-123456789012",  
      "CreateDate": "2019-10-01T04:31:41Z",  
      "ServiceSpecificCredentialId": "ACCAQFODXMPL4YFHP7DZE",  
      "ServiceName": "codecommit.amazonaws.com"  
    },  
    {  
      "UserName": "developer",  
      "Status": "Active",  
      "ServiceUserName": "developer+1-at-123456789012",  
      "CreateDate": "2019-10-01T04:31:45Z",  
      "ServiceSpecificCredentialId": "ACCAQFOXMPML6VW57M7AJP",  
      "ServiceName": "codecommit.amazonaws.com"  
    }  
  ]  
}
```

```
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の「[HTTPSへの接続用の Git 認証情報の作成 CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServiceSpecificCredentials](#)」の「」を参照してください。

list-signing-certificates

次の例は、list-signing-certificates を使用する方法を説明しています。

AWS CLI

IAMユーザーの署名証明書を一覧表示するには

次のlist-signing-certificatesコマンドは、 という名前のIAMユーザーの署名証明書を一覧表示しますBob。

```
aws iam list-signing-certificates \  
  --user-name Bob
```

出力:

```
{  
  "Certificates": [  
    {  
      "UserName": "Bob",  
      "Status": "Inactive",  
      "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----  
END CERTIFICATE-----",  
      "CertificateId": "TA7SMP42TDN5Z260BPJE7EXAMPLE",  
      "UploadDate": "2013-06-06T21:40:08Z"  
    }  
  ]  
}
```

詳細については、「Amazon ユーザーガイド」の「[署名証明書の管理](#)」を参照してください。

EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSigningCertificates](#)」の「」を参照してください。

list-ssh-public-keys

次の例は、list-ssh-public-keys を使用する方法を説明しています。

AWS CLI

IAM ユーザーに添付されたSSHパブリックキーを一覧表示するには

次のlist-ssh-public-keys例では、IAMユーザー にアタッチされたSSHパブリックキーを一覧表示しますsofia。

```
aws iam list-ssh-public-keys \  
  --user-name sofia
```

出力:

```
{  
  "SSHPublicKeys": [  
    {  
      "UserName": "sofia",  
      "SSHPublicKeyId": "APKA1234567890EXAMPLE",  
      "Status": "Inactive",  
      "UploadDate": "2019-04-18T17:04:49+00:00"  
    }  
  ]  
}
```

詳細については、AWS IAM 「ユーザーガイド」の「[SSHキーと SSH を使用する CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSshPublicKeys](#)」の「」を参照してください。

list-user-policies

次の例は、list-user-policies を使用する方法を説明しています。

AWS CLI

IAMユーザーのポリシーを一覧表示するには

次のlist-user-policiesコマンドは、 という名前のIAMユーザーにアタッチされているポリシーを一覧表示しますBob。

```
aws iam list-user-policies \  
  --user-name Bob
```

出力:

```
{  
  "PolicyNames": [  
    "ExamplePolicy",  
    "TestPolicy"  
  ]  
}
```

詳細については、「ユーザーガイド」の[AWS「アカウントでのIAMユーザーの作成」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[ListUserPolicies](#)」の「」を参照してください。

list-user-tags

次の例は、list-user-tags を使用する方法を説明しています。

AWS CLI

ユーザーにアタッチされたタグを一覧表示するには

次のlist-user-tagsコマンドは、指定されたIAMユーザーに関連付けられたタグのリストを取得します。

```
aws iam list-user-tags \  
  --user-name alice
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    },  
    {
```

```
        "Key": "DeptID",
        "Value": "12345"
    }
  ],
  "IsTruncated": false
}
```

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUserTags](#)」の「」を参照してください。

list-users

次の例は、list-users を使用する方法を説明しています。

AWS CLI

IAM ユーザーを一覧表示するには

次のlist-usersコマンドは、現在のアカウントのIAMユーザーを一覧表示します。

```
aws iam list-users
```

出力:

```
{
  "Users": [
    {
      "UserName": "Adele",
      "Path": "/",
      "CreateDate": "2013-03-07T05:14:48Z",
      "UserId": "AKIAI44QH8DHBEXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/Adele"
    },
    {
      "UserName": "Bob",
      "Path": "/",
      "CreateDate": "2012-09-21T23:03:13Z",
      "UserId": "AKIAIOSFODNN7EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:user/Bob"
    }
  ]
}
```

```
    }  
  ]  
}
```

詳細については、AWS IAM 「ユーザーガイド」のIAM 「[ユーザーの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUsers](#)」の「」を参照してください。

list-virtual-mfa-devices

次のコード例は、list-virtual-mfa-devices を使用する方法を示しています。

AWS CLI

仮想MFAデバイスを一覧表示するには

次のlist-virtual-mfa-devicesコマンドは、現在のアカウント用に設定された仮想MFAデバイスを一覧表示します。

```
aws iam list-virtual-mfa-devices
```

出力:

```
{  
  "VirtualMFADevices": [  
    {  
      "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"  
    },  
    {  
      "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"  
    }  
  ]  
}
```

詳細については、AWS IAM ユーザーガイドの「[仮想多要素認証 \(MFA\) デバイスを有効にする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVirtualMfaDevices](#)」の「」を参照してください。

put-group-policy

次の例は、put-group-policy を使用する方法を説明しています。

AWS CLI

グループにポリシーを追加するには

次のput-group-policyコマンドは、 という名前のIAMグループにポリシーを追加しますAdmins。

```
aws iam put-group-policy \  
  --group-name Admins \  
  --policy-document file://AdminPolicy.json \  
  --policy-name AdminRoot
```

このコマンドでは何も出力されません。

ポリシーは、AdminPolicy.json ファイル内のJSONドキュメントとして定義されます。(ファイル名と拡張子には意味はありません。)

詳細については、「ユーザーガイド」の[IAM「ポリシーの管理」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[PutGroupPolicy](#)」の「」を参照してください。

put-role-permissions-boundary

次のコード例は、put-role-permissions-boundary を使用する方法を示しています。

AWS CLI

例 1: カスタムポリシーに基づいてアクセス許可の境界をIAMロールに適用するには

次のput-role-permissions-boundary例では、指定されたIAMロールのアクセス許可の境界intern-boundaryとして という名前のカスタムポリシーを適用します。

```
aws iam put-role-permissions-boundary \  
  --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
  --role-name MyRole
```

```
--role-name lambda-application-role
```

このコマンドでは何も出力されません。

例 2: AWS マネージドポリシーに基づいてアクセス許可の境界をIAMロールに適用するには

次のput-role-permissions-boundary例では、指定されたIAMロールのアクセス許可の境界として AWS マネージドPowerUserAccessポリシーを適用します。

```
aws iam put-role-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --role-name x-account-admin
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の「[ロールの変更](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスPutRolePermissionsBoundary](#)」の「」を参照してください。

put-role-policy

次の例は、put-role-policy を使用する方法を説明しています。

AWS CLI

アクセス許可ポリシーをIAMロールにアタッチするには

次の put-role-policy コマンドは、Test-Role という名前のロールにアクセス許可ポリシーを追加します。

```
aws iam put-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

このコマンドでは何も出力されません。

ポリシーは、AdminPolicy.json ファイル内のJSONドキュメントとして定義されます。(ファイル名と拡張子には意味はありません。)

信頼ポリシーをロールにアタッチするには、`update-assume-role-policy` コマンドを使用します。

詳細については、AWS IAM「ユーザーガイド」の「[ロールの変更](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[PutRolePolicy](#)」の「」を参照してください。

`put-user-permissions-boundary`

次の例は、`put-user-permissions-boundary` を使用する方法を説明しています。

AWS CLI

例 1: カスタムポリシーに基づいてアクセス許可の境界をIAMユーザーに適用するには

次の`put-user-permissions-boundary`例では、指定されたIAMユーザーのアクセス許可の境界`intern-boundary`として という名前のカスタムポリシーを適用します。

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
  --user-name intern
```

このコマンドでは何も出力されません。

例 2: AWS マネージドポリシーに基づいてアクセス許可の境界をIAMユーザーに適用するには

次の`put-user-permissions-boundary`例では、指定されたIAMユーザーのアクセス許可の境界`PowerUserAccess`として という名前の AWS マネージドポリシーを適用します。

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --user-name developer
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[IAM「ID アクセス許可の追加と削除](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[PutUserPermissionsBoundary](#)」の「」を参照してください。

put-user-policy

次のコード例は、put-user-policy を使用する方法を示しています。

AWS CLI

IAMユーザーにポリシーをアタッチするには

次のput-user-policyコマンドは、 という名前のIAMユーザーにポリシーをアタッチしますBob。

```
aws iam put-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

このコマンドでは何も出力されません。

ポリシーは、AdminPolicy.json ファイル内のJSONドキュメントとして定義されます。(ファイル名と拡張子には意味はありません。)

詳細については、「ユーザーガイド」の [IAM「ID アクセス許可の追加と削除」](#) を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス [PutUserPolicy](#)」の「」を参照してください。

remove-client-id-from-open-id-connect-provider

次のコード例は、remove-client-id-from-open-id-connect-provider を使用する方法を示しています。

AWS CLI

指定された OpenID Connect プロバイダーIDsに登録されたクライアントのリストから指定されたクライアント ID IAM を削除するには

この例では、ARNであるIAMOIDCプロバイダーIDsに関連付けられたクライアントのリストMy-TestApp-3からクライアント ID を削除しますarn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com。


```
aws iam remove-client-id-from-open-id-connect-provider
  --client-id My-TestApp-3 \
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[OpenID Connect \(OIDC\) ID プロバイダーの作成](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[RemoveClientIdFromOpenIdConnectProvider](#)」の「」を参照してください。

remove-role-from-instance-profile

次の例は、`remove-role-from-instance-profile` を使用する方法を説明しています。

AWS CLI

ロールをインスタンスプロファイルから削除するには

次の `remove-role-from-instance-profile` コマンドは、`ExampleInstanceProfile` という名前のインスタンスプロファイルから `Test-Role` という名前のロールを削除します。

```
aws iam remove-role-from-instance-profile \
  --instance-profile-name ExampleInstanceProfile \
  --role-name Test-Role
```

詳細については、「ユーザーガイド」の[「インスタンスプロファイルの使用」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI「コマンドリファレンス[RemoveRoleFromInstanceProfile](#)」の「」を参照してください。

remove-user-from-group

次のコード例は、`remove-user-from-group` を使用する方法を示しています。

AWS CLI

IAMグループからユーザーを削除するには

次のremove-user-from-groupコマンドは、 という名前のユーザーを という名前のIAMグループBobから削除しますAdmins。

```
aws iam remove-user-from-group \  
  --user-name Bob \  
  --group-name Admins
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の [「ユーザーグループ内のIAMユーザーの追加と削除」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[RemoveUserFromGroup](#)」の「」を参照してください。

reset-service-specific-credential

次のコード例は、reset-service-specific-credential を使用する方法を示しています。

AWS CLI

例 1: リクエストを行うユーザーにアタッチされたサービス固有の認証情報のパスワードをリセットする

次のreset-service-specific-credential例では、リクエストを行うユーザーにアタッチされた、指定されたサービス固有の認証情報の新しい暗号化された強力なパスワードを生成します。

```
aws iam reset-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

出力:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",  
    "ServiceName": "codecommit.amazonaws.com",  
    "ServiceUserName": "sofia-at-123456789012",  
    "ServicePassword": "+oaFsNk7tLco+C/obP9Ghhc0zGcK0ayTmE3LnAmAmH4=",  
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
```

```
    "UserName": "sofia",  
    "Status": "Active"  
  }  
}
```

例 2: 指定されたユーザーにアタッチされたサービス固有の認証情報のパスワードをリセットする

次の `reset-service-specific-credential` 例では、指定されたユーザーにアタッチされたサービス固有の認証情報の新しい暗号化された強力なパスワードを生成します。

```
aws iam reset-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

出力:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",  
    "ServiceName": "codecommit.amazonaws.com",  
    "ServiceUserName": "sofia-at-123456789012",  
    "ServicePassword": "+oaFsNk7tLco+C/obP9Ghhc0zGcK0ayTmE3LnAmAmH4=",  
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",  
    "UserName": "sofia",  
    "Status": "Active"  
  }  
}
```

詳細については、AWS CodeCommit 「ユーザーガイド」の [「HTTPSへの接続用の Git 認証情報の作成 CodeCommit」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ResetServiceSpecificCredential](#)」の「」を参照してください。

resync-mfa-device

次の例は、`resync-mfa-device` を使用する方法を説明しています。

AWS CLI

MFAデバイスを同期するには

次のresync-mfa-device例では、IAMユーザーに関連付けられBob、がであるMFAデバイスを、2つの認証コードを提供する認証プログラムARNarn:aws:iam::123456789012:mfa/BobsMFADeviceと同期します。

```
aws iam resync-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \  
  --authentication-code1 123456 \  
  --authentication-code2 987654
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の「[での多要素認証 \(MFA\) の使用 AWS](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ResyncMfaDevice](#)」の「」を参照してください。

set-default-policy-version

次のコード例は、set-default-policy-version を使用する方法を示しています。

AWS CLI

指定されたポリシーの指定されたバージョンを、ポリシーのデフォルトバージョンとして設定するには

この例では、ARN がデフォルトのアクティブv2バージョンarn:aws:iam::123456789012:policy/MyPolicyであるポリシーのバージョンを設定します。

```
aws iam set-default-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

詳細については、AWS IAM「ユーザーガイド」の「[のポリシーとアクセス許可IAM](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[SetDefaultPolicyVersion](#)」の「」を参照してください。

set-security-token-service-preferences

次の例は、set-security-token-service-preferences を使用する方法を説明しています。

AWS CLI

グローバルエンドポイントトークンバージョンを設定するには

次のset-security-token-service-preferences例では、グローバルエンドポイントに対して認証するときにバージョン 2 トークンを使用するSTSように Amazon を設定します。

```
aws iam set-security-token-service-preferences \  
  --global-endpoint-token-version v2Token
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[AWS リージョンでの管理 AWS STS](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetSecurityTokenServicePreferences](#)」の「」を参照してください。

simulate-custom-policy

次のコード例は、simulate-custom-policy を使用する方法を示しています。

AWS CLI

例 1: IAM ユーザーまたはロールに関連付けられているすべてのIAMポリシーの効果をシミュレートするには

以下に、ポリシーと変数値の両方を指定し、API呼び出しをシミュレートして、許可または拒否されているかどうかを確認する方法simulate-custom-policyを示します。次の例は、指定された日付と時刻の後にのみデータベースアクセスを有効にするポリシーを示しています。シミュレーションは成功します。シミュレーションされたアクションと指定されたaws:CurrentTime変数がすべてポリシーの要件と一致するためです。

```
aws iam simulate-custom-policy \  
  --policy-input-list '{"Version":"2012-10-17","Statement":  
{"Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":  
{"DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}' \  
  --action-names dynamodb>CreateBackup \  
  \
```

```
--context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2019-04-25T11:00:00Z',ContextKey
```

出力:

```
{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb:CreateBackup",
      "EvalResourceName": "*",
      "EvalDecision": "allowed",
      "MatchedStatements": [
        {
          "SourcePolicyId": "PolicyInputList.1",
          "StartPosition": {
            "Line": 1,
            "Column": 38
          },
          "EndPosition": {
            "Line": 1,
            "Column": 167
          }
        }
      ],
      "MissingContextValues": []
    }
  ]
}
```

例 2: ポリシーで禁止されているコマンドをシミュレートするには

次のsimulate-custom-policy例は、ポリシーで禁止されているコマンドをシミュレートした結果を示しています。この例では、指定された日付は、ポリシーの条件で必要とされる日付より前です。

```
aws iam simulate-custom-policy \
  --policy-input-list '{"Version":"2012-10-17","Statement":
{ "Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":
{ "DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}' \
  --action-names dynamodb:CreateBackup \
  --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2014-04-25T11:00:00Z',ContextKey
```

出力:

```
{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb:CreateBackup",
      "EvalResourceName": "*",
      "EvalDecision": "implicitDeny",
      "MatchedStatements": [],
      "MissingContextValues": []
    }
  ]
}
```

詳細については、「[ユーザーガイド](#)」のIAM「[ポリシーシミュレータを使用したIAMポリシーのテスト](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンス](#)[SimulateCustomPolicy](#)」の「」を参照してください。

simulate-principal-policy

次の例は、simulate-principal-policy を使用する方法を説明しています。

AWS CLI

例 1: 任意のIAMポリシーの効果をシミュレートするには

以下は、APIアクションを呼び出したユーザーをシミュレートし、そのユーザーに関連付けられたポリシーがアクションを許可または拒否するかどうかを決定する方法simulate-principal-policyを示しています。次の例では、ユーザーにcodecommit:ListRepositoriesアクションのみを許可するポリシーがあります。

```
aws iam simulate-principal-policy \
  --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
  --action-names codecommit:ListRepositories
```

出力:

```
{
```

```

    "EvaluationResults": [
      {
        "EvalActionName": "codecommit:ListRepositories",
        "EvalResourceName": "*",
        "EvalDecision": "allowed",
        "MatchedStatements": [
          {
            "SourcePolicyId": "Grant-Access-To-CodeCommit-ListRepo",
            "StartPosition": {
              "Line": 3,
              "Column": 19
            },
            "EndPosition": {
              "Line": 9,
              "Column": 10
            }
          }
        ],
        "MissingContextValues": []
      }
    ]
  }
}

```

例 2: 禁止されているコマンドの影響をシミュレートするには

次のsimulate-custom-policy例は、ユーザーのポリシーのいずれかで禁止されているコマンドをシミュレートした結果を示しています。次の例では、特定の日付と時刻の後にのみ DynamoDB データベースへのアクセスを許可するポリシーがユーザーにあります。シミュレーションでは、ポリシーの条件が許可する値よりも前のaws:CurrentTime値でデータベースにアクセスしようとしています。

```

aws iam simulate-principal-policy \
  --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
  --action-names dynamodb:CreateBackup \
  --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2018-04-25T11:00:00Z',ContextKey

```

出力:

```

{
  "EvaluationResults": [
    {

```



```
    "EvalActionName": "dynamodb:CreateBackup",
    "EvalResourceName": "*",
    "EvalDecision": "implicitDeny",
    "MatchedStatements": [],
    "MissingContextValues": []
  }
]
```

詳細については、「[ユーザーガイド](#)」のIAM「[ポリシーシミュレータを使用したIAMポリシーのテスト](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI「[コマンドリファレンスSimulatePrincipalPolicy](#)」の「」を参照してください。

tag-instance-profile

次のコード例は、tag-instance-profile を使用する方法を示しています。

AWS CLI

インスタンスプロファイルにタグを追加するには

次のtag-instance-profileコマンドは、指定されたインスタンスプロファイルに部門名のタグを追加します。

```
aws iam tag-instance-profile \
  --instance-profile-name deployment-role \
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスTagInstanceProfile](#)」の「」を参照してください。

tag-mfa-device

次のコード例は、tag-mfa-device を使用する方法を示しています。

AWS CLI

MFAデバイスにタグを追加するには

次のtag-mfa-deviceコマンドは、指定されたMFAデバイスに部門名のタグを追加します。

```
aws iam tag-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/alice \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[IAM「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TagMfaDevice](#)」の「」を参照してください。

tag-open-id-connect-provider

次の例は、tag-open-id-connect-provider を使用する方法を説明しています。

AWS CLI

OpenID Connect (OIDC) 互換 ID プロバイダーにタグを追加するには

次のtag-open-id-connect-providerコマンドは、指定された OIDC ID プロバイダーに部門名のタグを追加します。

```
aws iam tag-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[IAM「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TagOpenIdConnectProvider](#)」の「」を参照してください。

tag-policy

次の例は、tag-policy を使用する方法を説明しています。

AWS CLI

カスタマー管理ポリシーにタグを追加するには

次のtag-policyコマンドは、指定されたカスタマー管理ポリシーに部門名のタグを追加します。

```
aws iam tag-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/billing-access \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagPolicy](#)」の「」を参照してください。

tag-role

次の例は、tag-role を使用する方法を説明しています。

AWS CLI

ロールにタグを追加するには

次の tag-role コマンドは、部門名を持つタグを指定したロールに追加します。

```
aws iam tag-role --role-name my-role \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagRole](#)」の「」を参照してください。

tag-saml-provider

次のコード例は、tag-saml-provider を使用する方法を示しています。

AWS CLI

SAMLプロバイダーにタグを追加するには

次のtag-saml-providerコマンドは、指定されたSAMLプロバイダーに部門名のタグを追加します。

```
aws iam tag-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagSamlProvider](#)」の「」を参照してください。

tag-server-certificate

次のコード例は、tag-server-certificate を使用する方法を示しています。

AWS CLI

サーバー証明書にタグを追加するには

次のtag-saml-providerコマンドは、指定されたサーバー証明書に部門名のタグを追加します。

```
aws iam tag-server-certificate \  
  --server-certificate-name ExampleCertificate \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[IAM「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TagServerCertificate](#)」の「」を参照してください。

tag-user

次の例は、tag-user を使用する方法を説明しています。

AWS CLI

ユーザーにタグを追加するには

次の tag-user コマンドは、関連付けられた部門を持つタグを指定したユーザーに追加します。

```
aws iam tag-user \  
  --user-name alice \  
  --tags '{"Key": "Department", "Value": "Accounting"}'
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の[IAM「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TagUser](#)」の「」を参照してください。

untag-instance-profile

次のコード例は、untag-instance-profile を使用する方法を示しています。

AWS CLI

インスタンスプロファイルからタグを削除するには

次の untag-instance-profile コマンドは、キー名 'Department' のタグを指定されたインスタンスプロファイルから削除します。

```
aws iam untag-instance-profile \  
  --instance-profile-name InstanceProfileName \  
  --tags '{"Key": "Department", "Value": "Accounting"}'
```

```
--instance-profile-name deployment-role \  
--tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagInstanceProfile](#)」の「」を参照してください。

untag-mfa-device

次のコード例は、`untag-mfa-device` を使用する方法を示しています。

AWS CLI

MFAデバイスからタグを削除するには

次の`untag-mfa-device`コマンドは、キー名「`Department`」のタグを指定したMFAデバイスからすべて削除します。

```
aws iam untag-mfa-device \  
--serial-number arn:aws:iam::123456789012:mfa/alice \  
--tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagMfaDevice](#)」の「」を参照してください。

untag-open-id-connect-provider

次のコード例は、`untag-open-id-connect-provider` を使用する方法を示しています。

AWS CLI

OIDC ID プロバイダーからタグを削除するには

次の`untag-open-id-connect-provider`コマンドは、キー名 'Department' のタグを指定された OIDC ID プロバイダーから削除します。

```
aws iam untag-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagOpenIdConnectProvider](#)」の「」を参照してください。

untag-policy

次の例は、`untag-policy` を使用する方法を説明しています。

AWS CLI

カスタマー管理ポリシーからタグを削除するには

次の`untag-policy`コマンドは、キー名 'Department' のタグを指定されたカスタマー管理ポリシーから削除します。

```
aws iam untag-policy \  
  --policy-arn arn:aws:iam::452925170507:policy/billing-access \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagPolicy](#)」の「」を参照してください。

untag-role

次のコード例は、`untag-role` を使用する方法を示しています。

AWS CLI

ロールからタグを削除するには

次の `untag-role` コマンドは、キー名「Department」のタグを指定されたロールから削除します。

```
aws iam untag-role \  
  --role-name my-role \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の [IAM「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [UntagRole](#)」の「」を参照してください。

untag-saml-provider

次のコード例は、`untag-saml-provider` を使用方法を示しています。

AWS CLI

SAMLプロバイダーからタグを削除するには

次の `untag-saml-provider` コマンドは、キー名 'Department' のタグを指定したインスタンスプロファイルから削除します。

```
aws iam untag-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM「ユーザーガイド」の [IAM「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [UntagSamlProvider](#)」の「」を参照してください。

untag-server-certificate

次の例は、`untag-server-certificate` を使用する方法を説明しています。

AWS CLI

サーバー証明書からタグを削除するには

次の`untag-server-certificate` コマンドは、キー名 'Department' のタグを指定されたサーバー証明書から削除します。

```
aws iam untag-server-certificate \  
  --server-certificate-name ExampleCertificate \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagServerCertificate](#)」の「」を参照してください。

untag-user

次のコード例は、`untag-user` を使用する方法を示しています。

AWS CLI

ユーザーからタグを削除するには

次の `untag-user` コマンドは、キー名が「Department」のタグを指定されたユーザーから削除します。

```
aws iam untag-user \  
  --user-name alice \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「リソースのタグ付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagUser](#)」の「」を参照してください。

update-access-key

次の例は、update-access-key を使用する方法を説明しています。

AWS CLI

IAMユーザーのアクセスキーをアクティブ化または非アクティブ化するには

次のupdate-access-keyコマンドは、 という名前のIAMユーザーの指定されたアクセスキー (アクセスキー ID とシークレットアクセスキー) を非アクティブ化しますBob。

```
aws iam update-access-key \  
  --access-key-id AKIAIOSFODNN7EXAMPLE \  
  --status Inactive \  
  --user-name Bob
```

このコマンドでは何も出力されません。

キーを非アクティブ化すると、へのプログラムによるアクセスには使用できません AWS。ただし、キーは引き続き使用可能であり、再アクティブ化することができます。

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「ユーザーのアクセスキーの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAccessKey](#)」の「」を参照してください。

update-account-password-policy

次のコード例は、update-account-password-policy を使用する方法を示しています。

AWS CLI

現在のアカウントのパスワードポリシーを設定または変更するには

次の update-account-password-policy コマンドは、8 文字以上の長さを要求し、パスワードに 1 つ以上の数字を要求するようにパスワードポリシーを設定します。

```
aws iam update-account-password-policy \  
  --minimum-password-length 8 \  
  --require-numbers
```

このコマンドでは何も出力されません。

アカウントのパスワードポリシーを変更すると、アカウント内のIAMユーザー用に作成された新しいパスワードに影響します。パスワードポリシーを変更しても、既存のパスワードには影響しません。

詳細については、AWS IAM「ユーザーガイド」の[IAM「ユーザーのアカウントパスワードポリシーの設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateAccountPasswordPolicy](#)」の「」を参照してください。

update-assume-role-policy

次の例は、update-assume-role-policy を使用する方法を説明しています。

AWS CLI

IAMロールの信頼ポリシーを更新するには

次の update-assume-role-policy コマンドは、Test-Role という名前のロールの信頼ポリシーを更新します。

```
aws iam update-assume-role-policy \  
  --role-name Test-Role \  
  --policy-document file:///Test-Role-Trust-Policy.json
```

このコマンドでは何も出力されません。

信頼ポリシーは、Test-Role-Trust-Policy.json ファイル内のJSONドキュメントとして定義されます。(ファイル名と拡張子には意味はありません。) 信頼ポリシーはプリンシパルを指定する必要があります。

ロールのアクセス許可ポリシーを更新するには、put-role-policy コマンドを使用します。

詳細については、「ユーザーガイド」の[IAM「ロールの作成」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAssumeRolePolicy](#)」の「」を参照してください。

update-group

次のコード例は、update-group を使用する方法を示しています。

AWS CLI

IAMグループの名前を変更するには

次のupdate-groupコマンドは、IAMグループの名前Testを に変更しますTest-1。

```
aws iam update-group \  
  --group-name Test \  
  --new-group-name Test-1
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」のIAM「[ユーザーグループの名前変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateGroup](#)」の「」を参照してください。

update-login-profile

次の例は、update-login-profile を使用する方法を説明しています。

AWS CLI

IAMユーザーのパスワードを更新するには

次のupdate-login-profileコマンドは、 という名前のIAMユーザーの新しいパスワードを作成しますBob。

```
aws iam update-login-profile \  
  --user-name Bob \  
  --password <password>
```

このコマンドでは何も出力されません。

アカウントのパスワードポリシーを設定するには、`update-account-password-policy` コマンドを使用します。新しいパスワードがアカウントのパスワードポリシーに違反している場合、コマンドは `PasswordPolicyViolation` エラーを返します。

アカウントパスワードポリシーで許可されている場合、IAMユーザーは `change-password` コマンドを使用して独自のパスワードを変更できます。

パスワードを安全な場所に保存します。パスワードを紛失した場合は回復できないため、`create-login-profile` コマンドを使用して新しいパスワードを作成する必要があります。

詳細については、「[ユーザーガイド](#)」のIAM「[ユーザーのパスワードの管理](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateLoginProfile](#)」の「」を参照してください。

update-open-id-connect-provider-thumbprint

次のコード例は、`update-open-id-connect-provider-thumbprint` を使用する方法を示しています。

AWS CLI

サーバー証明書のサムプリントの既存のリストを新しいリストに置き換えるには

この例では、ARN が新しいサムプリント `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com` を使用するOIDCプロバイダーの証明書サムプリントリストを更新します。

```
aws iam update-open-id-connect-provider-thumbprint \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
example.oidcprovider.com \  
  --thumbprint-list 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「[ユーザーガイド](#)」の[OpenID Connect \(OIDC\) ID プロバイダーの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateOpenIdConnectProviderThumbprint](#)」の「」を参照してください。

update-role-description

次の例は、update-role-description を使用する方法を説明しています。

AWS CLI

IAMロールの説明を変更するには

次のupdate-roleコマンドは、IAMロールの説明production-roleを に変更しますMain production role。

```
aws iam update-role-description \  
  --role-name production-role \  
  --description 'Main production role'
```

出力:

```
{  
  "Role": {  
    "Path": "/",  
    "RoleName": "production-role",  
    "RoleId": "AROA1234567890EXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:role/production-role",  
    "CreateDate": "2017-12-06T17:16:37+00:00",  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Effect": "Allow",  
          "Principal": {  
            "AWS": "arn:aws:iam::123456789012:root"  
          },  
          "Action": "sts:AssumeRole",  
          "Condition": {}  
        }  
      ]  
    },  
    "Description": "Main production role"  
  }  
}
```

詳細については、AWS IAM [「ユーザーガイド」](#)の[「ロールの変更」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRoleDescription](#)」の「」を参照してください。

update-role

次のコード例は、update-role を使用する方法を示しています。

AWS CLI

IAMロールの説明またはセッション期間を変更するには

次のupdate-roleコマンドは、IAMロールの説明を production-role に変更Main production roleし、最大セッション時間を 12 時間に設定します。

```
aws iam update-role \  
  --role-name production-role \  
  --description 'Main production role' \  
  --max-session-duration 43200
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[ロールの変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRole](#)」の「」を参照してください。

update-saml-provider

次の例は、update-saml-provider を使用する方法を説明しています。

AWS CLI

既存のSAMLプロバイダーのメタデータドキュメントを更新するには

この例では、ARN IAMがであるSAMLプロバイダーを、ファイルからの新しいSAMLメタデータドキュメントarn:aws:iam::123456789012:saml-provider/SAMLADFSで更新しますSAMLMetaData.xml。

```
aws iam update-saml-provider \  
  --saml-metadata-document file://SAMLMetaData.xml \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

出力:

```
{
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"
}
```

詳細については、AWS IAM 「ユーザーガイド」の[IAM SAML 「ID プロバイダーの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSamlProvider](#)」の「」を参照してください。

update-server-certificate

次のコード例は、update-server-certificate を使用する方法を示しています。

AWS CLI

AWS アカウントのサーバー証明書のパスまたは名前を変更するには

次の update-server-certificate コマンドは、証明書の名前を myServerCertificate から myUpdatedServerCertificate に変更します。また、Amazon CloudFront サービスがアクセス/cloudfront/できるようにパスを に変更します。このコマンドでは何も出力されません。list-server-certificates コマンドを実行すると、更新の結果を表示できます。

```
aws-iam update-server-certificate \
  --server-certificate-name myServerCertificate \
  --new-server-certificate-name myUpdatedServerCertificate \
  --new-path /cloudfront/
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の「[でのサーバー証明書の管理IAM](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateServerCertificate](#)」の「」を参照してください。

update-service-specific-credential

次のコード例は、update-service-specific-credential を使用する方法を示しています。

AWS CLI

例 1: リクエストするユーザーのサービス固有の認証情報のステータスを更新するには

次のupdate-service-specific-credential例では、リクエストを行うユーザーの指定された認証情報のステータスを `Inactive` に変更します。

```
aws iam update-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

このコマンドでは何も出力されません。

例 2: 指定されたユーザーのサービス固有の認証情報のステータスを更新するには

次のupdate-service-specific-credential例では、指定されたユーザーの認証情報のステータスを非アクティブに変更します。

```
aws iam update-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

このコマンドでは何も出力されません。

詳細については、AWS CodeCommit ユーザーガイドの[HTTPS「への接続用の Git 認証情報の作成 CodeCommit」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateServiceSpecificCredential](#)」の「」を参照してください。

update-signing-certificate

次のコード例は、update-signing-certificate を使用する方法を示しています。

AWS CLI

IAMユーザーの署名証明書をアクティブ化または非アクティブ化するには

次のupdate-signing-certificateコマンドは、`Bob` という名前のIAMユーザーの指定された署名証明書を非アクティブ化します。

```
aws iam update-signing-certificate \  
  --certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE \  
  --status Inactive \  
  --user-name Bob
```

署名証明書の ID を取得するには、`list-signing-certificates` コマンドを使用します。

詳細については、「Amazon ユーザーガイド」の [「署名証明書の管理」](#) を参照してください。
EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateSigningCertificate](#)」の「」を参照してください。

update-ssh-public-key

次の例は、`update-ssh-public-key` を使用する方法を説明しています。

AWS CLI

SSHパブリックキーのステータスを変更するには

次の`update-ssh-public-key`コマンドは、指定されたパブリックキーのステータスを `Inactive` に変更します。

```
aws iam update-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA1234567890EXAMPLE \  
  --status Inactive
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の [「SSHキーと SSH を使用する CodeCommit」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateSshPublicKey](#)」の「」を参照してください。

update-user

次の例は、`update-user` を使用する方法を説明しています。

AWS CLI

IAMユーザー名を変更するには

次のupdate-userコマンドは、IAMユーザーの名前を Bob に変更しますRobert。

```
aws iam update-user \  
  --user-name Bob \  
  --new-user-name Robert
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[IAM「ユーザーグループの名前変更」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUser](#)」の「」を参照してください。

upload-server-certificate

次の例は、upload-server-certificate を使用する方法を説明しています。

AWS CLI

サーバー証明書を AWS アカウントにアップロードするには

次のupload-server-certificateコマンドは、サーバー証明書を AWS アカウントにアップロードします。この例では、証明書はファイル public_key_cert_file.pem 内に、関連付けられたプライベートキーはファイル my_private_key.pem 内に、認証機関 (CA) によって提供される証明書チェーンは my_certificate_chain_file.pem ファイル内に、それぞれ存在しています。ファイルのアップロードが完了すると、という名前で使用できますmyServerCertificate。file:// で始まるパラメータは、ファイルの内容を読み取り、それをファイル名自体の代わりにパラメータ値として使用するようコマンドに指示します。

```
aws iam upload-server-certificate \  
  --server-certificate-name myServerCertificate \  
  --certificate-body file://public_key_cert_file.pem \  
  --private-key file://my_private_key.pem \  
  --certificate-chain file://my_certificate_chain_file.pem
```

出力:

```
{
  "ServerCertificateMetadata": {
    "Path": "/",
    "ServerCertificateName": "myServerCertificate",
    "ServerCertificateId": "ASCAEXAMPLE123EXAMPLE",
    "Arn": "arn:aws:iam::1234567989012:server-certificate/myServerCertificate",
    "UploadDate": "2019-04-22T21:13:44+00:00",
    "Expiration": "2019-10-15T22:23:16+00:00"
  }
}
```

詳細については、「IAMの使用ガイド」の「サーバー証明書の作成、アップロード、削除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadServerCertificate](#)」の「」を参照してください。

upload-signing-certificate

次のコード例は、upload-signing-certificate を使用する方法を示しています。

AWS CLI

IAMユーザーの署名証明書をアップロードするには

次のupload-signing-certificateコマンドは、 という名前のIAMユーザーの署名証明書をアップロードしますBob。

```
aws iam upload-signing-certificate \
  --user-name Bob \
  --certificate-body file://certificate.pem
```

出力:

```
{
  "Certificate": {
    "UserName": "Bob",
    "Status": "Active",
    "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END
CERTIFICATE-----",
    "CertificateId": "TA7SMP42TDN5Z2260BPJE7EXAMPLE",
    "UploadDate": "2013-06-06T21:40:08.121Z"
  }
}
```

```
}  
}
```

証明書は、certificate.pem という PEM 形式のファイルです。

詳細については、「IAM の使用ガイド」の「ユーザー署名証明書の作成とアップロード」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UploadSigningCertificate](#)」の「」を参照してください。

upload-ssh-public-key

次の例は、upload-ssh-public-key を使用する方法を説明しています。

AWS CLI

SSH パブリックキーをアップロードしてユーザーに関連付けるには

次の upload-ssh-public-key コマンドは、ファイルにあるパブリックキーをアップロードして sshkey.pub し、ユーザー にアタッチします sofia。

```
aws iam upload-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-body file://sshkey.pub
```

出力:

```
{  
  "SSHPublicKey": {  
    "UserName": "sofia",  
    "SSHPublicKeyId": "APKA1234567890EXAMPLE",  
    "Fingerprint": "12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef",  
    "SSHPublicKeyBody": "ssh-rsa <<long string generated by ssh-keygen  
command>>",  
    "Status": "Active",  
    "UploadDate": "2019-04-18T17:04:49+00:00"  
  }  
}
```

このコマンドに適した形式でキーを生成する方法の詳細については、AWS CodeCommit 「ユーザーガイド」の [SSH 「および Linux、macOS、または Unix: Git および または Windows のパブ](#)

[リックキーとプライベートキーをセットアップ CodeCommitSSHする: Git および のパブリックキーとプライベートキーをセットアップする CodeCommit](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadSshPublicKey](#)」の「」を参照してください。

IAM を使用した Access Analyzer の例 AWS CLI

次のコード例は、IAM Access Analyzer AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

apply-archive-rule

次のコード例は、`apply-archive-rule` を使用する方法を示しています。

AWS CLI

アーカイブルールの基準を満たす既存の検出結果にアーカイブルールを適用するには

次の`apply-archive-rule`例では、アーカイブルールの基準を満たす既存の検出結果にアーカイブルールを適用します。

```
aws accessanalyzer apply-archive-rule \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[アーカイブルール](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[ApplyArchiveRule](#)」の「」を参照してください。

cancel-policy-generation

次の例は、cancel-policy-generation を使用する方法を説明しています。

AWS CLI

リクエストされたポリシー生成をキャンセルするには

次のcancel-policy-generation例では、リクエストされたポリシー生成ジョブ ID をキャンセルします。

```
aws accessanalyzer cancel-policy-generation \  
  --job-id 923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」のIAM 「[Access Analyzer ポリシーの生成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelPolicyGeneration](#)」の「」を参照してください。

check-access-not-granted

次の例は、check-access-not-granted を使用する方法を説明しています。

AWS CLI

指定されたアクセスがポリシーで許可されていないかどうかを確認するには

次のcheck-access-not-granted例では、指定されたアクセスがポリシーで許可されていないかどうかを確認します。

```
aws accessanalyzer check-access-not-granted \  
  --policy-document file://myfile.json \  
  --access actions="s3:DeleteBucket","s3:GetBucketLocation" \  
  --policy-type IDENTITY_POLICY
```

myfile.json の内容:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ]
    }
  ]
}
```

出力:

```
{
  "result": "PASS",
  "message": "The policy document does not grant access to perform one or more of
the listed actions."
}
```

詳細については、AWS IAM「ユーザーガイド」のIAM「[Access Analyzer によるアクセスのプレビューAPIs](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CheckAccessNotGranted](#)」の「」を参照してください。

check-no-new-access

次の例は、check-no-new-access を使用する方法を説明しています。

AWS CLI

既存のポリシーと比較して、更新されたポリシーに新しいアクセスが許可されているかどうかを確認するには

次のcheck-no-new-access例では、既存のポリシーと比較して、更新されたポリシーに新しいアクセスが許可されているかどうかを確認します。

```
aws accessanalyzer check-no-new-access \  
  --existing-policy-document file://existing-policy.json \  
  --new-policy-document file://new-policy.json \  
  --policy-type IDENTITY_POLICY
```

existing-policy.json の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject",  
        "s3:ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ]  
    }  
  ]  
}
```

new-policy.json の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject",  
        "s3:GetObjectAcl",  
        "s3:ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ]  
    }  
  ]  
}
```

```

    ]
  }
]
}

```

出力:

```

{
  "result": "FAIL",
  "message": "The modified permissions grant new access compared to your existing policy.",
  "reasons": [
    {
      "description": "New access in the statement with index: 0.",
      "statementIndex": 0
    }
  ]
}

```

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[Access AnalyzerによるアクセスのプレビューAPIs](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスCheckNoNewAccess](#)」の「」を参照してください。

check-no-public-access

次のコード例は、check-no-public-access を使用する方法を示しています。

AWS CLI

リソースポリシーが指定されたリソースタイプへのパブリックアクセスを許可できるかどうかを確認するには

次のcheck-no-public-access例では、リソースポリシーが指定されたリソースタイプへのパブリックアクセスを許可できるかどうかをチェックします。

```

aws accessanalyzer check-no-public-access \
  --policy-document file://check-no-public-access-myfile.json \
  --resource-type AWS::S3::Bucket

```

myfile.json の内容:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CheckNoPublicAccess",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:user/JohnDoe" },
      "Action": [
        "s3:GetObject"
      ]
    }
  ]
}
```

出力:

```
{
  "result": "PASS",
  "message": "The resource policy does not grant public access for the given
resource type."
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[Access Analyzer によるアクセスのプレビューAPIs](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスCheckNoPublicAccess](#)」の「」を参照してください。

create-access-preview

次の例は、create-access-preview を使用する方法を説明しています。

AWS CLI

リソースアクセス許可をデプロイする前に、リソースの IAM Access Analyzer の検出結果をプレビューできるアクセスプレビューを作成するには

次のcreate-access-preview例では、AWS アカウントにリソースアクセス許可をデプロイする前に、リソースの IAM Access Analyzer の検出結果をプレビューできるアクセスプレビューを作成します。

```
aws accessanalyzer create-access-preview \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \
  --configurations file://myfile.json
```

myfile.json の内容:

```
{
  "arn:aws:s3:::DOC-EXAMPLE-BUCKET": {
    "s3Bucket": {
      "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":"
      "\": \"Allow\", \"Principal\": {\"AWS\": [\"arn:aws:iam::111122223333:root\"]}, \"Action\":"
      "\": [\"s3:PutObject\", \"s3:PutObjectAcl\"], \"Resource\": \"arn:aws:s3:::DOC-EXAMPLE-\"
      \"BUCKET/*\"}]}",
      "bucketPublicAccessBlock": {
        "ignorePublicAcls": true,
        "restrictPublicBuckets": true
      },
      "bucketAclGrants": [
        {
          "grantee": {
            "id":
"79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
          },
          "permission": "READ"
        }
      ]
    }
  }
}
```

出力:

```
{
  "id": "3c65eb13-6ef9-4629-8919-a32043619e6b"
}
```

詳細については、AWS IAM「ユーザーガイド」のIAM「[Access Analyzer によるアクセスのプレビューAPIs](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateAccessPreview](#)」の「」を参照してください。

create-analyzer

次の例は、create-analyzer を使用する方法を説明しています。

AWS CLI

アナライザーを作成するには

次のcreate-analyzer例では、AWS アカウントにアナライザーを作成します。

```
aws accessanalyzer create-analyzer \  
  --analyzer-name example \  
  --type ACCOUNT
```

出力:

```
{  
  "arn": "arn:aws:access-analyzer:us-east-2:111122223333:analyzer/example"  
}
```

詳細については、AWS IAM 「[ユーザーガイド](#)」の [AWS 「Identity and Access Management Access Analyzer の検出結果の開始方法](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateAnalyzer](#)」の「」を参照してください。

create-archive-rule

次のコード例は、create-archive-rule を使用する方法を示しています。

AWS CLI

指定されたアナライザーのアーカイブルールを作成するには

次のcreate-archive-rule例では、AWS アカウントで指定されたアナライザーのアーカイブルールを作成します。

```
aws accessanalyzer create-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyRule \  
  --
```

```
--filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq": ["AWS::IAM::Role"]}}'
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[アーカイブルール](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateArchiveRule](#)」の「」を参照してください。

delete-analyzer

次の例は、delete-analyzer を使用する方法を説明しています。

AWS CLI

指定されたアナライザーを削除するには

次のdelete-analyzer例では、AWS アカウントの指定されたアナライザーを削除します。

```
aws accessanalyzer delete-analyzer \  
  --analyzer-name example
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[アーカイブルール](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAnalyzer](#)」の「」を参照してください。

delete-archive-rule

次のコード例は、delete-archive-rule を使用する方法を示しています。

AWS CLI

指定されたアーカイブルールを削除するには

次のdelete-archive-rule例では、AWS アカウントの指定されたアーカイブルールを削除します。

```
aws accessanalyzer delete-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyRule
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の「[アーカイブルール](#)」を参照してください。AWS IAM
• API 詳細については、AWS CLI「コマンドリファレンス[DeleteArchiveRule](#)」の「」を参照してください。

get-access-preview

次のコード例は、get-access-preview を使用する方法を示しています。

AWS CLI

指定されたアナライザーのアクセスプレビューに関する情報を取得するには

次のget-access-preview例では、AWS アカウント内の指定されたアナライザーのアクセスプレビューに関する情報を取得します。

```
aws accessanalyzer get-access-preview \  
  --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account
```

出力:

```
{  
  "accessPreview": {  
    "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",  
    "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account",  
    "configurations": {  
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET": {  
        "s3Bucket": {  
          "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":  
[{\n\"Effect\": \"Allow\", \"Principal\": { \"AWS\": [ \"arn:aws:iam::111122223333:root\" ] },\n\"Action\": [ \"s3:PutObject\", \"s3:PutObjectAcl\" ], \"Resource\": \"arn:aws:s3:::DOC-  
EXAMPLE-BUCKET/*\" } ] }",  
        }  
      }  
    }  
  }  
}
```

```
        "bucketAclGrants": [
            {
                "permission": "READ",
                "grantee": {
                    "id":
"79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
                }
            }
        ],
        "bucketPublicAccessBlock": {
            "ignorePublicAcls": true,
            "restrictPublicBuckets": true
        }
    }
},
"createdAt": "2024-02-17T00:18:44+00:00",
"status": "COMPLETED"
}
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[Access AnalyzerによるアクセスのプレビューAPIs](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetAccessPreview](#)」の「」を参照してください。

get-analyzed-resource

次のコード例は、get-analyzed-resource を使用する方法を示しています。

AWS CLI

分析されたリソースに関する情報を取得するには

次のget-analyzed-resource例では、AWS アカウントで分析されたリソースに関する情報を取得します。

```
aws accessanalyzer get-analyzed-resource \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ConsoleAnalyzer-account \
  --resource-arn arn:aws:s3:::DOC-EXAMPLE-BUCKET
```


出力:

```
{
  "resource": {
    "analyzedAt": "2024-02-15T18:01:53.002000+00:00",
    "isPublic": false,
    "resourceArn": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
    "resourceOwnerAccount": "111122223333",
    "resourceType": "AWS::S3::Bucket"
  }
}
```

詳細については、AWS IAM「ユーザーガイド」の[AWS「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetAnalyzedResource](#)」の「」を参照してください。

get-analyzer

次の例は、get-analyzer を使用する方法を説明しています。

AWS CLI

指定されたアナライザーに関する情報を取得するには

次のget-analyzer例では、AWS アカウント内の指定されたアナライザーに関する情報を取得します。

```
aws accessanalyzer get-analyzer \  
  --analyzer-name ConsoleAnalyzer-account
```

出力:

```
{
  "analyzer": {
    "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account",
    "createdAt": "2019-12-03T07:28:17+00:00",
    "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-topic",
    "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
```

```
    "name": "ConsoleAnalyzer-account",
    "status": "ACTIVE",
    "tags": {
      "auto-delete": "no"
    },
    "type": "ACCOUNT"
  }
}
```

詳細については、AWS IAM「ユーザーガイド」の[AWS「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetAnalyzer](#)」の「」を参照してください。

get-archive-rule

次のコード例は、get-archive-rule を使用する方法を示しています。

AWS CLI

アーカイブルールに関する情報を取得するには

次のget-archive-rule例では、AWS アカウントのアーカイブルールに関する情報を取得します。

```
aws accessanalyzer get-archive-rule \
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
  --rule-name MyArchiveRule
```

出力:

```
{
  "archiveRule": {
    "createdAt": "2024-02-15T00:49:27+00:00",
    "filter": {
      "resource": {
        "contains": [
          "Cognito"
        ]
      }
    }
  },
}
```

```
    "resourceType": {
      "eq": [
        "AWS::IAM::Role"
      ]
    },
    "ruleName": "MyArchiveRule",
    "updatedAt": "2024-02-15T00:49:27+00:00"
  }
}
```

詳細については、「ユーザーガイド」の「[アーカイブルール](#)」を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[GetArchiveRule](#)」の「」を参照してください。

get-finding-v2

次の例は、get-finding-v2 を使用する方法を説明しています。

AWS CLI

指定された検出結果に関する情報を取得するには

次のget-finding-v2例では、AWS アカウントで指定された検出結果に関する情報を取得します。

```
aws accessanalyzer get-finding-v2 \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ConsoleAnalyzer-organization \
  --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

出力:

```
{
  "findingDetails": [
    {
      "externalAccessDetails": {
        "action": [
          "sts:AssumeRoleWithWebIdentity"
        ],
        "condition": {
```

```
        "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
      },
      "isPublic": false,
      "principal": {
        "Federated": "cognito-identity.amazonaws.com"
      }
    }
  ],
  "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
  "status": "ACTIVE",
  "error": null,
  "createdAt": "2021-02-26T21:17:50.905000+00:00",
  "resourceType": "AWS::IAM::Role",
  "findingType": "ExternalAccess",
  "resourceOwnerAccount": "111122223333",
  "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
  "id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
  "updatedAt": "2021-02-26T21:17:50.905000+00:00"
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」の「[検出結果の確認](#)」を参照してください。

- API 詳細については、「[コマンドリファレンス](#)」の[GetFindingV2](#)を参照してください。

AWS CLI

get-finding

次の例は、get-finding を使用する方法を説明しています。

AWS CLI

指定された検出結果に関する情報を取得するには

次のget-finding例では、AWS アカウントで指定された検出結果に関する情報を取得します。

```
aws accessanalyzer get-finding \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
  --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

出力:

```
{
  "finding": {
    "id": "0910eedb-381e-4e95-adda-0d25c19e6e90",
    "principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "action": [
      "sts:AssumeRoleWithWebIdentity"
    ],
    "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
    "isPublic": false,
    "resourceType": "AWS::IAM::Role",
    "condition": {
      "cognito-identity.amazonaws.com:aud": "us-west-2:EXAMPLE0-0000-0000-0000-000000000000"
    },
    "createdAt": "2021-02-26T21:17:50.905000+00:00",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "updatedAt": "2021-02-26T21:17:50.905000+00:00",
    "status": "ACTIVE",
    "resourceOwnerAccount": "111122223333"
  }
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」の「[検出結果の確認](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetFinding](#)」の「」を参照してください。

get-generated-policy

次の例は、get-generated-policy を使用する方法を説明しています。

AWS CLI

StartPolicyGeneration` を使用して生成されたポリシーを取得するには API

次のget-generated-policy例では、AWS アカウントの StartPolicyGeneration APIを使用して生成されたポリシーを取得します。

```
aws accessanalyzer get-generated-policy \
  --job-id c557dc4a-0338-4489-95dd-739014860ff9
```

出力:

```
{
  "generatedPolicyResult": {
    "generatedPolicies": [
      {
        "policy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\n\"Sid\":\n\"SupportedServiceSid0\", \"Effect\": \"Allow\", \"Action\":
[\"access-analyzer:GetAnalyzer\", \"access-analyzer:ListAnalyzers\",
\n\"access-analyzer:ListArchiveRules\", \"access-analyzer:ListFindings
\", \"cloudtrail:DescribeTrails\", \"cloudtrail:GetEventDataStore\",
\n\"cloudtrail:GetEventSelectors\", \"cloudtrail:GetInsightSelectors
\", \"cloudtrail:GetTrailStatus\", \"cloudtrail:ListChannels\",
\n\"cloudtrail:ListEventDataStores\", \"cloudtrail:ListQueries\", \"cloudtrail:ListTags
\", \"cloudtrail:LookupEvents\", \"ec2:DescribeRegions\", \"iam:GetAccountSummary
\", \"iam:GetOpenIDConnectProvider\", \"iam:GetRole\", \"iam:ListAccessKeys\",
\n\"iam:ListAccountAliases\", \"iam:ListOpenIDConnectProviders\", \"iam:ListRoles
\", \"iam:ListSAMLProviders\", \"kms:ListAliases\", \"s3:GetBucketLocation\",
\n\"s3:ListAllMyBuckets\"], \"Resource\": \"*\"]}]"
      }
    ],
    "properties": {
      "cloudTrailProperties": {
        "endTime": "2024-02-14T22:44:40+00:00",
        "startTime": "2024-02-13T00:30:00+00:00",
        "trailProperties": [
          {
            "allRegions": true,
            "cloudTrailArn": "arn:aws:cloudtrail:us-
west-2:111122223333:trail/my-trail",
            "regions": []
          }
        ]
      },
      "isComplete": false,
      "principalArn": "arn:aws:iam::111122223333:role/Admin"
    }
  },
  "jobDetails": {
    "completedOn": "2024-02-14T22:47:01+00:00",
    "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
    "startedOn": "2024-02-14T22:44:41+00:00",
    "status": "SUCCEEDED"
  }
}
```

```
}
```

詳細については、AWS IAM「ユーザーガイド」のIAM「[Access Analyzer ポリシーの生成](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetGeneratedPolicy](#)」の「」を参照してください。

list-access-preview-findings

次の例は、list-access-preview-findings を使用する方法を説明しています。

AWS CLI

指定されたアクセスプレビューによって生成されたアクセスプレビュー結果のリストを取得するには

次のlist-access-preview-findings例では、AWS アカウントで指定されたアクセスプレビューによって生成されたアクセスプレビューの検出結果のリストを取得します。

```
aws accessanalyzer list-access-preview-findings \  
  --access-preview-id 3c65eb13-6ef9-4629-8919-a32043619e6b \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account
```

出力:

```
{  
  "findings": [  
    {  
      "id": "e22fc158-1c87-4c32-9464-e7f405ce8d74",  
      "principal": {  
        "AWS": "111122223333"  
      },  
      "action": [  
        "s3:PutObject",  
        "s3:PutObjectAcl"  
      ],  
      "condition": {},  
      "resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
      "isPublic": false,  
      "resourceType": "AWS::S3::Bucket",
```

```
    "createdAt": "2024-02-17T00:18:46+00:00",
    "changeType": "NEW",
    "status": "ACTIVE",
    "resourceOwnerAccount": "111122223333",
    "sources": [
      {
        "type": "POLICY"
      }
    ]
  }
]
```

詳細については、AWS IAM「[ユーザーガイド](#)」のIAM「[Access AnalyzerによるアクセスのプレビューAPIs](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスListAccessPreviewFindings](#)」の「」を参照してください。

list-access-previews

次の例は、list-access-previews を使用する方法を説明しています。

AWS CLI

指定されたアナライザーのアクセスプレビューのリストを取得するには

次のlist-access-previews例では、AWS アカウント内の指定されたアナライザーのアクセスプレビューのリストを取得します。

```
aws accessanalyzer list-access-previews \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

出力:

```
{
  "accessPreviews": [
    {
      "id": "3c65eb13-6ef9-4629-8919-a32043619e6b",
      "analyzerArn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
```



```
        "createdAt": "2024-02-17T00:18:44+00:00",
        "status": "COMPLETED"
    }
  ]
}
```

詳細については、AWS IAM「ユーザーガイド」のIAM「[Access Analyzer によるアクセスのプレビューAPIs](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListAccessPreviews](#)」の「」を参照してください。

list-analyzed-resources

次の例は、list-analyzed-resources を使用する方法を説明しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のlist-analyzed-resources例では、AWS アカウントで使用可能なウィジェットを一覧表示します。

```
aws accessanalyzer list-analyzed-resources \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
  --resource-type AWS::IAM::Role
```

出力:

```
{
  "analyzedResources": [
    {
      "resourceArn": "arn:aws:sns:us-west-2:111122223333:Validation-Email",
      "resourceOwnerAccount": "111122223333",
      "resourceType": "AWS::SNS::Topic"
    },
    {
      "resourceArn": "arn:aws:sns:us-west-2:111122223333:admin-alerts",
      "resourceOwnerAccount": "111122223333",
      "resourceType": "AWS::SNS::Topic"
    }
  ],
}
```

```
{
  "resourceArn": "arn:aws:sns:us-west-2:111122223333:config-topic",
  "resourceOwnerAccount": "111122223333",
  "resourceType": "AWS::SNS::Topic"
},
{
  "resourceArn": "arn:aws:sns:us-west-2:111122223333:inspector-topic",
  "resourceOwnerAccount": "111122223333",
  "resourceType": "AWS::SNS::Topic"
}
]
```

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAnalyzedResources](#)」の「」を参照してください。

list-analyzers

次の例は、list-analyzers を使用する方法を説明しています。

AWS CLI

アナライザーのリストを取得するには

次のlist-analyzers例では、AWS アカウント内のアナライザーのリストを取得します。

```
aws accessanalyzer list-analyzers
```

出力:

```
{
  "analyzers": [
    {
      "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/UnusedAccess-ConsoleAnalyzer-organization",
      "createdAt": "2024-02-15T00:46:40+00:00",
      "name": "UnusedAccess-ConsoleAnalyzer-organization",
      "status": "ACTIVE",
      "tags": {
```

```
        "auto-delete": "no"
      },
      "type": "ORGANIZATION_UNUSED_ACCESS"
    },
    {
      "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization",
      "createdAt": "2020-04-25T07:43:28+00:00",
      "lastResourceAnalyzed": "arn:aws:s3::DOC-EXAMPLE-BUCKET",
      "lastResourceAnalyzedAt": "2024-02-15T21:51:56.517000+00:00",
      "name": "ConsoleAnalyzer-organization",
      "status": "ACTIVE",
      "tags": {
        "auto-delete": "no"
      },
      "type": "ORGANIZATION"
    },
    {
      "arn": "arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account",
      "createdAt": "2019-12-03T07:28:17+00:00",
      "lastResourceAnalyzed": "arn:aws:sns:us-west-2:111122223333:config-
topic",
      "lastResourceAnalyzedAt": "2024-02-15T18:01:53.003000+00:00",
      "name": "ConsoleAnalyzer-account",
      "status": "ACTIVE",
      "tags": {
        "auto-delete": "no"
      },
      "type": "ACCOUNT"
    }
  ]
}
```

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAnalyzers](#)」の「」を参照してください。

list-archive-rules

次の例は、list-archive-rules を使用する方法を説明しています。

AWS CLI

指定されたアナライザー用に作成されたアーカイブルールのリストを取得するには

次のlist-archive-rules例では、AWS アカウント内の指定されたアナライザー用に作成されたアーカイブルールのリストを取得します。

```
aws accessanalyzer list-archive-rules \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization
```

出力:

```
{  
  "archiveRules": [  
    {  
      "createdAt": "2024-02-15T00:49:27+00:00",  
      "filter": {  
        "resource": {  
          "contains": [  
            "Cognito"  
          ]  
        },  
        "resourceType": {  
          "eq": [  
            "AWS::IAM::Role"  
          ]  
        }  
      },  
      "ruleName": "MyArchiveRule",  
      "updatedAt": "2024-02-15T00:49:27+00:00"  
    },  
    {  
      "createdAt": "2024-02-15T23:27:45+00:00",  
      "filter": {  
        "findingType": {  
          "eq": [  
            "UnusedIAMUserAccessKey"  
          ]  
        }  
      },  
      "ruleName": "ArchiveRule-56125a39-e517-4ff8-afb1-ef06f58db612",  
      "updatedAt": "2024-02-15T23:27:45+00:00"  
    }  
  ]  
}
```

```
]
}
```

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListArchiveRules](#)」の「」を参照してください。

list-findings-v2

次のコード例は、list-findings-v2 を使用する方法を示しています。

AWS CLI

指定されたアナライザーによって生成された検出結果のリストを取得するには

次のlist-findings-v2例では、AWS アカウント内の指定されたアナライザーによって生成された結果のリストを取得します。この例では、名前が含まれているIAMロールのみを含めるように結果をフィルタリングしますCognito。

```
aws accessanalyzer list-findings-v2 \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
["AWS::IAM::Role"]}]'
```

出力:

```
{
  "findings": [
    {
      "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
      "createdAt": "2021-02-26T21:17:24.710000+00:00",
      "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
      "resource": "arn:aws:iam::111122223333:role/
Cognito_testpoolUnauth_Role",
      "resourceType": "AWS::IAM::Role",
      "resourceOwnerAccount": "111122223333",
      "status": "ACTIVE",
      "updatedAt": "2021-02-26T21:17:24.710000+00:00",
      "findingType": "ExternalAccess"
```

```
    },  
    {  
      "analyzedAt": "2024-02-16T18:17:47.888000+00:00",  
      "createdAt": "2021-02-26T21:17:50.905000+00:00",  
      "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",  
      "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",  
      "resourceType": "AWS::IAM::Role",  
      "resourceOwnerAccount": "111122223333",  
      "status": "ACTIVE",  
      "updatedAt": "2021-02-26T21:17:50.905000+00:00",  
      "findingType": "ExternalAccess"  
    }  
  ]  
}
```

詳細については、AWS IAM ユーザーガイドの[AWS 「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、「コマンドリファレンス」の[ListFindingsV2](#)を参照してください。
AWS CLI

list-findings

次のコード例は、list-findings を使用する方法を示しています。

AWS CLI

指定されたアナライザーによって生成された検出結果のリストを取得するには

次のlist-findings例では、AWS アカウント内の指定されたアナライザーによって生成された検出結果のリストを取得します。この例では、名前が含まれているIAMロールのみを含めるように結果をフィルタリングしますCognito。

```
aws accessanalyzer list-findings \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":  
["AWS::IAM::Role"]}}'
```

出力:

```
{
```

```
"findings": [
  {
    "id": "597f3bc2-3adc-4c18-9879-5c4b23485e46",
    "principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "action": [
      "sts:AssumeRoleWithWebIdentity"
    ],
    "resource": "arn:aws:iam::111122223333:role/
Cognito_testpoolUnauth_Role",
    "isPublic": false,
    "resourceType": "AWS::IAM::Role",
    "condition": {
      "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
    },
    "createdAt": "2021-02-26T21:17:24.710000+00:00",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "updatedAt": "2021-02-26T21:17:24.710000+00:00",
    "status": "ACTIVE",
    "resourceOwnerAccount": "111122223333"
  },
  {
    "id": "ce0e221a-85b9-4d52-91ff-d7678075442f",
    "principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "action": [
      "sts:AssumeRoleWithWebIdentity"
    ],
    "resource": "arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role",
    "isPublic": false,
    "resourceType": "AWS::IAM::Role",
    "condition": {
      "cognito-identity.amazonaws.com:aud": "us-
west-2:EXAMPLE0-0000-0000-0000-000000000000"
    },
    "createdAt": "2021-02-26T21:17:50.905000+00:00",
    "analyzedAt": "2024-02-16T18:17:47.888000+00:00",
    "updatedAt": "2021-02-26T21:17:50.905000+00:00",
    "status": "ACTIVE",
    "resourceOwnerAccount": "111122223333"
  }
]
```

```
]
}
```

詳細については、AWS IAM ユーザーガイドの[AWS 「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFindings](#)」の「」を参照してください。

list-policy-generations

次の例は、list-policy-generations を使用する方法を説明しています。

AWS CLI

過去 7 日間にリクエストされたすべてのポリシー生成を一覧表示するには

次のlist-policy-generations例では、過去 7 日間に AWS アカウントでリクエストされたすべてのポリシー生成を一覧表示します。

```
aws accessanalyzer list-policy-generations
```

出力:

```
{
  "policyGenerations": [
    {
      "completedOn": "2024-02-14T23:43:38+00:00",
      "jobId": "923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2",
      "principalArn": "arn:aws:iam::111122223333:role/Admin",
      "startedOn": "2024-02-14T23:43:02+00:00",
      "status": "CANCELED"
    },
    {
      "completedOn": "2024-02-14T22:47:01+00:00",
      "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
      "principalArn": "arn:aws:iam::111122223333:role/Admin",
      "startedOn": "2024-02-14T22:44:41+00:00",
      "status": "SUCCEEDED"
    }
  ]
}
```



```
}
```

詳細については、AWS IAM 「ユーザーガイド」のIAM [「Access Analyzer ポリシーの生成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicyGenerations](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

指定されたリソースに適用されたタグのリストを取得するには

次のlist-tags-for-resource例では、AWS アカウント内の指定されたリソースに適用されたタグのリストを取得します。

```
aws accessanalyzer list-tags-for-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
  ConsoleAnalyzer-account
```

出力:

```
{  
  "tags": {  
    "Zone-of-trust": "Account",  
    "Name": "ConsoleAnalyzer"  
  }  
}
```

詳細については、AWS IAM 「ユーザーガイド」のIAM [「Access Analyzer ポリシーの生成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

start-policy-generation

次の例は、start-policy-generation を使用する方法を説明しています。

AWS CLI

ポリシー生成リクエストを開始するには

次のstart-policy-generation例では、AWS アカウントでポリシー生成リクエストを開始します。

```
aws accessanalyzer start-policy-generation \  
  --policy-generation-details '{"principalArn": "arn:aws:iam::111122223333:role/  
Admin"}' \  
  --cloud-trail-details file://myfile.json
```

myfile.json の内容:

```
{  
  "accessRole": "arn:aws:iam::111122223333:role/service-role/  
AccessAnalyzerMonitorServiceRole",  
  "startTime": "2024-02-13T00:30:00Z",  
  "trails": [  
    {  
      "allRegions": true,  
      "cloudTrailArn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/my-  
trail"  
    }  
  ]  
}
```

出力:

```
{  
  "jobId": "c557dc4a-0338-4489-95dd-739014860ff9"  
}
```

詳細については、AWS IAM 「ユーザーガイド」の [IAM 「Access Analyzer ポリシーの生成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartPolicyGeneration](#)」の「」を参照してください。

start-resource-scan

次のコード例は、start-resource-scan を使用する方法を示しています。

AWS CLI

指定されたリソースに適用されるポリシーのスキャンをすぐに開始するには

次のstart-resource-scan例では、AWS アカウント内の指定されたリソースに適用されるポリシーのスキャンをすぐに開始します。

```
aws accessanalyzer start-resource-scan \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --resource-arn arn:aws:iam::111122223333:role/Cognito_testpoolAuth_Role
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[IAM 「Access Analyzer ポリシーの生成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartResourceScan](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

指定されたリソースにタグを追加するには

次のtag-resource例では、AWS アカウントの指定されたリソースにタグを追加します。

```
aws accessanalyzer tag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --tags Environment=dev,Purpose=testing
```

このコマンドでは何も出力されません。

詳細については、AWS IAM ユーザーガイドの[AWS 「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、`untag-resource` を使用する方法を説明しています。

AWS CLI

指定されたリソースからタグを削除するには

次の`untag-resource`例では、AWS アカウントの指定されたリソースからタグを削除します。

```
aws accessanalyzer untag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --tag-keys Environment Purpose
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「Identity and Access Management Access Analyzer の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-archive-rule

次の例は、`update-archive-rule` を使用する方法を説明しています。

AWS CLI

指定されたアーカイブルールの基準と値を更新するには

次の`update-archive-rule`例では、AWS アカウントで指定されたアーカイブルールの基準と値を更新します。

```
aws accessanalyzer update-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule \  
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":  
["AWS::IAM::Role"]}}'
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の[「アーカイブルール」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateArchiveRule](#)」の「」を参照してください。

update-findings

次のコード例は、update-findings を使用する方法を示しています。

AWS CLI

指定された検出結果のステータスを更新するには

次のupdate-findings例では、AWS アカウントで指定された検出結果のステータスを更新します。

```
aws accessanalyzer update-findings \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
UnusedAccess-ConsoleAnalyzer-organization \  
  --ids 4f319ac3-2e0c-4dc4-bf51-7013a086b6ae 780d586a-2cce-4f72-aff6-359d450e7500 \  
  --status ARCHIVED
```

このコマンドでは何も出力されません。

詳細については、AWS IAM 「ユーザーガイド」の[AWS 「Identity and Access Management Access Analyzer の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFindings](#)」の「」を参照してください。

validate-policy

次のコード例は、validate-policy を使用する方法を示しています。

AWS CLI

ポリシーの検証をリクエストし、検出結果のリストを返すには

次のvalidate-policy例では、ポリシーの検証をリクエストし、検出結果のリストを返します。この例のポリシーは、ウェブ ID フェデレーションに使用される Amazon Cognito ロールのロール信頼ポリシーです。信頼ポリシーから生成された検出結果は、使用されているロールの引き受けアクション が正しくないために、空のSid要素値とポリシープリンシパルがー

致しません `sts:AssumeRole`。Cognito で使用する正しいロールの引き受けアクションは `sts:AssumeRoleWithWebIdentity`。

```
aws accessanalyzer validate-policy \  
  --policy-document file://myfile.json \  
  --policy-type RESOURCE_POLICY
```

`myfile.json` の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "",  
      "Effect": "Allow",  
      "Principal": {  
        "Federated": "cognito-identity.amazonaws.com"  
      },  
      "Action": [  
        "sts:AssumeRole",  
        "sts:TagSession"  
      ],  
      "Condition": {  
        "StringEquals": {  
          "cognito-identity.amazonaws.com:aud": "us-west-2_EXAMPLE"  
        }  
      }  
    }  
  ]  
}
```

出力:

```
{  
  "findings": [  
    {  
      "findingDetails": "Add a value to the empty string in the Sid element.",  
      "findingType": "SUGGESTION",  
      "issueCode": "EMPTY_SID_VALUE",  
      "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/  
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-  
checks-suggestion-empty-sid-value",  
    }  
  ]  
}
```

```

    "locations": [
      {
        "path": [
          {
            "value": "Statement"
          },
          {
            "index": 0
          },
          {
            "value": "Sid"
          }
        ],
        "span": {
          "end": {
            "column": 21,
            "line": 5,
            "offset": 81
          },
          "start": {
            "column": 19,
            "line": 5,
            "offset": 79
          }
        }
      }
    ],
  },
  {
    "findingDetails": "The sts:AssumeRole action is invalid with the
following principal(s): cognito-identity.amazonaws.com. Use a SAML provider
principal with the sts:AssumeRoleWithSAML action or use an OIDC provider principal
with the sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if
you use either of the two options.",
    "findingType": "ERROR",
    "issueCode": "MISMATCHED_ACTION_FOR_PRINCIPAL",
    "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-mismatched-action-for-principal",
    "locations": [
      {
        "path": [
          {
            "value": "Statement"

```

```
    },
    {
      "index": 0
    },
    {
      "value": "Action"
    },
    {
      "index": 0
    }
  ],
  "span": {
    "end": {
      "column": 32,
      "line": 11,
      "offset": 274
    },
    "start": {
      "column": 16,
      "line": 11,
      "offset": 258
    }
  }
},
{
  "path": [
    {
      "value": "Statement"
    },
    {
      "index": 0
    },
    {
      "value": "Principal"
    },
    {
      "value": "Federated"
    }
  ],
  "span": {
    "end": {
      "column": 61,
      "line": 8,
      "offset": 202
    }
  }
}
```



```

    },
    "start": {
      "column": 29,
      "line": 8,
      "offset": 170
    }
  }
]
},
{
  "findingDetails": "The following actions: sts:TagSession are not supported by the condition key cognito-identity.amazonaws.com:aud. The condition will not be evaluated for these actions. We recommend that you move these actions to a different statement without this condition key.",
  "findingType": "ERROR",
  "issueCode": "UNSUPPORTED_ACTION_FOR_CONDITION_KEY",
  "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-checks-error-unsupported-action-for-condition-key",
  "locations": [
    {
      "path": [
        {
          "value": "Statement"
        },
        {
          "index": 0
        },
        {
          "value": "Action"
        },
        {
          "index": 1
        }
      ],
      "span": {
        "end": {
          "column": 32,
          "line": 12,
          "offset": 308
        },
        "start": {
          "column": 16,

```

```
        "line": 12,
        "offset": 292
      }
    },
    {
      "path": [
        {
          "value": "Statement"
        },
        {
          "index": 0
        },
        {
          "value": "Condition"
        },
        {
          "value": "StringEquals"
        },
        {
          "value": "cognito-identity.amazonaws.com:aud"
        }
      ],
      "span": {
        "end": {
          "column": 79,
          "line": 16,
          "offset": 464
        },
        "start": {
          "column": 58,
          "line": 16,
          "offset": 443
        }
      }
    }
  ]
}
```

詳細については、AWS IAM [「ユーザーガイド」](#)の[「ポリシーの検証の確認」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ValidatePolicy](#)」の「」を参照してください。

を使用した Image Builder の例 AWS CLI

次のコード例は、Image Builder AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-component

次の例は、create-component を使用する方法を説明しています。

AWS CLI

コンポーネントを作成するには

次のcreate-component例では、JSONドキュメントファイルを使用するコンポーネントを作成し、Amazon S3 バケットにアップロードされたYAML形式のコンポーネントドキュメントを参照します。

```
aws imagebuilder create-component \  
  --cli-input-json file://create-component.json
```

create-component.json の内容:

```
{
```

```
"name": "MyExampleComponent",
"semanticVersion": "2019.12.02",
"description": "An example component that builds, validates and tests an image",
"changeDescription": "Initial version.",
"platform": "Windows",
"uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"
}
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "componentBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/examplecomponent/2019.12.02/1"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[CreateComponent](#)」の「」を参照してください。

create-distribution-configuration

次の例は、create-distribution-configuration を使用する方法を説明しています。

AWS CLI

ディストリビューション設定を作成するには

次のcreate-distribution-configuration例では、JSON ファイルを使用してディストリビューション設定を作成します。

```
aws imagebuilder create-distribution-configuration \
  --cli-input-json file:/create-distribution-configuration.json
```

create-distribution-configuration.json の内容:

```
{
  "name": "MyExampleDistribution",
```

```

"description": "Copies AMI to eu-west-1",
"distributions": [
  {
    "region": "us-west-2",
    "amiDistributionConfiguration": {
      "name": "Name {{imagebuilder:buildDate}}",
      "description": "An example image name with parameter references",
      "amiTags": {
        "KeyName": "{{ssm:parameter_name}}"
      },
      "launchPermission": {
        "userIds": [
          "123456789012"
        ]
      }
    }
  },
  {
    "region": "eu-west-1",
    "amiDistributionConfiguration": {
      "name": "My {{imagebuilder:buildVersion}} image
{{imagebuilder:buildDate}}",
      "amiTags": {
        "KeyName": "Value"
      },
      "launchPermission": {
        "userIds": [
          "123456789012"
        ]
      }
    }
  }
]
}

```

出力:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution"
}

```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[CreateDistributionConfiguration](#)」の「」を参照してください。

create-image-pipeline

次のコード例は、create-image-pipeline を使用する方法を示しています。

AWS CLI

イメージパイプラインを作成するには

次のcreate-image-pipeline例では、JSON ファイルを使用してイメージパイプラインを作成します。

```
aws imagebuilder create-image-pipeline \  
  --cli-input-json file://create-image-pipeline.json
```

create-image-pipeline.json の内容:

```
{  
  "name": "MyWindows2016Pipeline",  
  "description": "Builds Windows 2016 Images",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",  
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
  "imageTestsConfiguration": {  
    "imageTestsEnabled": true,  
    "timeoutMinutes": 60  
  },  
  "schedule": {  
    "scheduleExpression": "cron(0 0 * * SUN)",  
    "pipelineExecutionStartCondition":  
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"  
  },  
  "status": "ENABLED"  
}
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[CreateImagePipeline](#)」の「」を参照してください。

create-image-recipe

次のコード例は、create-image-recipe を使用する方法を示しています。

AWS CLI

レシピを作成するには

次のcreate-image-recipe例では、JSON ファイルを使用してイメージレシピを作成します。コンポーネントは、指定された順序でインストールされます。

```
aws imagebuilder create-image-recipe \
  --cli-input-json file://create-image-recipe.json
```

create-image-recipe.json の内容:

```
{
  "name": "MyBasicRecipe",
  "description": "This example image recipe creates a Windows 2016 image.",
  "semanticVersion": "2019.12.03",
  "components":
  [
    {
      "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02/1"
    },
    {
```

```
        "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myimportedcomponent/1.0.0/1"
      }
    ],
    "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-
english-full-base-x86/xxxx.x.x"
  }
}
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[CreateImageRecipe](#)」の「」を参照してください。

create-image

次の例は、create-image を使用する方法を説明しています。

AWS CLI

イメージを作成するには

次のcreate-image例では、イメージを作成します。

```
aws imagebuilder create-image \  
  --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03 \  
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

出力:

```
{
```



```
"requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
"imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス [CreateImage](#)」の「」を参照してください。

create-infrastructure-configuration

次の例は、create-infrastructure-configuration を使用する方法を説明しています。

AWS CLI

インフラストラクチャ設定を作成するには

次のcreate-infrastructure-configuration例では、JSON ファイルを使用してインフラストラクチャ設定を作成します。

```
aws imagebuilder create-infrastructure-configuration \
  --cli-input-json file://create-infrastructure-configuration.json
```

create-infrastructure-configuration.json の内容:

```
{
  "name": "MyExampleInfrastructure",
  "description": "An example that will retain instances of failed builds",
  "instanceTypes": [
    "m5.large", "m5.xlarge"
  ],
  "instanceProfileName": "EC2InstanceProfileForImageBuilder",
  "securityGroupIds": [
    "sg-a1b2c3d4"
  ],
  "subnetId": "subnet-a1b2c3d4",
  "logging": {
    "s3Logs": {
      "s3BucketName": "bucket-name",
```

```
        "s3KeyPrefix": "bucket-path"
      }
    },
    "keyPair": "key-pair-name",
    "terminateInstanceOnFailure": false,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-topic-name"
  }
}
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[CreateInfrastructureConfiguration](#)」の「」を参照してください。

delete-component

次の例は、delete-component を使用する方法を説明しています。

AWS CLI

コンポーネントを削除するには

次のdelete-component例では、 を指定してコンポーネントビルドバージョンを削除します
ARN。

```
aws imagebuilder delete-component \  
  --component-build-version-arn arn:aws:imagebuilder:us-  
west-2:123456789012:component/myexamplecomponent/2019.12.02/1
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
"componentBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteComponent](#)」の「」を参照してください。

delete-image-pipeline

次のコード例は、delete-image-pipeline を使用する方法を示しています。

AWS CLI

イメージパイプラインを削除するには

次のdelete-image-pipeline例では、 を指定してイメージパイプラインを削除しますARN。

```
aws imagebuilder delete-image-pipeline \
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/mywindows2016pipeline"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteImagePipeline](#)」の「」を参照してください。

delete-image-recipe

次のコード例は、delete-image-recipe を使用する方法を示しています。

AWS CLI

イメージレシピを削除するには

次のdelete-image-recipe例では、 を指定してイメージレシピを削除しますARN。

```
aws imagebuilder delete-image-recipe \  
  --image-recipe-arn arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/  
mybasicrecipe/2019.12.03
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteImageRecipe](#)」の「」を参照してください。

delete-image

次のコード例は、delete-image を使用する方法を示しています。

AWS CLI

イメージを削除するには

次のdelete-image例では、 を指定してイメージビルドバージョンを削除しますARN。

```
aws imagebuilder delete-image \  
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-  
example-image/2019.12.02/1
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
"imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03/1"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteImage](#)」の「」を参照してください。

delete-infrastructure-configuration

次のコード例は、delete-infrastructure-configuration を使用する方法を示しています。

AWS CLI

インフラストラクチャ設定を削除するには

次のdelete-infrastructure-configuration例では、 を指定してイメージパイプラインを削除しますARN。

```
aws imagebuilder delete-infrastructure-configuration \  
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-  
east-1:123456789012:infrastructure-configuration/myexampleinfrastructure
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインのセットアップと管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteInfrastructureConfiguration](#)」の「」を参照してください。

get-component-policy

次の例は、get-component-policy を使用する方法を説明しています。

AWS CLI

コンポーネントポリシーの詳細を取得するには

次のget-component-policy例では、 を指定してコンポーネントポリシーの詳細を一覧表示しますARN。

```
aws imagebuilder get-component-policy \  
  --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.03/1
```

出力:

```
{  
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
    \"Allow\", \"Principal\": { \"AWS\": [ \"123456789012\" ] }, \"Action\":  
    [ \"imagebuilder:GetComponent\", \"imagebuilder:ListComponents\" ], \"Resource\":  
    [ \"arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.03/1\" ] } ] }\"  
}
```

詳細については、EC2「Image Builder ユーザーガイド」のAWS CLI「<<https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html>>`__`を使用した Image Builder Image Pipeline のセットアップと管理」を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[GetComponentPolicy](#)」の「」を参照してください。

get-component

次のコード例は、get-component を使用する方法を示しています。

AWS CLI

コンポーネントの詳細を取得するには

次のget-component例では、 を指定してコンポーネントの詳細を一覧表示しますARN。

```
aws imagebuilder get-component \  
  --component-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/component-name/1.0.0/1
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "component": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0/1",
    "name": "component-name",
    "version": "1.0.0",
    "type": "TEST",
    "platform": "Linux",
    "owner": "123456789012",
    "data": "name: HelloWorldTestingDocument\ndescription: This is hello world
testing document.\nschemaVersion: 1.0\n\nphases:\n - name: test\n   steps:\n
- name: HelloWorldStep\n   action: ExecuteBash\n   inputs:\n
commands:\n   - echo \"Hello World! Test.\\\"\\n\",
    "encrypted": true,
    "dateCreated": "2020-01-27T20:43:30.306Z",
    "tags": {}
  }
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[GetComponent](#)」の「」を参照してください。

get-distribution-configuration

次のコード例は、get-distribution-configuration を使用する方法を示しています。

AWS CLI

ディストリビューション設定の詳細を取得するには

次のget-distribution-configuration例では、ディストリビューション設定の詳細をを指定して表示しますARN。

```
aws imagebuilder get-distribution-configuration \
  --distribution-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "distributionConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/myexampledistribution",
    "name": "MyExampleDistribution",
    "description": "Copies AMI to eu-west-1 and exports to S3",
    "distributions": [
      {
        "region": "us-west-2",
        "amiDistributionConfiguration": {
          "name": "Name {{imagebuilder:buildDate}}",
          "description": "An example image name with parameter
references",
          "amiTags": {
            "KeyName": "{{ssm:parameter_name}}"
          },
          "launchPermission": {
            "userIds": [
              "123456789012"
            ]
          }
        },
        {
          "region": "eu-west-1",
          "amiDistributionConfiguration": {
            "name": "My {{imagebuilder:buildVersion}} image
{{imagebuilder:buildDate}}",
            "amiTags": {
              "KeyName": "Value"
            },
            "launchPermission": {
              "userIds": [
                "123456789012"
              ]
            }
          }
        }
      ]
    },
    "dateCreated": "2020-02-19T18:40:10.529Z",
    "tags": {}
  }
```



```
}  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[GetDistributionConfiguration](#)」の「」を参照してください。

get-image-pipeline

次の例は、get-image-pipeline を使用する方法を説明しています。

AWS CLI

イメージパイプラインの詳細を取得するには

次のget-image-pipeline例では、 を指定して、イメージパイプラインの詳細を一覧表示しますARN。

```
aws imagebuilder get-image-pipeline \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imagePipeline": {  
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline",  
    "name": "MyWindows2016Pipeline",  
    "description": "Builds Windows 2016 Images",  
    "platform": "Windows",  
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03",  
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",  
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
    "imageTestsConfiguration": {  
      "imageTestsEnabled": true,  

```

```

        "timeoutMinutes": 60
    },
    "schedule": {
        "scheduleExpression": "cron(0 0 * * SUN)",
        "pipelineExecutionStartCondition":
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
    },
    "status": "ENABLED",
    "dateCreated": "2020-02-19T19:04:01.253Z",
    "dateUpdated": "2020-02-19T19:04:01.253Z",
    "tags": {}
}
}

```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[GetImagePipeline](#)」の「」を参照してください。

get-image-policy

次のコード例は、get-image-policy を使用する方法を示しています。

AWS CLI

イメージポリシーの詳細を取得するには

次のget-image-policy例では、 を指定してイメージポリシーの詳細を一覧表示しますARN。

```

aws imagebuilder get-image-policy \
  --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/2019.12.03/1

```

出力:

```

{
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow\",
\"Principal\": { \"AWS\": [ \"123456789012\" ] }, \"Action\": [ \"imagebuilder:GetImage\",
\"imagebuilder:ListImages\" ], \"Resource\": [ \"arn:aws:imagebuilder:us-
west-2:123456789012:image/my-example-image/2019.12.03/1\" ] } ] }"
}

```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[GetImagePolicy](#)」の「」を参照してください。

get-image-recipe-policy

次の例は、get-image-recipe-policy を使用する方法を説明しています。

AWS CLI

イメージレシピポリシーの詳細を取得するには

次のget-image-recipe-policy例では、 を指定して、イメージレシピポリシーの詳細を一覧表示しますARN。

```
aws imagebuilder get-image-recipe-policy \  
  --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-recipe/2019.12.03/1
```

出力:

```
{  
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
    \"Allow\", \"Principal\": { \"AWS\": [ \"123456789012\" ] }, \"Action\":  
    [ \"imagebuilder:GetImageRecipe\", \"imagebuilder:ListImageRecipes\" ], \"Resource\":  
    [ \"arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-  
    recipe/2019.12.03/1\" ] } ] }\"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[GetImageRecipePolicy](#)」の「」を参照してください。

get-image

次のコード例は、get-image を使用する方法を示しています。

AWS CLI

イメージの詳細を取得するには

次のget-image例では、 を指定してイメージの詳細を一覧表示しますARN。

```
aws imagebuilder get-image \  
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03/1
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "image": {  
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03/1",  
    "name": "MyBasicRecipe",  
    "version": "2019.12.03/1",  
    "platform": "Windows",  
    "state": {  
      "status": "BUILDING"  
    },  
    "imageRecipe": {  
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03",  
      "name": "MyBasicRecipe",  
      "description": "This example image recipe creates a Windows 2016  
image.",  
      "platform": "Windows",  
      "version": "2019.12.03",  
      "components": [  
        {  
          "componentArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"  
        },  
        {  
          "componentArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:component/myimportedcomponent/1.0.0/1"  
        }  
      ],  
      "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-  
server-2016-english-full-base-x86/2019.12.17/1",
```

```
    "dateCreated": "2020-02-14T19:46:16.904Z",
    "tags": {}
  },
  "infrastructureConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "name": "MyExampleInfrastructure",
    "description": "An example that will retain instances of failed builds",
    "instanceTypes": [
      "m5.large",
      "m5.xlarge"
    ],
    "instanceProfileName": "EC2InstanceProfileForImageFactory",
    "securityGroupIds": [
      "sg-a1b2c3d4"
    ],
    "subnetId": "subnet-a1b2c3d4",
    "logging": {
      "s3Logs": {
        "s3BucketName": "bucket-name",
        "s3KeyPrefix": "bucket-path"
      }
    },
    "keyPair": "Sam",
    "terminateInstanceOnFailure": false,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
    "dateCreated": "2020-02-14T21:21:05.098Z",
    "tags": {}
  },
  "imageTestsConfiguration": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 720
  },
  "dateCreated": "2020-02-14T23:14:13.597Z",
  "outputResources": {
    "amis": []
  },
  "tags": {}
}
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder Image Pipeline のセットアップと管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[GetImage](#)」の「」を参照してください。

get-infrastructure-configuration

次のコード例は、get-infrastructure-configuration を使用する方法を示しています。

AWS CLI

インフラストラクチャ設定の詳細を取得するには

次のget-infrastructure-configuration例では、 を指定してインフラストラクチャ設定の詳細を一覧表示しますARN。

```
aws imagebuilder get-infrastructure-configuration \  
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "infrastructureConfiguration": {  
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-  
configuration/myexampleinfrastructure",  
    "name": "MyExampleInfrastructure",  
    "description": "An example that will retain instances of failed builds",  
    "instanceTypes": [  
      "m5.large",  
      "m5.xlarge"  
    ],  
    "instanceProfileName": "EC2InstanceProfileForImageBuilder",  
    "securityGroupIds": [  
      "sg-a48c95ef"  
    ],  
    "subnetId": "subnet-a48c95ef",  
    "logging": {  
      "s3Logs": {  
        "s3BucketName": "bucket-name",  
        "s3KeyPrefix": "bucket-path"  
      }  
    }  
  },  
}
```

```
    "keyPair": "Name",
    "terminateInstanceOnFailure": false,
    "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name",
    "dateCreated": "2020-02-19T19:11:51.858Z",
    "tags": {}
  }
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[GetInfrastructureConfiguration](#)」の「」を参照してください。

import-component

次のコード例は、import-component を使用する方法を示しています。

AWS CLI

コンポーネントをインポートするには

次のimport-component例では、JSON ファイルを使用して既存のスクリプトをインポートします。

```
aws imagebuilder import-component \
  --cli-input-json file://import-component.json
```

import-component.json の内容:

```
{
  "name": "MyImportedComponent",
  "semanticVersion": "1.0.0",
  "description": "An example of how to import a component",
  "changeDescription": "First commit message.",
  "format": "SHELL",
  "platform": "Windows",
  "type": "BUILD",
  "uri": "s3://s3-bucket-name/s3-bucket-path/component.yaml"
}
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "componentBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/myimportedcomponent/1.0.0/1"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ImportComponent](#)」の「」を参照してください。

list-component-build-versions

次のコード例は、list-component-build-versions を使用する方法を示しています。

AWS CLI

コンポーネントビルドバージョンを一覧表示するには

次のlist-component-build-versions例では、特定のセマンティックバージョンを持つコンポーネントビルドバージョンを一覧表示します。

```
aws imagebuilder list-component-build-versions --component-
version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02/1",
      "name": "MyExampleComponent",
      "version": "2019.12.02",
      "platform": "Windows",
      "type": "BUILD",
      "owner": "123456789012",
    }
  ]
}
```



```
    "description": "An example component that builds, validates and tests an
image",
    "changeDescription": "Initial version.",
    "dateCreated": "2020-02-19T18:53:45.940Z",
    "tags": {
      "KeyName": "KeyValue"
    }
  ]
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListComponentBuildVersions](#)」の「」を参照してください。

list-components

次のコード例は、list-components を使用する方法を示しています。

AWS CLI

すべてのコンポーネントセマンティックバージョンを一覧表示するには

次のlist-components例では、アクセスできるすべてのコンポーネントセマンティックバージョンを一覧表示します。必要に応じて、自分が所有するコンポーネント、Amazon が所有するコンポーネント、または他のアカウントによって共有されているコンポーネントを一覧表示するかどうかをフィルタリングできます。

```
aws imagebuilder list-components
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0",
      "name": "component-name",
      "version": "1.0.0",
```

```
        "platform": "Linux",
        "type": "TEST",
        "owner": "123456789012",
        "dateCreated": "2020-01-27T20:43:30.306Z"
    }
]
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListComponents](#)」の「」を参照してください。

list-distribution-configurations

次の例は、list-distribution-configurations を使用する方法を説明しています。

AWS CLI

ディストリビューションを一覧表示するには

次のlist-distribution-configurations例では、すべてのディストリビューションを一覧表示します。

```
aws imagebuilder list-distribution-configurations
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "distributionConfigurationSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/myexampledistribution",
      "name": "MyExampleDistribution",
      "description": "Copies AMI to eu-west-1 and exports to S3",
      "dateCreated": "2020-02-19T18:40:10.529Z",
      "tags": {
        "KeyName": "KeyValue"
      }
    }
  ]
}
```

```
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListDistributionConfigurations](#)」の「」を参照してください。

list-image-build-versions

次の例は、list-image-build-versions を使用する方法を説明しています。

AWS CLI

イメージビルドバージョンを一覧表示するには

次のlist-image-build-versions例では、セマンティックバージョンを持つすべてのイメージビルドバージョンを一覧表示します。

```
aws imagebuilder list-image-build-versions \  
  --image-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageSummaryList": [  
    {  
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03/7",  
      "name": "MyBasicRecipe",  
      "version": "2019.12.03/7",  
      "platform": "Windows",  
      "state": {  
        "status": "FAILED",  
        "reason": "Can't start SSM Automation for arn  
arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/7 during  
building. Parameter \"iamInstanceProfileName\" has a null value."  
      },  
      "owner": "123456789012",  
      "dateCreated": "2020-02-19T18:56:11.511Z",  
      "outputResources": {
```

```
        "amis": []
      },
      "tags": {}
    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/6",
      "name": "MyBasicRecipe",
      "version": "2019.12.03/6",
      "platform": "Windows",
      "state": {
        "status": "FAILED",
        "reason": "An internal error has occurred."
      },
      "owner": "123456789012",
      "dateCreated": "2020-02-18T22:49:08.142Z",
      "outputResources": {
        "amis": [
          {
            "region": "us-west-2",
            "image": "ami-a1b2c3d4567890ab",
            "name": "MyBasicRecipe 2020-02-18T22-49-38.704Z",
            "description": "This example image recipe creates a Windows
2016 image."
          },
          {
            "region": "us-west-2",
            "image": "ami-a1b2c3d4567890ab",
            "name": "Name 2020-02-18T22-49-08.131Z",
            "description": "Copies AMI to eu-west-2 and exports to S3"
          },
          {
            "region": "eu-west-2",
            "image": "ami-a1b2c3d4567890ab",
            "name": "My 6 image 2020-02-18T22-49-08.131Z",
            "description": "Copies AMI to eu-west-2 and exports to S3"
          }
        ]
      },
      "tags": {}
    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/5",
```

```
    "name": "MyBasicRecipe",
    "version": "2019.12.03/5",
    "platform": "Windows",
    "state": {
      "status": "AVAILABLE"
    },
    "owner": "123456789012",
    "dateCreated": "2020-02-18T16:51:48.403Z",
    "outputResources": {
      "amis": [
        {
          "region": "us-west-2",
          "image": "ami-a1b2c3d4567890ab",
          "name": "MyBasicRecipe 2020-02-18T16-52-18.965Z",
          "description": "This example image recipe creates a Windows
2016 image."
        }
      ]
    },
    "tags": {}
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/4",
    "name": "MyBasicRecipe",
    "version": "2019.12.03/4",
    "platform": "Windows",
    "state": {
      "status": "AVAILABLE"
    },
    "owner": "123456789012",
    "dateCreated": "2020-02-18T16:50:01.827Z",
    "outputResources": {
      "amis": [
        {
          "region": "us-west-2",
          "image": "ami-a1b2c3d4567890ab",
          "name": "MyBasicRecipe 2020-02-18T16-50-32.280Z",
          "description": "This example image recipe creates a Windows
2016 image."
        }
      ]
    },
    "tags": {}
  }
}
```

```

    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/3",
      "name": "MyBasicRecipe",
      "version": "2019.12.03/3",
      "platform": "Windows",
      "state": {
        "status": "AVAILABLE"
      },
      "owner": "123456789012",
      "dateCreated": "2020-02-14T23:14:13.597Z",
      "outputResources": {
        "amis": [
          {
            "region": "us-west-2",
            "image": "ami-a1b2c3d4567890ab",
            "name": "MyBasicRecipe 2020-02-14T23-14-44.243Z",
            "description": "This example image recipe creates a Windows
2016 image."
          }
        ]
      },
      "tags": {}
    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/2",
      "name": "MyBasicRecipe",
      "version": "2019.12.03/2",
      "platform": "Windows",
      "state": {
        "status": "FAILED",
        "reason": "SSM execution 'a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'
failed with status = 'Failed' and failure message = 'Step fails when it is
verifying the command has completed. Command a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
returns unexpected invocation result: \n{Status=[Failed], ResponseCode=[1],
Output=[\n-----ERROR-----\nfailed to run commands: exit status 1],
OutputPayload=[{\"Status\": \"Failed\", \"ResponseCode\": 1, \"Output\": \"\
\n-----ERROR-----\nfailed to run commands: exit status 1\", \"CommandId\":
\"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111\"}], CommandId=[a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111]}. Please refer to Automation Service Troubleshooting Guide for more
diagnosis details.'"
      }
    },
  ],
}

```

```
    "owner": "123456789012",
    "dateCreated": "2020-02-14T22:57:42.593Z",
    "outputResources": {
      "amis": []
    },
    "tags": {}
  }
]
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListImageBuildVersions](#)」の「」を参照してください。

list-image-pipeline-images

次のコード例は、list-image-pipeline-images を使用する方法を示しています。

AWS CLI

イメージパイプラインパイプラインイメージを一覧表示するには

次のlist-image-pipeline-images例では、特定のイメージパイプラインによって作成されたすべてのイメージを一覧表示します。

```
aws imagebuilder list-image-pipeline-images \
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imagePipelineList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
      "name": "MyWindows2016Pipeline",
      "description": "Builds Windows 2016 Images",
      "platform": "Windows",
```

```
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/mybasicrecipe/2019.12.03",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/myexampledistribution",
    "imageTestsConfiguration": {
      "imageTestsEnabled": true,
      "timeoutMinutes": 60
    },
    "schedule": {
      "scheduleExpression": "cron(0 0 * * SUN)",
      "pipelineExecutionStartCondition":
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
    },
    "status": "ENABLED",
    "dateCreated": "2020-02-19T19:04:01.253Z",
    "dateUpdated": "2020-02-19T19:04:01.253Z",
    "tags": {
      "KeyName": "KeyValue"
    }
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/sam",
    "name": "PipelineName",
    "platform": "Linux",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/recipe-name-a1b2c3d45678/1.0.0",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/infrastructureconfiguration-name-a1b2c3d45678",
    "imageTestsConfiguration": {
      "imageTestsEnabled": true,
      "timeoutMinutes": 720
    },
    "status": "ENABLED",
    "dateCreated": "2019-12-16T18:19:02.068Z",
    "dateUpdated": "2019-12-16T18:19:02.068Z",
    "tags": {
      "KeyName": "KeyValue"
    }
  }
]
```



```
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListImagePipelineImages](#)」の「」を参照してください。

list-image-recipes

次のコード例は、list-image-recipes を使用する方法を示しています。

AWS CLI

イメージレシピを一覧表示するには

次のlist-image-recipes例では、すべてのイメージレシピを一覧表示します。

```
aws imagebuilder list-image-recipes
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageRecipeSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/mybasicrecipe/2019.12.03",
      "name": "MyBasicRecipe",
      "platform": "Windows",
      "owner": "123456789012",
      "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-english-full-base-x86/2019.x.x",
      "dateCreated": "2020-02-19T18:54:25.975Z",
      "tags": {
        "KeyName": "KeyValue"
      }
    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/recipe-name-a1b2c3d45678/1.0.0",
      "name": "recipe-name-a1b2c3d45678",
      "platform": "Linux",
```

```
        "owner": "123456789012",
        "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/amazon-linux-2-
x86/2019.11.21",
        "dateCreated": "2019-12-16T18:19:00.120Z",
        "tags": {
            "KeyName": "KeyValue"
        }
    }
]
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListImageRecipes](#)」の「」を参照してください。

list-images

次の例は、list-images を使用する方法を説明しています。

AWS CLI

イメージを一覧表示するには

次のlist-images例では、アクセスできるすべてのセマンティックバージョンを一覧表示します。

```
aws imagebuilder list-images
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03",
      "name": "MyBasicRecipe",
      "version": "2019.12.03",
      "platform": "Windows",
      "owner": "123456789012",
      "dateCreated": "2020-02-14T21:29:18.810Z"
    }
  ]
}
```

```
    }  
  ]  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListImages](#)」の「」を参照してください。

list-infrastructure-configurations

次の例は、list-infrastructure-configurations を使用する方法を説明しています。

AWS CLI

インフラストラクチャ設定を一覧表示するには

次のlist-infrastructure-configurations例では、すべてのインフラストラクチャ設定を一覧表示します。

```
aws imagebuilder list-infrastructure-configurations
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "infrastructureConfigurationSummaryList": [  
    {  
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-  
configuration/myexampleinfrastructure",  
      "name": "MyExampleInfrastructure",  
      "description": "An example that will retain instances of failed builds",  
      "dateCreated": "2020-02-19T19:11:51.858Z",  
      "tags": {}  
    },  
    {  
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-  
configuration/infrastructureconfiguration-name-a1b2c3d45678",  
      "name": "infrastructureConfiguration-name-a1b2c3d45678",  
      "dateCreated": "2019-12-16T18:19:01.038Z",  
      "tags": {  
        "KeyName": "KeyValue"  
      }  
    }  
  ]  
}
```

```

    }
  }
]
}

```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListInfrastructureConfigurations](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

特定のリソースのタグを一覧表示するには

次のlist-tags-for-resource例では、特定のリソースのすべてのタグを一覧表示します。

```

aws imagebuilder list-tags-for-resource \
  --resource-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline

```

出力:

```

{
  "tags": {
    "KeyName": "KeyValue"
  }
}

```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。 EC2

- API 詳細については、AWS CLI「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-component-policy

次のコード例は、put-component-policy を使用する方法を示しています。

AWS CLI

リソースポリシーをコンポーネントに適用するには

次のput-component-policyコマンドは、ビルドコンポーネントにリソースポリシーを適用して、ビルドコンポーネントのクロスアカウント共有を有効にします。RAM CLI コマンド を使用することをお勧めしますcreate-resource-share。EC2 Image Builder CLI コマンド を使用する場合はput-component-policy、リソースを共有するすべてのプリンシパルにリソースを表示promote-resource-share-create-from-policyするために、RAM CLI コマンドも使用する必要があります。

```
aws imagebuilder put-component-policy \  
  --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/  
examplecomponent/2019.12.02/1 \  
  --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":  
"Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":  
[ "imagebuilder:GetComponent", "imagebuilder:ListComponents" ],  
"Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:component/  
examplecomponent/2019.12.02/1" ] } ] }'
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/  
examplecomponent/2019.12.02/1"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[PutComponentPolicy](#)」の「」を参照してください。

put-image-policy

次の例は、put-image-policy を使用する方法を説明しています。

AWS CLI

リソースポリシーをイメージに適用するには

次のput-image-policyコマンドは、イメージにリソースポリシーを適用して、イメージのクロスアカウント共有を有効にします。RAM CLI コマンド を使用することをお勧めします create-resource-share。EC2 Image Builder CLI コマンド を使用する場合は put-image-policy、リソースを共有するすべてのプリンシパルにリソースを表示するために、 promote-resource-share-create-from-policy RAM CLI コマンドも使用する必要があります。

```
aws imagebuilder put-image-policy \  
  --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1 \  
  --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow",  
  "Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetImage",  
  "imagebuilder:ListImages" ], "Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1" ] } ] }'
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/example-image/2019.12.02/1"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI 「コマンドリファレンス[PutImagePolicy](#)」の「」を参照してください。

put-image-recipe-policy

次のコード例は、put-image-recipe-policy を使用する方法を示しています。

AWS CLI

リソースポリシーをイメージレシピに適用するには

次のput-image-recipe-policyコマンドは、イメージレシピにリソースポリシーを適用して、イメージレシピのクロスアカウント共有を有効にします。RAM CLI コマンド を使用することをお勧めします create-resource-share。EC2 Image Builder CLI コマンド を使用する場合はput-image-recipe-policy、リソースを共有するすべてのプリンシパルにリソースを表

示 `promote-resource-share-create-from-policy` するために、RAM CLI コマンドも使用する必要があります。

```
aws imagebuilder put-image-recipe-policy \  
  --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
example-image-recipe/2019.12.02 \  
  --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect":  
"Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":  
[ "imagebuilder:GetImageRecipe", "imagebuilder:ListImageRecipes" ], "Resource":  
[ "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/example-image-  
recipe/2019.12.02" ] } ] }'
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
example-image-recipe/2019.12.02/1"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス [PutImageRecipePolicy](#)」の「」を参照してください。

start-image-pipeline-execution

次の例は、`start-image-pipeline-execution` を使用する方法を説明しています。

AWS CLI

イメージパイプラインを手動で開始するには

次の `start-image-pipeline-execution` 例では、イメージパイプラインを手動で開始します。

```
aws imagebuilder start-image-pipeline-execution \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline
```

出力:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[StartImagePipelineExecution](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを付けるには

次のtag-resource例では、JSON ファイルを使用して EC2 Image Builder にリソースを追加およびタグ付けします。

```
aws imagebuilder tag-resource \
  --cli-input-json file://tag-resource.json
```

tag-resource.json の内容:

```
{
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
  "tags": {
    "KeyName": "KeyValue"
  }
}
```

このコマンドでは何も出力されません。

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、`untag-resource` を使用方法を示しています。

AWS CLI

リソースからタグを削除するには

次の`untag-resource`例では、JSON ファイルを使用してリソースからタグを削除します。

```
aws imagebuilder untag-resource \  
  --cli-input-json file://tag-resource.json
```

`untag-resource.json` の内容:

```
{  
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline",  
  "tagKeys": [  
    "KeyName"  
  ]  
}
```

このコマンドでは何も出力されません。

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI](#)」を参照してください。EC2

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-distribution-configuration

次のコード例は、`update-distribution-configuration` を使用方法を示しています。

AWS CLI

ディストリビューション設定を更新するには

次のupdate-distribution-configuration例では、JSON ファイルを使用してディストリビューション設定を更新します。

```
aws imagebuilder update-distribution-configuration \  
  --cli-input-json file://update-distribution-configuration.json
```

update-distribution-configuration.json の内容:

```
{  
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
  "description": "Copies AMI to eu-west-2 and exports to S3",  
  "distributions": [  
    {  
      "region": "us-west-2",  
      "amiDistributionConfiguration": {  
        "name": "Name {{imagebuilder:buildDate}}",  
        "description": "An example image name with parameter references"  
      }  
    },  
    {  
      "region": "eu-west-2",  
      "amiDistributionConfiguration": {  
        "name": "My {{imagebuilder:buildVersion}} image  
{{imagebuilder:buildDate}}"  
      }  
    }  
  ]  
}
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateDistributionConfiguration](#)」の「」を参照してください。

update-image-pipeline

次の例は、update-image-pipeline を使用する方法を説明しています。

AWS CLI

イメージパイプラインを更新するには

次のupdate-image-pipeline例では、JSON ファイルを使用してイメージパイプラインを更新します。

```
aws imagebuilder update-image-pipeline \  
  --cli-input-json file://update-image-pipeline.json
```

update-image-pipeline.json の内容:

```
{  
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",  
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
  "imageTestsConfiguration": {  
    "imageTestsEnabled": true,  
    "timeoutMinutes": 120  
  },  
  "schedule": {  
    "scheduleExpression": "cron(0 0 * * MON)",  
    "pipelineExecutionStartCondition":  
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"  
  },  
  "status": "DISABLED"  
}
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateImagePipeline](#)」の「」を参照してください。

update-infrastructure-configuration

次のコード例は、update-infrastructure-configuration を使用する方法を示しています。

AWS CLI

インフラストラクチャ設定を更新するには

次のupdate-infrastructure-configuration例では、JSON ファイルを使用してインフラストラクチャ設定を更新します。

```
aws imagebuilder update-infrastructure-configuration \  
--cli-input-json file:/update-infrastructure-configuration.json
```

update-infrastructure-configuration.json の内容:

```
{  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure",  
  "description": "An example that will terminate instances of failed builds",  
  "instanceTypes": [  
    "m5.large", "m5.2xlarge"  
  ],  
  "instanceProfileName": "EC2InstanceProfileForImageFactory",  
  "securityGroupIds": [  
    "sg-a48c95ef"  
  ],  
  "subnetId": "subnet-a48c95ef",  
  "logging": {  
    "s3Logs": {  
      "s3BucketName": "bucket-name",  
      "s3KeyPrefix": "bucket-path"  
    }  
  },  
  "terminateInstanceOnFailure": true,  
  "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:sns-name"
```

```
}
```

出力:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[EC2「Image Builder ユーザーガイド」の「を使用した Image Builder イメージパイプラインの設定と管理 AWS CLI」](#)を参照してください。EC2

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateInfrastructureConfiguration](#)」の「」を参照してください。

を使用した Incident Manager の例 AWS CLI

次のコード例は、Incident Manager AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-replication-set

次の例は、create-replication-set を使用する方法を説明しています。

AWS CLI

レプリケーションセットを作成するには

次のcreate-replication-set例では、Incident Manager が Amazon Web Services アカウントのデータをレプリケートおよび暗号化するために使用するレプリケーションセットを作成しま

す。この例では、レプリケーションセットの作成時に us-east-1 リージョンと us-east-2 リージョンを使用します。

```
aws ssm-incidents create-replication-set \  
  --regions '{"us-east-1": {"sseKmsKeyId": "arn:aws:kms:us-  
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}, "us-east-2":  
  {"sseKmsKeyId": "arn:aws:kms:us-  
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"}}'
```

出力:

```
{  
  "replicationSetArns": [  
    "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
bb3f-413c-08df53673b57"  
  ]  
}
```

詳細については、[Incident Manager ユーザーガイドの「Incident Manager レプリケーションセットの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateReplicationSet](#)」の「」を参照してください。

create-response-plan

次の例は、create-response-plan を使用する方法を説明しています。

AWS CLI

レスポンスプランを作成するには

次のcreate-response-plan例では、指定された詳細を含むレスポンスプランを作成します。

```
aws ssm-incidents create-response-plan \  
  --chat-channel '{"chatbotSns": [{"arn:aws:sns:us-  
east-1:111122223333:Standard_User"}]}' \  
  --display-name "Example response plan" \  
  --incident-template '{"impact": 5, "title": "example-incident"}' \  
  --name "example-response" \  
  --actions '[{"ssmAutomation": {"documentName": "AWSIncidents-  
CriticalIncidentRunbookTemplate", "documentVersion": "$DEFAULT",
```

```
"roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-
incidents.amazonaws.com/AWSServiceRoleForIncidentManager", "targetAccount":
"RESPONSE_PLAN_OWNER_ACCOUNT"]}]' \
--engagements '["arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"]'
```

出力:

```
{
  "arn": "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
}
```

詳細については、[「Incident Manager ユーザーガイド」の「Incident preparation」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResponsePlan](#)」の「」を参照してください。

create-timeline-event

次のコード例は、create-timeline-event を使用する方法を示しています。

AWS CLI

例 1: カスタムタイムラインイベントを作成するには

次のcreate-timeline-event例では、指定されたインシデントの指定された時間にカスタムタイムラインイベントを作成します。

```
aws ssm-incidents create-timeline-event \
  --event-data "\"example timeline event\"" \
  --event-time 2022-10-01T20:30:00.000 \
  --event-type "Custom Event" \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
```

出力:

```
{
  "eventId": "c0bcc885-a41d-eb01-b4ab-9d2deEXAMPLE",
  "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
```

```
}
```

例 2: インシデントノートを使用してタイムラインイベントを作成するには

次のcreate-timeline-event例では、「インシデントメモ」パネルにリストされているタイムラインイベントを作成します。

```
aws ssm-incidents create-timeline-event \  
  --event-data "\"New Note\"" \  
  --event-type "Note" \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE" \  
  --event-time 2023-06-20T12:06:00.000 \  
  --event-references '["resource": "arn:aws:ssm-incidents::111122223333:incident-  
record/Test/6cc46130-ca6c-3b38-68f1-f6abeEXAMPLE"]]'
```

出力:

```
{  
  "eventId": "a41dc885-c0bc-b4ab-eb01-de9d2EXAMPLE",  
  "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"  
}
```

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTimelineEvent](#)」の「」を参照してください。

delete-incident-record

次の例は、delete-incident-record を使用する方法を説明しています。

AWS CLI

インシデントレコードを削除するには

次のdelete-incident-record例では、指定されたインシデントレコードを削除します。

```
aws ssm-incidents delete-incident-record \  

```



```
--arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

このコマンドでは何も出力されません。

詳細については、[Incident Manager ユーザーガイド](#)の「Incident Tracking」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIncidentRecord](#)」の「」を参照してください。

delete-replication-set

次の例は、delete-replication-set を使用する方法を説明しています。

AWS CLI

レプリケーションセットを削除するには

次のdelete-replication-set例では、Amazon Web Services アカウントからレプリケーションセットを削除します。レプリケーションセットを削除すると、すべての Incident Manager データも削除されます。これは元に戻すことはできません。

```
aws ssm-incidents delete-replication-set \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
bb3f-413c-08df53673b57"
```

このコマンドでは何も出力されません。

詳細については、[Incident Manager ユーザーガイド](#)の「[Incident Manager レプリケーションセットの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteReplicationSet](#)」の「」を参照してください。

delete-resource-policy

次の例は、delete-resource-policy を使用する方法を説明しています。

AWS CLI

リソースポリシーを削除するには

次のdelete-resource-policy例では、レスポンスプランからリソースポリシーを削除します。これにより、レスポンスプランが共有されたプリンシパルまたは組織からのアクセスが取り消されます。

```
aws ssm-incidents delete-resource-policy \  
  --policy-id "be8b57191f0371f1c6827341aa3f0a03" \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

このコマンドでは何も出力されません。

詳細については、「[Incident Manager ユーザーガイド](#)」の「[共有連絡先と対応計画の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteResourcePolicy](#)」の「」を参照してください。

delete-response-plan

次のコード例は、delete-response-plan を使用する方法を示しています。

AWS CLI

レスポンスプランを削除するには

次のdelete-response-plan例では、指定されたレスポンスプランを削除します。

```
aws ssm-incidents delete-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
```

このコマンドでは何も出力されません。

詳細については、「[Incident Manager ユーザーガイド](#)」の「[Incident preparation](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteResponsePlan](#)」の「」を参照してください。

delete-timeline-event

次のコード例は、delete-timeline-event を使用する方法を示しています。

AWS CLI

タイムラインイベントを削除するには

次のdelete-timeline-event例では、指定されたインシデントレコードからカスタムタイムラインイベントを削除します。

```
aws ssm-incidents delete-timeline-event \  
  --event-id "c0bcc885-a41d-eb01-b4ab-9d2de193643c" \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

このコマンドでは何も出力されません。

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTimelineEvent](#)」の「」を参照してください。

get-incident-record

次のコード例は、get-incident-record を使用する方法を示しています。

AWS CLI

インシデントレコードを取得するには

次のget-incident-record例では、指定されたインシデントレコードの詳細を取得します。

```
aws ssm-incidents get-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-  
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

出力:

```
{  
  "incidentRecord": {  
    "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-  
Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",  
    "automationExecutions": [],
```

```
"creationTime": "2021-05-21T18:16:57.579000+00:00",
"dedupeString": "c4bcc812-85e7-938d-2b78-17181176ee1a",
"impact": 5,
"incidentRecordSource": {
  "createdBy": "arn:aws:iam::111122223333:user/drალიატპ",
  "invokedBy": "arn:aws:iam::111122223333:user/drალიატპ",
  "source": "aws.ssm-incidents.custom"
},
"lastModifiedBy": "arn:aws:iam::111122223333:user/drალიატპ",
"lastModifiedTime": "2021-05-21T18:16:59.149000+00:00",
"notificationTargets": [],
"status": "OPEN",
"title": "Example-Incident"
}
}
```

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIncidentRecord](#)」の「」を参照してください。

get-replication-set

次のコード例は、get-replication-set を使用する方法を示しています。

AWS CLI

レプリケーションセットを取得するには

次のget-replication-set例では、Incident Manager が Amazon Web Services アカウントのデータをレプリケートおよび暗号化するために使用するレプリケーションセットの詳細を取得します。

```
aws ssm-incidents get-replication-set \
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
  bb3f-413c-08df53673b57"
```

出力:

```
{
  "replicationSet": {
```

```
"createdBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
"createdTime": "2021-05-14T17:57:22.010000+00:00",
"deletionProtected": false,
"lastModifiedBy": "arn:aws:sts::111122223333:assumed-role/Admin/username",
"lastModifiedTime": "2021-05-14T17:57:22.010000+00:00",
"regionMap": {
  "us-east-1": {
    "sseKmsKeyId": "DefaultKey",
    "status": "ACTIVE"
  },
  "us-east-2": {
    "sseKmsKeyId": "DefaultKey",
    "status": "ACTIVE",
    "statusMessage": "Tagging inaccessible"
  }
},
"status": "ACTIVE"
}
```

詳細については、[「Incident Manager ユーザーガイド」の「Incident Manager レプリケーションセットの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetReplicationSet](#)」の「」を参照してください。

get-resource-policies

次の例は、get-resource-policies を使用する方法を説明しています。

AWS CLI

レスポンスプランのリソースポリシーを一覧表示するには

次のcommand-name例では、指定されたレスポンスプランに関連付けられたリソースポリシーを一覧表示します。

```
aws ssm-incidents get-resource-policies \
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

出力:

```
{
  "resourcePolicies": [
    {
      "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"d901b37a-dbb0-458a-8842-75575c464219-external-principals\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":{\"arn:aws:iam::222233334444:root\"}},\"Action\":[\"ssm-incidents:GetResponsePlan\",\"ssm-incidents:StartIncident\",\"ssm-incidents:UpdateIncidentRecord\",\"ssm-incidents:GetIncidentRecord\",\"ssm-incidents:CreateTimelineEvent\",\"ssm-incidents:UpdateTimelineEvent\",\"ssm-incidents:GetTimelineEvent\",\"ssm-incidents:ListTimelineEvents\",\"ssm-incidents:UpdateRelatedItems\",\"ssm-incidents:ListRelatedItems\"]},\"Resource\":[\"arn:aws:ssm-incidents:*:111122223333:response-plan/Example-Response-Plan\",\"arn:aws:ssm-incidents:*:111122223333:incident-record/Example-Response-Plan/*\"]}]}",
      "policyId": "be8b57191f0371f1c6827341aa3f0a03",
      "ramResourceShareRegion": "us-east-1"
    }
  ]
}
```

詳細については、「Incident Manager ユーザーガイド」の「[共有連絡先と対応計画の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourcePolicies](#)」の「」を参照してください。

get-response-plan

次の例は、get-response-plan を使用する方法を説明しています。

AWS CLI

レスポンスプランの詳細を取得するには

次のcommand-name例では、AWS アカウントで指定されたレスポンスプランの詳細を取得します。

```
aws ssm-incidents get-response-plan \
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

出力:

```
{
  "actions": [
    {
      "ssmAutomation": {
        "documentName": "AWSIncidents-CriticalIncidentRunbookTemplate",
        "documentVersion": "$DEFAULT",
        "roleArn": "arn:aws:iam::111122223333:role/aws-service-role/ssm-
incidents.amazonaws.com/AWSServiceRoleForIncidentManager",
        "targetAccount": "RESPONSE_PLAN_OWNER_ACCOUNT"
      }
    }
  ],
  "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-
Plan",
  "chatChannel": {
    "chatbotSns": [
      "arn:aws:sns:us-east-1:111122223333:Standard_User"
    ]
  },
  "displayName": "Example response plan",
  "engagements": [
    "arn:aws:ssm-contacts:us-east-1:111122223333:contact/example"
  ],
  "incidentTemplate": {
    "impact": 5,
    "title": "Example-Incident"
  },
  "name": "Example-Response-Plan"
}
```

詳細については、[「Incident Manager ユーザーガイド」の「Incident preparation」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResponsePlan](#)」の「」を参照してください。

get-timeline-event

次のコード例は、get-timeline-event を使用する方法を示しています。

AWS CLI

タイムラインイベントの詳細を取得するには

次のget-timeline-event例では、指定されたタイムラインイベントの詳細を返します。

```
aws ssm-incidents get-timeline-event \  
  --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

出力:

```
{  
  "event": {  
    "eventData": "\"Incident Started\"",  
    "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",  
    "eventTime": "2021-05-21T18:16:57+00:00",  
    "eventType": "Custom Event",  
    "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",  
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
  }  
}
```

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTimelineEvent](#)」の「」を参照してください。

list-incident-records

次の例は、list-incident-records を使用する方法を説明しています。

AWS CLI

インシデントレコードを一覧表示するには

次のcommand-name例では、Amazon Web Services アカウントのインシデントレコードを一覧表示します。

```
aws ssm-incidents list-incident-records
```

出力:


```
{
  "incidentRecordSummaries": [
    {
      "arn": "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308",
      "creationTime": "2021-05-21T18:16:57.579000+00:00",
      "impact": 5,
      "incidentRecordSource": {
        "createdBy": "arn:aws:iam::111122223333:user/draliatp",
        "invokedBy": "arn:aws:iam::111122223333:user/draliatp",
        "source": "aws.ssm-incidents.custom"
      },
      "status": "OPEN",
      "title": "Example-Incident"
    }
  ]
}
```

詳細については、[「Incident Manager ユーザーガイド」の「Incident list」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListIncidentRecords](#)」の「」を参照してください。

list-related-items

次の例は、list-related-items を使用する方法を説明しています。

AWS CLI

関連項目を一覧表示するには

次のlist-related-items例では、指定されたインシデントの関連項目を一覧表示します。

```
aws ssm-incidents list-related-items \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

出力:

```
{
  "relatedItems": [
    {
```

```
    "identifier": {
      "type": "OTHER",
      "value": {
        "url": "https://console.aws.amazon.com/systems-manager/opsitems/
oi-8ef82158e190/workbench?region=us-east-1"
      }
    },
    "title": "Example related item"
  },
  {
    "identifier": {
      "type": "PARENT",
      "value": {
        "arn": "arn:aws:ssm:us-east-1:111122223333:opsitem/
oi-8084126392ac"
      }
    },
    "title": "parentItem"
  }
]
}
```

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRelatedItems](#)」の「」を参照してください。

list-replication-sets

次のコード例は、list-replication-sets を使用する方法を示しています。

AWS CLI

レプリケーションセットを一覧表示するには

次のlist-replication-set例では、Incident Manager が AWS アカウント内のデータをレプリケートおよび暗号化するために使用するレプリケーションセットを一覧表示します。

```
aws ssm-incidents list-replication-sets
```

出力:

```
{
  "replicationSetArns": [
    "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
  ]
}
```

詳細については、[Incident Manager ユーザーガイドの「Incident Manager レプリケーションセットの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListReplicationSets](#)」の「」を参照してください。

list-response-plans

次のコード例は、list-response-plans を使用する方法を示しています。

AWS CLI

利用可能なレスポンスプランを一覧表示するには

次のlist-response-plans例は、Amazon Web Services アカウントで使用可能なレスポンスプランを一覧表示します。

```
aws ssm-incidents list-response-plans
```

出力:

```
{
  "responsePlanSummaries": [
    {
      "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan",
      "displayName": "Example response plan",
      "name": "Example-Response-Plan"
    }
  ]
}
```

詳細については、[「Incident Manager ユーザーガイド」の「Incident preparation」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResponsePlans](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

レスポンスプランのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたレスポンスプランに関連付けられたタグを一覧表示します。

```
aws ssm-incidents list-tags-for-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
  Response-Plan"
```

出力:

```
{  
  "tags": {  
    "group1": "1"  
  }  
}
```

詳細については、「Incident Manager ユーザーガイド」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-timeline-events

次のコード例は、list-timeline-events を使用する方法を示しています。

AWS CLI

インシデントのタイムラインイベントを一覧表示するには

次のcommand-name例では、指定されたインシデントのタイムラインイベントを一覧表示します。

```
aws ssm-incidents list-timeline-events \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

出力:

```
{  
  "eventSummaries": [  
    {  
      "eventId": "8cbcc889-35e1-a42d-2429-d6f100799915",  
      "eventTime": "2021-05-21T22:36:13.766000+00:00",  
      "eventType": "SSM Incident Record Update",  
      "eventUpdatedTime": "2021-05-21T22:36:13.766000+00:00",  
      "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-  
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
    },  
    {  
      "eventId": "a2bcc825-aab5-1787-c605-f9bb2640d85b",  
      "eventTime": "2021-05-21T18:58:46.443000+00:00",  
      "eventType": "SSM Incident Record Update",  
      "eventUpdatedTime": "2021-05-21T18:58:46.443000+00:00",  
      "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-  
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
    },  
    {  
      "eventId": "5abcc812-89c0-b0a8-9437-1c74223d4685",  
      "eventTime": "2021-05-21T18:16:59.149000+00:00",  
      "eventType": "SSM Incident Record Update",  
      "eventUpdatedTime": "2021-05-21T18:16:59.149000+00:00",  
      "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-  
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
    },  
    {  
      "eventId": "06bcc812-8820-405e-4065-8d2b14d29b92",  
      "eventTime": "2021-05-21T18:16:58+00:00",  
      "eventType": "SSM Automation Execution Start Failure for Incident",  
      "eventUpdatedTime": "2021-05-21T18:16:58.689000+00:00",  
      "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-  
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
    },  
    {  
      "eventId": "20bcc812-8a94-4cd7-520c-0ff742111424",  
      "eventTime": "2021-05-21T18:16:57+00:00",  
      "eventType": "SSM Automation Execution Start Failure for Incident",  
      "eventUpdatedTime": "2021-05-21T18:16:57.689000+00:00",  
      "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-  
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
    }  
  ]  
}
```

```

        "eventType": "Custom Event",
        "eventUpdatedTime": "2021-05-21T18:16:59.944000+00:00",
        "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
    },
    {
        "eventId": "c0bcc885-a41d-eb01-b4ab-9d2de193643c",
        "eventTime": "2020-10-01T20:30:00+00:00",
        "eventType": "Custom Event",
        "eventUpdatedTime": "2021-05-21T22:28:26.299000+00:00",
        "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-
record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
    }
]
}

```

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTimelineEvents](#)」の「」を参照してください。

put-resource-policy

次の例は、put-resource-policy を使用する方法を説明しています。

AWS CLI

対応計画とインシデントを共有するには

次のcommand-name例では、レスポンスプランと関連するインシデントを指定されたプリンシパルと共有するリソースポリシーを Example-Response-Plan に追加します。

```

aws ssm-incidents put-resource-policy \
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan" \
  --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"ExampleResourcePolciy\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
\"arn:aws:iam::222233334444:root\"},\"Action\":[\"ssm-incidents:GetResponsePlan
\",\"ssm-incidents:StartIncident\",\"ssm-incidents:UpdateIncidentRecord
\",\"ssm-incidents:GetIncidentRecord\",\"ssm-incidents:CreateTimelineEvent
\",\"ssm-incidents:UpdateTimelineEvent\",\"ssm-incidents:GetTimelineEvent
\",\"ssm-incidents:ListTimelineEvents\",\"ssm-incidents:UpdateRelatedItems

```

```
\",\"ssm-incidents:ListRelatedItems\"),\"Resource\":[\"arn:aws:ssm-incidents:*:111122223333:response-plan/Example-Response-Plan\", \"arn:aws:ssm-incidents:*:111122223333:incident-record/Example-Response-Plan/*\"]}]}"
```

出力:

```
{  
  "policyId": "be8b57191f0371f1c6827341aa3f0a03"  
}
```

詳細については、「[Incident Manager ユーザーガイド](#)」の「[共有連絡先と対応計画の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutResourcePolicy](#)」の「」を参照してください。

start-incident

次の例は、start-incident を使用する方法を説明しています。

AWS CLI

インシデントを開始するには

次のstart-incident例では、指定されたレスポンスプランを使用してインシデントを開始します。

```
aws ssm-incidents start-incident \  
  --response-plan-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

出力:

```
{  
  "incidentRecordArn": "arn:aws:ssm-incidents::682428703967:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
}
```

詳細については、「[Incident Manager ユーザーガイド](#)」の「[Incident creation](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartIncident](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

レスポンスプランにタグを付けるには

次のtag-resource例では、指定されたレスポンスプランに指定されたタグキーと値のペアをタグ付けします。

```
aws ssm-incidents tag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
Response-Plan" \  
  --tags '{"group1":"1"}'
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

レスポンスプランからタグを削除するには

次のuntag-resource例では、レスポンスプランから指定されたタグを削除します。

```
aws ssm-incidents untag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
Response-Plan" \  
  --tag-keys '['group1']'
```

このコマンドでは何も出力されません。

詳細については、「[Incident Manager ユーザーガイド](#)」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUntagResource](#)」の「」を参照してください。

update-deletion-protection

次のコード例は、update-deletion-protection を使用する方法を示しています。

AWS CLI

レプリケーションセットの削除保護を更新するには

次のupdate-deletion-protection例では、レプリケーションセットの最後のリージョンを削除しないように、アカウントの削除保護を更新します。

```
aws ssm-incidents update-deletion-protection \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/  
a2bcc5c9-0f53-8047-7fef-c20749989b40" \  
  --deletion-protected
```

このコマンドでは何も出力されません。

詳細については、[Incident Manager ユーザーガイドの「Incident Manager レプリケーションセットの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateDeletionProtection](#)」の「」を参照してください。

update-incident-record

次のコード例は、update-incident-record を使用する方法を示しています。

AWS CLI

インシデントレコードを更新するには

次のcommand-name例では、指定されたインシデントを解決します。

```
aws ssm-incidents update-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-  
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \  
  --resolved
```

```
--status "RESOLVED"
```

このコマンドでは何も出力されません。

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIncidentRecord](#)」の「」を参照してください。

update-related-items

次の例は、update-related-items を使用する方法を説明しています。

AWS CLI

インシデント関連項目を更新するには

次のupdate-related-item例では、指定されたインシデントレコードから関連項目を削除します。

```
aws ssm-incidents update-related-items \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \  
  --related-items-update '{"itemToRemove": {"type": "OTHER", "value": {"url":  
  "https://console.aws.amazon.com/systems-manager/opsitems/oi-8ef82158e190/workbench?  
region=us-east-1"}}}'
```

このコマンドでは何も出力されません。

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRelatedItems](#)」の「」を参照してください。

update-replication-set

次のコード例は、update-replication-set を使用する方法を示しています。

AWS CLI

レプリケーションセットを更新するには

次のcommand-name例では、us-east-2 リージョンをレプリケーションセットから削除します。

```
aws ssm-incidents update-replication-set \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/  
a2bcc5c9-0f53-8047-7fef-c20749989b40" \  
  --actions '[{"deleteRegionAction": {"regionName": "us-east-2"}}]'
```

このコマンドでは何も出力されません。

詳細については、[Incident Manager ユーザーガイドの「Incident Manager レプリケーションセットの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateReplicationSet](#)」の「」を参照してください。

update-response-plan

次の例は、update-response-plan を使用する方法を説明しています。

AWS CLI

レスポンスプランを更新するには

次のupdate-response-plan例では、指定されたレスポンスプランからチャットチャンネルを削除します。

```
aws ssm-incidents update-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"  
 \  
  --chat-channel '{"empty":{}}'
```

このコマンドでは何も出力されません。

詳細については、[「Incident Manager ユーザーガイド」の「Incident preparation」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResponsePlan](#)」の「」を参照してください。

update-timeline-event

次の例は、update-timeline-event を使用する方法を説明しています。

AWS CLI

タイムラインイベントを更新するには

次のupdate-timeline-event例では、イベントが発生した時刻を更新します。

```
aws ssm-incidents update-timeline-event \  
  --event-id 20bcc812-8a94-4cd7-520c-0ff742111424 \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \  
  --event-time "2021-05-21T18:10:57+00:00"
```

このコマンドでは何も出力されません。

詳細については、[「Incident Manager ユーザーガイド」](#)の「Incident details」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTimelineEvent](#)」の「」を参照してください。

を使用した Incident Manager Contacts の例 AWS CLI

次のコード例は、Incident Manager Contacts AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-page

次のコード例は、accept-page を使用する方法を示しています。

AWS CLI

および エンゲージメント中にページを受け入れるには

次のaccept-page例では、問い合わせチャンネルに送信された承認コードを使用してページを受け入れます。

```
aws ssm-contacts accept-page \  
  --page-id "arn:aws:ssm-contacts:us-east-2:682428703967:page/  
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3" \  
  --accept-type READ \  
  --accept-code 425440
```

このコマンドは出力を生成しません

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptPage](#)」の「」を参照してください。

activate-contact-channel

次の例は、activate-contact-channel を使用する方法を説明しています。

AWS CLI

問い合わせの問い合わせチャンネルをアクティブ化する

次のactivate-contact-channel例では、問い合わせチャンネルをアクティブ化し、インシデントの一部として使用できます。

```
aws ssm-contacts activate-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d" \  
  --activation-code "466136"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ActivateContactChannel](#)」の「」を参照してください。

command-name

次のコード例は、command-name を使用する方法を示しています。

AWS CLI

連絡先を削除するには

次のcommand-name例では、連絡先を削除します。問い合わせを参照するエスカレーションプランから問い合わせにアクセスできなくなります。

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:682428703967:contact/alejr"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の[「問い合わせ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CommandName](#)」の「」を参照してください。

create-contact-channel

次のコード例は、create-contact-channel を使用する方法を示しています。

AWS CLI

問い合わせチャネルを作成するには

問い合わせ Akua Mansa SMSの種類の問い合わせチャネルを作成します。問い合わせチャネルは、タイプ SMS、EMAIL、または で作成できますVOICE。

```
aws ssm-contacts create-contact-channel \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --name "akuas sms-test" \  
  --type SMS \  
  --delivery-address '{"SimpleAddress": "+15005550199}"'
```

出力:

```
{  
  "ContactChannelArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact-  
channel/akuam/02f506b9-ea5d-4764-af89-2daa793ff024"
```

```
}
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateContactChannel](#)」の「」を参照してください。

create-contact

次の例は、create-contact を使用する方法を説明しています。

AWS CLI

連絡先を作成するには

次のcreate-contact例では、空白のプランを使用して、環境でコンタクトを作成します。プランは、問い合わせチャネルの作成後に更新できます。このコマンドARNの出力で コマンドを使用します create-contact-channel。この問い合わせの問い合わせチャネルを作成したら、update-contact を使用してプランを更新します。

```
aws ssm-contacts create-contact \  
  --alias "akuam" \  
  --display-name "Akua Mansa" \  
  --type PERSONAL \  
  --plan '{"Stages": []}'
```

出力:

```
{  
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"  
}
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateContact](#)」の「」を参照してください。

deactivate-contact-channel

次のコード例は、deactivate-contact-channel を使用する方法を示しています。

AWS CLI

問い合わせチャンネルを非アクティブ化するには

次のdeactivate-contact-channel例では、問い合わせチャンネルを非アクティブ化します。問い合わせチャンネルを非アクティブ化すると、インシデント中に問い合わせチャンネルがページングされなくなります。activate-contact-channel コマンドを使用して、いつでも問い合わせチャンネルを再アクティブ化することもできます。

```
aws ssm-contacts deactivate-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeactivateContactChannel](#)」の「」を参照してください。

delete-contact-channel

次の例は、delete-contact-channel を使用する方法を説明しています。

AWS CLI

問い合わせチャンネルを削除するには

次のdelete-contact-channel例では、問い合わせチャンネルを削除します。問い合わせチャンネルを削除すると、インシデント中に問い合わせチャンネルがページングされることはありません。

```
aws ssm-contacts delete-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-  
channel/akuam/13149bad-52ee-45ea-ae1e-45857f78f9b2"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteContactChannel](#)」の「」を参照してください。

delete-contact

次のコード例は、delete-contact を使用する方法を示しています。

AWS CLI

連絡先を削除するには

次のdelete-contact例では、連絡先を削除します。問い合わせを参照するエスカレーションプランから問い合わせにアクセスできなくなります。

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/alejr"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteContact](#)」の「」を参照してください。

describe-engagement

次のコード例は、describe-engagement を使用する方法を示しています。

AWS CLI

エンゲージメントの詳細を記述するには

次のdescribe-engagement例では、問い合わせまたはエスカレーションプランへのエンゲージメントの詳細を一覧表示します。件名とコンテンツは問い合わせチャンネルに送信されます。

```
aws ssm-contacts describe-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

出力:

```
{  
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/  
example_escalation",
```

```

    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356",
    "Sender": "cli",
    "Subject": "cli-test",
    "Content": "Testing engagements via CLI",
    "PublicSubject": "cli-test",
    "PublicContent": "Testing engagements va CLI",
    "StartTime": "2021-05-18T18:25:41.151000+00:00"
}

```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEngagement](#)」の「」を参照してください。

describe-page

次のコード例は、describe-page を使用する方法を示しています。

AWS CLI

問い合わせチャンネルにページの詳細を一覧表示するには

次のdescribe-page例では、問い合わせチャンネルへのページの詳細を一覧表示します。このページには、提供された件名とコンテンツが含まれます。

```

aws ssm-contacts describe-page \
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93"

```

出力:

```

{
  "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93",
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0",
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
  "Sender": "cli",
  "Subject": "cli-test",
  "Content": "Testing engagements via CLI",
  "PublicSubject": "cli-test",

```

```
"PublicContent": "Testing engagements va CLI",
"SentTime": "2021-05-18T18:43:29.301000+00:00",
"ReadTime": "2021-05-18T18:43:55.708000+00:00",
"DeliveryTime": "2021-05-18T18:43:55.265000+00:00"
}
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePage](#)」の「」を参照してください。

get-contact-channel

次のコード例は、get-contact-channel を使用する方法を示しています。

AWS CLI

問い合わせチャネルの詳細を一覧表示するには

次のget-contact-channel例では、問い合わせチャネルの詳細を一覧表示します。

```
aws ssm-contacts get-contact-channel \
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d"
```

出力:

```
{
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
  "ContactChannelArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
  "Name": "akuas sms",
  "Type": "SMS",
  "DeliveryAddress": {
    "SimpleAddress": "+15005550199"
  },
  "ActivationStatus": "ACTIVATED"
}
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetContactChannel](#)」の「」を参照してください。

get-contact-policy

次の例は、get-contact-policy を使用する方法を説明しています。

AWS CLI

問い合わせのリソースポリシーを一覧表示するには

次のget-contact-policy例では、指定された問い合わせに関連付けられているリソースポリシーを一覧表示します。

```
aws ssm-contacts get-contact-policy \
  --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

出力:

```
{
  "ContactArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam",
  "Policy": "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Sid\": \"SharePolicyForDocumentationDralia\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"222233334444\"}, \"Action\": [\"ssm-contacts:GetContact\", \"ssm-contacts:StartEngagement\", \"ssm-contacts:DescribeEngagement\", \"ssm-contacts:ListPagesByEngagement\", \"ssm-contacts:StopEngagement\"], \"Resource\": [\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\", \"arn:aws:ssm-contacts:*:111122223333:engagement/akuam/*\"]}]}"
}
```

詳細については、「Incident Manager ユーザーガイド」の[「共有連絡先と対応計画の操作」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetContactPolicy](#)」の「」を参照してください。

get-contact

次のコード例は、get-contact を使用する方法を示しています。

AWS CLI

例 1: 問い合わせプランを記述するには

次のget-contact例では、問い合わせについて説明します。

```
aws ssm-contacts get-contact \  
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

出力:

```
{  
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",  
  "Alias": "akuam",  
  "DisplayName": "Akua Mansa",  
  "Type": "PERSONAL",  
  "Plan": {  
    "Stages": [  
      {  
        "DurationInMinutes": 5,  
        "Targets": [  
          {  
            "ChannelTargetInfo": {  
              "ContactChannelId": "arn:aws:ssm-contacts:us-  
east-2:111122223333:contact-channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65",  
              "RetryIntervalInMinutes": 1  
            }  
          }  
        ]  
      },  
      {  
        "DurationInMinutes": 5,  
        "Targets": [  
          {  
            "ChannelTargetInfo": {  
              "ContactChannelId": "arn:aws:ssm-contacts:us-  
east-2:111122223333:contact-channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad",  
              "RetryIntervalInMinutes": 1  
            }  
          }  
        ]  
      },  
      {  
        "DurationInMinutes": 5,  
        "Targets": [  
          {  
            "ChannelTargetInfo": {  
              "ContactChannelId": "arn:aws:ssm-contacts:us-  
east-2:111122223333:contact-channel/akuam/77d4f447-f619-4954-afff-85551e369c2a",  
              "RetryIntervalInMinutes": 1  
            }  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
        "RetryIntervalInMinutes": 1
      }
    ]
  }
}
```

例 2: エスカレーションプランを記述するには

次のget-contact例では、エスカレーションプランについて説明します。

```
aws ssm-contacts get-contact \
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation"
```

出力:

```
{
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
  "Alias": "example_escalation",
  "DisplayName": "Example Escalation",
  "Type": "ESCALATION",
  "Plan": {
    "Stages": [
      {
        "DurationInMinutes": 5,
        "Targets": [
          {
            "ContactTargetInfo": {
              "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/akuam",
              "IsEssential": true
            }
          }
        ]
      },
      {
        "DurationInMinutes": 5,
        "Targets": [
          {
```

```
        "ContactTargetInfo": {
            "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/alejr",
            "IsEssential": false
        }
    ],
    {
        "DurationInMinutes": 0,
        "Targets": [
            {
                "ContactTargetInfo": {
                    "ContactId": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact/anasi",
                    "IsEssential": false
                }
            }
        ]
    }
]
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetContact](#)」の「」を参照してください。

list-contact-channels

次の例は、list-contact-channels を使用する方法を説明しています。

AWS CLI

問い合わせの問い合わせチャンネルを一覧表示するには

次のlist-contact-channels例は、指定された問い合わせで使用可能な問い合わせチャンネルを一覧表示します。

```
aws ssm-contacts list-contact-channels \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

出力:

```
{
  [
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
      "Name": "akuas email",
      "Type": "EMAIL",
      "DeliveryAddress": {
        "SimpleAddress": "akuam@example.com"
      },
      "ActivationStatus": "NOT_ACTIVATED"
    },
    {
      "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
      "Name": "akuas sms",
      "Type": "SMS",
      "DeliveryAddress": {
        "SimpleAddress": "+15005550100"
      },
      "ActivationStatus": "ACTIVATED"
    }
  ]
}
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListContactChannels](#)」の「」を参照してください。

list-contacts

次のコード例は、list-contacts を使用する方法を示しています。

AWS CLI

すべてのエスカレーションプランと連絡先を一覧表示するには

次のlist-contacts例では、アカウントの問い合わせとエスカレーションプランを一覧表示します。

```
aws ssm-contacts list-contacts
```

出力:

```
{
  "Contacts": [
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
      "Alias": "akuam",
      "DisplayName": "Akua Mansa",
      "Type": "PERSONAL"
    },
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
alejr",
      "Alias": "alejr",
      "DisplayName": "Alejandro Rosalez",
      "Type": "PERSONAL"
    },
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
anasi",
      "Alias": "anasi",
      "DisplayName": "Ana Carolina Silva",
      "Type": "PERSONAL"
    },
    {
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
      "Alias": "example_escalation",
      "DisplayName": "Example Escalation",
      "Type": "ESCALATION"
    }
  ]
}
```

詳細については、「[Incident Manager ユーザーガイド](#)」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListContacts](#)」の「」を参照してください。

list-engagements

次の例は、list-engagements を使用する方法を説明しています。

AWS CLI

すべてのエンゲージメントを一覧表示するには

次のlist-engagements例では、エスカレーションプランと問い合わせへのエンゲージメントを一覧表示します。1つのインシデントのエンゲージメントを一覧表示することもできます。

```
aws ssm-contacts list-engagements
```

出力:

```
{
  "Engagements": [
    {
      "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/akuam/91792571-0b53-4821-9f73-d25d13d9e529",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
      "Sender": "cli",
      "StartTime": "2021-05-18T20:37:50.300000+00:00"
    },
    {
      "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
      "Sender": "cli",
      "StartTime": "2021-05-18T18:40:26.666000+00:00"
    },
    {
      "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356",
      "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/example_escalation",

```

```

    "Sender": "cli",
    "StartTime": "2021-05-18T18:25:41.151000+00:00"
  },
  {
    "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f",
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
    "Sender": "cli",
    "StartTime": "2021-05-18T18:20:58.093000+00:00"
  }
]
}

```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEngagements](#)」の「」を参照してください。

list-page-receipts

次のコード例は、list-page-receipts を使用する方法を示しています。

AWS CLI

ページ受信を一覧表示するには

次のcommand-name例は、問い合わせによってページが受信されたかどうかを示しています。

```

aws ssm-contacts list-page-receipts \
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3"

```

出力:

```

{
  "Receipts": [
    {
      "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
      "ReceiptType": "DELIVERED",
      "ReceiptInfo": "425440",
    }
  ]
}

```

```

    "ReceiptTime": "2021-05-18T20:42:57.485000+00:00"
  },
  {
    "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
    "ReceiptType": "READ",
    "ReceiptInfo": "425440",
    "ReceiptTime": "2021-05-18T20:42:57.907000+00:00"
  },
  {
    "ContactChannelArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:contact-channel/akuam/fc7405c4-46b2-48b7-87b2-93e2f225b90d",
    "ReceiptType": "SENT",
    "ReceiptInfo": "SM6656c19132f1465f9c9c1123a5dde7c9",
    "ReceiptTime": "2021-05-18T20:40:52.962000+00:00"
  }
]
}

```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPageReceipts](#)」の「」を参照してください。

list-pages-by-contact

次のコード例は、list-pages-by-contact を使用する方法を示しています。

AWS CLI

連絡先別にページを一覧表示するには

次のlist-pages-by-contact例では、指定された連絡先へのすべてのページを一覧表示します。

```

aws ssm-contacts list-pages-by-contact \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"

```

出力:

```

{
  "Pages": [

```

```
{
  "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
ad0052bd-e606-498a-861b-25726292eb93",
  "EngagementArn": "arn:aws:ssm-contacts:us-
east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
  "Sender": "cli",
  "SentTime": "2021-05-18T18:43:29.301000+00:00",
  "DeliveryTime": "2021-05-18T18:43:55.265000+00:00",
  "ReadTime": "2021-05-18T18:43:55.708000+00:00"
}
]
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPagesByContact](#)」の「」を参照してください。

list-pages-by-engagement

次のコード例は、list-pages-by-engagement を使用する方法を示しています。

AWS CLI

エンゲージメントから開始された問い合わせチャネルのページを一覧表示するには。

次のlist-pages-by-engagement例では、定義されたエンゲージメントプランのエンゲージメント中に発生したページを一覧表示します。

```
aws ssm-contacts list-pages-by-engagement \
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0"
```

出力:

```
{
  "Pages": [
    {
      "PageArn": "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/
ad0052bd-e606-498a-861b-25726292eb93",
```

```
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/akuam/78a29753-3674-4ac5-9f83-0468563567f0",
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam",
    "Sender": "cli",
    "SentTime": "2021-05-18T18:40:27.245000+00:00"
  }
]
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPagesByEngagement](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

連絡先のタグを一覧表示するには

次のlist-tags-for-resource例では、指定された問い合わせのタグを一覧表示します。

```
aws ssm-contacts list-tags-for-resource \
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

出力:

```
{
  "Tags": [
    {
      "Key": "group1",
      "Value": "1"
    }
  ]
}
```

詳細については、「Incident Manager ユーザーガイド」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-contact-policy

次の例は、put-contact-policy を使用する方法を説明しています。

AWS CLI

問い合わせとエンゲージメントを共有するには

次のput-contact-policy例では、問い合わせおよび関連するエンゲージメントをプリンシパルと共有するリソースポリシーを問い合わせ Akua に追加します。

```
aws ssm-contacts put-contact-policy \  
  --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\  
  \"ExampleResourcePolicy\",\"Action\":[\"ssm-contacts:GetContact\",\"ssm-\  
  contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-\  
  contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],\  
  \"Principal\":{\"AWS\":\"222233334444\"},\"Effect\":\"Allow\",\"Resource\  
  \":[\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\",\"arn:aws:ssm-\  
  contacts:*:111122223333:engagement/akuam/*\"]}]}"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[共有連絡先と対応計画の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutContactPolicy](#)」の「」を参照してください。

send-activation-code

次のコード例は、send-activation-code を使用する方法を示しています。

AWS CLI

アクティベーションコードを送信するには

次のsend-activation-code例は、指定された問い合わせチャンネルにアクティベーションコードとメッセージを送信します。

```
aws ssm-contacts send-activation-code \  
  --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --channel-id "channel-id" \  
  --message "message" \  
  --activation-code "activation-code" \  
  --policy "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\  
  \"ExampleResourcePolicy\",\"Action\":[\"ssm-contacts:GetContact\",\"ssm-\  
  contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-\  
  contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],\  
  \"Principal\":{\"AWS\":\"222233334444\"},\"Effect\":\"Allow\",\"Resource\  
  \":[\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\",\"arn:aws:ssm-\  
  contacts:*:111122223333:engagement/akuam/*\"]}]}"
```

```
--contact-channel-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact-channel/akuam/8ddae2d1-12c8-4e45-b852-c8587266c400"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendActivationCode](#)」の「」を参照してください。

start-engagement

次のコード例は、start-engagement を使用する方法を示しています。

AWS CLI

例 1: 問い合わせの問い合わせチャンネルをページングするには

次のstart-engagementページでは、連絡先の問い合わせチャンネルについて説明します。送信者、サブジェクト、パブリックサブジェクト、パブリックコンテンツにはすべてフィールドがありません。Incident Manager は、提供されたチャンネルVOICEまたはEMAIL問い合わせチャンネルに件名とコンテンツを送信します。Incident Manager は、提供されたSMS問い合わせチャンネルにパブリックサブジェクトとパブリックコンテンツを送信します。送信者は、エンゲージメントを開始したユーザーを追跡するために使用されます。

```
aws ssm-contacts start-engagement \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \  
  --sender "cli" \  
  --subject "cli-test" \  
  --content "Testing engagements via CLI" \  
  --public-subject "cli-test" \  
  --public-content "Testing engagements va CLI"
```

出力:

```
{  
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f"  
}
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

例 2: 指定されたエスカレーションプランの連絡先をページングするには。

以下は、エスカレーションプランを通じて問い合わせをstart-engagement関与させます。各問い合わせは、エンゲージメントプランに従ってページ分割されます。

```
aws ssm-contacts start-engagement \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/  
example_escalation" \  
  --sender "cli" \  
  --subject "cli-test" \  
  --content "Testing engagements via CLI" \  
  --public-subject "cli-test" \  
  --public-content "Testing engagements va CLI"
```

出力:

```
{  
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"  
}
```

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartEngagement](#)」の「」を参照してください。

stop-engagement

次の例は、stop-engagement を使用する方法を説明しています。

AWS CLI

エンゲージメントを停止するには

次のstop-engagement例では、エンゲージメントが追加の問い合わせと問い合わせチャンネルをページングするのを停止します。

```
aws ssm-contacts stop-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopEngagement](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

連絡先にタグを付けるには

次のtag-resource例では、指定された連絡先に指定されたタグキー値ペアをタグ付けします。

```
aws ssm-contacts tag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tags '[{"Key":"group1","Value":"1"}]'
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

連絡先からタグを削除するには

次のuntag-resource例では、指定された問い合わせから group1 タグを削除します。

```
aws ssm-contacts untag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tag-keys "group1"
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-contact-channel

次の例は、update-contact-channel を使用する方法を説明しています。

AWS CLI

問い合わせチャネルを更新するには

次のupdate-contact-channel例では、問い合わせチャネルの名前と配信アドレスを更新します。

```
aws ssm-contacts update-contact-channel \  
  --contact-channel-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad" \  
  --name "akuas voice channel" \  
  --delivery-address '{"SimpleAddress": "+15005550198"}'
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateContactChannel](#)」の「」を参照してください。

update-contact

次のコード例は、update-contact を使用する方法を示しています。

AWS CLI

問い合わせのエンゲージメントプランを更新するには

次のupdate-contact例では、問い合わせ Akua のエンゲージメントプランを更新して、3 種類の問い合わせチャネルを含めます。これは、Akua の問い合わせチャネルを作成した後に行われます。

```
aws ssm-contacts update-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \  
  --plan '{"Stages": [{"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":  
  {"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65", "RetryIntervalInMinutes":  
1 }]}], {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":  
{"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/  
akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad", "RetryIntervalInMinutes": 1}]}}],  
{"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo": {"ContactChannelId":  
"arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/77d4f447-  
f619-4954-afff-85551e369c2a", "RetryIntervalInMinutes": 1 }]}]}'
```

このコマンドでは何も出力されません。

詳細については、「Incident Manager ユーザーガイド」の「[問い合わせ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateContact](#)」の「」を参照してください。

を使用した Amazon Inspector の例 AWS CLI

次のコード例は、Amazon Inspector AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-attributes-to-findings

次の例は、add-attributes-to-findings を使用する方法を説明しています。

AWS CLI

検出結果に属性を追加するには

次のadd-attribute-to-findingコマンドは、のキーExampleと の値を持つ属性exampleを、ARNの を持つ検出結果に割り当てますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU。

```
aws inspector add-attributes-to-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attributes key=Example,value=example
```

出力:

```
{
  "failedItems": {}
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector の検出結果」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddAttributesToFindings](#)」の「」を参照してください。

create-assessment-target

次のコード例は、create-assessment-target を使用する方法を示しています。

AWS CLI

評価ターゲットを作成するには

次のcreate-assessment-targetコマンドは、リソースグループExampleAssessmentTargetを使用して という名前の評価ターゲットを ARNの で作成しますarn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv。

```
aws inspector create-assessment-target --assessment-target-name ExampleAssessmentTarget --resource-group-arn arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv
```

出力:

```
{
  "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX"
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価ターゲット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateAssessmentTarget](#)」の「」を参照してください。

create-assessment-template

次のコード例は、create-assessment-template を使用する方法を示しています。

AWS CLI

評価ターゲットテンプレートを作成するには

次のcreate-assessment-templateコマンドは、 の評価ターゲット ExampleAssessmentTemplate という評価テンプレートを ARN の で作成します。arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX。

```
aws inspector create-assessment-template --assessment-target-
arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX --assessment-
template-name ExampleAssessmentTemplate --duration-in-seconds 180 --rules-package-
arns arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p --user-
attributes-for-findings key=ExampleTag,value=examplevalue
```

出力:

```
{
  "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T"
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAssessmentTemplate](#)」の「」を参照してください。

create-filter

次のコード例は、create-filter を使用する方法を示しています。

AWS CLI

フィルターを作成するには

次のcreate-filter例では、ECRインスタンスタイプの検出結果を省略する抑制ルールを作成します。

```
aws inspector2 create-filter \  
  --name "ExampleSuppressionRuleECR" \  
  --description "This suppression rule omits ECR instance type findings" \  
  --action SUPPRESS \  
  --filter-criteria 'resourceType=[{comparison="EQUALS",  
value="AWS_ECR_INSTANCE"}]'
```

出力:

```
{  
  "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/  
EXAMPLE444444444444"  
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector の検出結果のフィルタリング](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFilter](#)」の「」を参照してください。

create-findings-report

次のコード例は、create-findings-report を使用する方法を示しています。

AWS CLI

検出結果レポートを作成するには

次のcreate-findings-report例では、検出結果レポートを作成します。

```
aws inspector2 create-findings-report \  
  --report-format CSV \  
  --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-  
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE33333 \  
  --filter-criteria '{"ecrImageRepositoryName":  
[{"comparison":"EQUALS","value":"debian"}]}'
```

出力:

```
{  
  "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector での検出結果の管理](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFindingsReport](#)」の「」を参照してください。

create-resource-group

次のコード例は、create-resource-group を使用する方法を示しています。

AWS CLI

リソースグループを作成するには

次のcreate-resource-groupコマンドは、 のタグキーNameと の値を使用してリソースグループを作成しますexample。

```
aws inspector create-resource-group --resource-group-tags key=Name,value=example
```

出力:

```
{  
  "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-  
AB6DMKnv"
```



```
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価ターゲット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateResourceGroup](#)」の「」を参照してください。

create-sbom-export

次のコード例は、create-sbom-export を使用する方法を示しています。

AWS CLI

ソフトウェア部品表 (SBOM) レポートを作成するには

次のcreate-sbom-export例では、ソフトウェア部品表 (SBOM) レポートを作成します。

```
aws inspector2 create-sbom-export \  
  --report-format SPDX_2_3 \  
  --resource-filter-criteria  
  'ecrRepositoryName=[{comparison="EQUALS",value="debian"}]' \  
  --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-  
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE33333
```

出力:

```
{  
  "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector SBOMsを使用したエクスポート](#)」を参照してください。 Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSbomExport](#)」の「」を参照してください。

delete-assessment-run

次のコード例は、delete-assessment-run を使用する方法を示しています。

AWS CLI

評価実行を削除するには

次のdelete-assessment-runコマンドは、ARNの を使用して評価実行を削除しますarn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe。

```
aws inspector delete-assessment-run --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe
```

詳細については、Amazon Inspector ガイドのAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAssessmentRun](#)」の「」を参照してください。

delete-assessment-target

次の例は、delete-assessment-target を使用する方法を説明しています。

AWS CLI

評価ターゲットを削除するには

次のdelete-assessment-targetコマンドは、ARNの を使用して評価ターゲットを削除しますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq。

```
aws inspector delete-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

詳細については、Amazon Inspector ガイド」のAmazon Inspector 評価ターゲット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAssessmentTarget](#)」の「」を参照してください。

delete-assessment-template

次のコード例は、delete-assessment-template を使用する方法を示しています。

AWS CLI

評価テンプレートを削除するには

次のdelete-assessment-templateコマンドは、ARNの を使用して評価テンプレートを削除しますarn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T。

```
aws inspector delete-assessment-template --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

詳細については、「Amazon Inspector ガイド」のAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAssessmentTemplate](#)」の「」を参照してください。

delete-filter

次の例は、delete-filter を使用する方法を説明しています。

AWS CLI

フィルターを削除するには

次のdelete-filter例では、フィルターを削除します。

```
aws inspector2 delete-filter \  
  --arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE444444444"
```

出力:

```
{  
  "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE444444444"  
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector の検出結果のフィルタリング](#)」を参照してください。 Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFilter](#)」の「」を参照してください。

describe-assessment-runs

次の例は、describe-assessment-runs を使用する方法を説明しています。

AWS CLI

評価実行を記述するには

次のdescribe-assessment-runコマンドは、ARNの を使用した評価実行について説明しますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE。

```
aws inspector describe-assessment-runs --assessment-run-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

出力:

```
{
  "assessmentRuns": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
      "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw",
      "completedAt": 1458680301.4,
      "createdAt": 1458680170.035,
      "dataCollected": true,
      "durationInSeconds": 3600,
      "name": "Run 1 for ExampleAssessmentTemplate",
      "notifications": [],
      "rulesPackageArns": [
        "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
      ],
      "startedAt": 1458680170.161,
      "state": "COMPLETED",
      "stateChangedAt": 1458680301.4,
      "stateChanges": [
        {
```

```
        "state": "CREATED",
        "stateChangedAt": 1458680170.035
    },
    {
        "state": "START_DATA_COLLECTION_PENDING",
        "stateChangedAt": 1458680170.065
    },
    {
        "state": "START_DATA_COLLECTION_IN_PROGRESS",
        "stateChangedAt": 1458680170.096
    },
    {
        "state": "COLLECTING_DATA",
        "stateChangedAt": 1458680170.161
    },
    {
        "state": "STOP_DATA_COLLECTION_PENDING",
        "stateChangedAt": 1458680239.883
    },
    {
        "state": "DATA_COLLECTED",
        "stateChangedAt": 1458680299.847
    },
    {
        "state": "EVALUATING_RULES",
        "stateChangedAt": 1458680300.099
    },
    {
        "state": "COMPLETED",
        "stateChangedAt": 1458680301.4
    }
],
"userAttributesForFindings": []
}
],
"failedItems": {}
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAssessmentRuns](#)」の「」を参照してください。

describe-assessment-targets

次のコード例は、describe-assessment-targets を使用する方法を示しています。

AWS CLI

評価ターゲットを記述するには

次のdescribe-assessment-targetsコマンドは、 の評価ターゲットを ARNの で説明していますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq。

```
aws inspector describe-assessment-targets --assessment-target-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

出力:

```
{
  "assessmentTargets": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq",
      "createdAt": 1458074191.459,
      "name": "ExampleAssessmentTarget",
      "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI",
      "updatedAt": 1458074191.459
    }
  ],
  "failedItems": {}
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価ターゲット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAssessmentTargets](#)」の「」を参照してください。

describe-assessment-templates

次のコード例は、describe-assessment-templates を使用する方法を示しています。

AWS CLI

評価テンプレートを記述するには

次のdescribe-assessment-templatesコマンドは、 の評価テンプレートを ARN ので説明していますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw。

```
aws inspector describe-assessment-templates --assessment-template-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw
```

出力:

```
{
  "assessmentTemplates": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw",
      "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq",
      "createdAt": 1458074191.844,
      "durationInSeconds": 3600,
      "name": "ExampleAssessmentTemplate",
      "rulesPackageArns": [
        "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-X1KXtawP"
      ],
      "userAttributesForFindings": []
    }
  ],
  "failedItems": {}
}
```

詳細については、「Amazon Inspector ガイド」のAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAssessmentTemplates](#)」の「」を参照してください。

describe-cross-account-access-role

次のコード例は、describe-cross-account-access-role を使用する方法を示しています。

AWS CLI

クロスアカウントアクセスロールを記述するには

次のdescribe-cross-account-access-roleコマンドは、Amazon Inspector が AWS アカウントにアクセスできるようにするIAMロールについて説明します。

```
aws inspector describe-cross-account-access-role
```

出力:

```
{
  "registeredAt": 1458069182.826,
  "roleArn": "arn:aws:iam::123456789012:role/inspector",
  "valid": true
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector のセットアップ」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeCrossAccountAccessRole](#)」の「」を参照してください。

describe-findings

次の例は、describe-findings を使用する方法を説明しています。

AWS CLI

検出結果を記述するには

次のdescribe-findingsコマンドは、ARN のを使用して検出結果について説明しますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4。

```
aws inspector describe-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4
```

出力:

```
{
  "failedItems": {},
}
```



```
"findings": [
  {
    "arn": "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
    "assetAttributes": {
      "ipv4Addresses": [],
      "schemaVersion": 1
    },
    "assetType": "ec2-instance",
    "attributes": [],
    "confidence": 10,
    "createdAt": 1458680301.37,
    "description": "Amazon Inspector did not find any potential security
issues during this assessment.",
    "indicatorOfCompromise": false,
    "numericSeverity": 0,
    "recommendation": "No remediation needed.",
    "schemaVersion": 1,
    "service": "Inspector",
    "serviceAttributes": {
      "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
      "rulesPackageArn": "arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-X1KXtawP",
      "schemaVersion": 1
    },
    "severity": "Informational",
    "title": "No potential security issues found",
    "updatedAt": 1458680301.37,
    "userAttributes": []
  }
]
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector の検出結果」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeFindings](#)」の「」を参照してください。

describe-resource-groups

次のコード例は、describe-resource-groups を使用する方法を示しています。

AWS CLI

リソースグループを記述するには

次のdescribe-resource-groupsコマンドは、ARNの を持つリソースグループについて説明しますarn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI。

```
aws inspector describe-resource-groups --resource-group-arns arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI
```

出力:

```
{
  "failedItems": {},
  "resourceGroups": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI",
      "createdAt": 1458074191.098,
      "tags": [
        {
          "key": "Name",
          "value": "example"
        }
      ]
    }
  ]
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector Assessment Targets」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeResourceGroups](#)」の「」を参照してください。

describe-rules-packages

次の例は、describe-rules-packages を使用する方法を説明しています。

AWS CLI

ルールパッケージを記述するには

次のdescribe-rules-packagesコマンドは、ARN のを持つルールパッケージについて説明しますarn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p。

```
aws inspector describe-rules-packages --rules-package-arns arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p
```

出力:

```
{
  "failedItems": {},
  "rulesPackages": [
    {
      "arn": "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p",
      "description": "The rules in this package help verify whether the EC2 instances in your application are exposed to Common Vulnerabilities and Exposures (CVEs). Attacks can exploit unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of your service or data. The CVE system provides a reference for publicly known information security vulnerabilities and exposures. For more information, see [https://cve.mitre.org/](https://cve.mitre.org/). If a particular CVE appears in one of the produced Findings at the end of a completed Inspector assessment, you can search [https://cve.mitre.org/](https://cve.mitre.org/) using the CVE's ID (for example, \"CVE-2009-0021\") to find detailed information about this CVE, its severity, and how to mitigate it. ",
      "name": "Common Vulnerabilities and Exposures",
      "provider": "Amazon Web Services, Inc.",
      "version": "1.1"
    }
  ]
}
```

詳細については、「Amazon Inspector ガイド」のAmazon Inspector ルールパッケージとルール」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRulesPackages](#)」の「」を参照してください。

get-configuration

次のコード例は、get-configuration を使用する方法を示しています。

AWS CLI

Inspector スキャンの設定設定を取得するには

次のget-configuration例では、Inspector スキャンの設定設定を取得します。

```
aws inspector2 get-configuration
```

出力:

```
{
  "ec2Configuration": {
    "scanModeState": {
      "scanMode": "EC2_HYBRID",
      "scanModeStatus": "SUCCESS"
    }
  },
  "ecrConfiguration": {
    "rescanDurationState": {
      "pullDateRescanDuration": "DAYS_90",
      "rescanDuration": "DAYS_30",
      "status": "SUCCESS",
      "updatedAt": "2024-05-14T21:16:20.237000+00:00"
    }
  }
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector による自動リソーススキャン](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConfiguration](#)」の「」を参照してください。

get-telemetry-metadata

次のコード例は、get-telemetry-metadata を使用する方法を示しています。

AWS CLI

テレメトリメタデータを取得するには

次のget-telemetry-metadataコマンドは、ARNの を使用して評価実行用に収集されたデータに関する情報を生成しますarn:aws:inspector:us-

west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE。

```
aws inspector get-telemetry-metadata --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

出力:

```
{
  "telemetryMetadata": [
    {
      "count": 2,
      "dataSize": 345,
      "messageType": "InspectorDuplicateProcess"
    },
    {
      "count": 3,
      "dataSize": 255,
      "messageType": "InspectorTimeEventMsg"
    },
    {
      "count": 4,
      "dataSize": 1082,
      "messageType": "InspectorNetworkInterface"
    },
    {
      "count": 2,
      "dataSize": 349,
      "messageType": "InspectorDnsEntry"
    },
    {
      "count": 11,
      "dataSize": 2514,
      "messageType": "InspectorDirectoryInfoMsg"
    },
    {
      "count": 1,
      "dataSize": 179,
      "messageType": "InspectorTcpV6ListeningPort"
    },
    {
      "count": 101,
      "dataSize": 10949,
```

```
    "messageType": "InspectorTerminal"
  },
  {
    "count": 26,
    "dataSize": 5916,
    "messageType": "InspectorUser"
  },
  {
    "count": 282,
    "dataSize": 32148,
    "messageType": "InspectorDynamicallyLoadedCodeModule"
  },
  {
    "count": 18,
    "dataSize": 10172,
    "messageType": "InspectorCreateProcess"
  },
  {
    "count": 3,
    "dataSize": 8001,
    "messageType": "InspectorProcessPerformance"
  },
  {
    "count": 1,
    "dataSize": 360,
    "messageType": "InspectorOperatingSystem"
  },
  {
    "count": 6,
    "dataSize": 546,
    "messageType": "InspectorStopProcess"
  },
  {
    "count": 1,
    "dataSize": 1553,
    "messageType": "InspectorInstanceMetaData"
  },
  {
    "count": 2,
    "dataSize": 434,
    "messageType": "InspectorTcpV4Connection"
  },
  {
    "count": 474,
```

```
    "dataSize": 2960322,
    "messageType": "InspectorPackageInfo"
  },
  {
    "count": 3,
    "dataSize": 2235,
    "messageType": "InspectorSystemPerformance"
  },
  {
    "count": 105,
    "dataSize": 46048,
    "messageType": "InspectorCodeModule"
  },
  {
    "count": 1,
    "dataSize": 182,
    "messageType": "InspectorUdpV6ListeningPort"
  },
  {
    "count": 2,
    "dataSize": 371,
    "messageType": "InspectorUdpV4ListeningPort"
  },
  {
    "count": 18,
    "dataSize": 8362,
    "messageType": "InspectorKernelModule"
  },
  {
    "count": 29,
    "dataSize": 48788,
    "messageType": "InspectorConfigurationInfo"
  },
  {
    "count": 1,
    "dataSize": 79,
    "messageType": "InspectorMonitoringStart"
  },
  {
    "count": 5,
    "dataSize": 0,
    "messageType": "InspectorSplitMsgBegin"
  },
  {
```

```
    "count": 51,  
    "dataSize": 4593,  
    "messageType": "InspectorGroup"  
  },  
  {  
    "count": 1,  
    "dataSize": 184,  
    "messageType": "InspectorTcpV4ListeningPort"  
  },  
  {  
    "count": 1159,  
    "dataSize": 3146579,  
    "messageType": "Total"  
  },  
  {  
    "count": 5,  
    "dataSize": 0,  
    "messageType": "InspectorSplitMsgEnd"  
  },  
  {  
    "count": 1,  
    "dataSize": 612,  
    "messageType": "InspectorLoadImageInProgress"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTelemetryMetadata](#)」の「」を参照してください。

list-account-permissions

次の例は、list-account-permissions を使用する方法を説明しています。

AWS CLI

アカウントアクセス許可を一覧表示するには

次のlist-account-permissions例では、アカウントのアクセス許可を一覧表示します。

```
aws inspector2 list-account-permissions
```

出力:


```
{
  "permissions": [
    {
      "operation": "ENABLE_SCANNING",
      "service": "ECR"
    },
    {
      "operation": "DISABLE_SCANNING",
      "service": "ECR"
    },
    {
      "operation": "ENABLE_REPOSITORY",
      "service": "ECR"
    },
    {
      "operation": "DISABLE_REPOSITORY",
      "service": "ECR"
    },
    {
      "operation": "ENABLE_SCANNING",
      "service": "EC2"
    },
    {
      "operation": "DISABLE_SCANNING",
      "service": "EC2"
    },
    {
      "operation": "ENABLE_SCANNING",
      "service": "LAMBDA"
    },
    {
      "operation": "DISABLE_SCANNING",
      "service": "LAMBDA"
    }
  ]
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector のアイデンティティとアクセスの管理](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccountPermissions](#)」の「」を参照してください。

list-assessment-run-agents

次のコード例は、list-assessment-run-agents を使用する方法を示しています。

AWS CLI

評価実行エージェントを一覧表示するには

次のlist-assessment-run-agentsコマンドは、指定された で実行された評価のエージェントを一覧表示しますARN。

```
aws inspector list-assessment-run-agents \
  --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE
```

出力:

```
{
  "assessmentRunAgents": [
    {
      "agentHealth": "HEALTHY",
      "agentHealthCode": "HEALTHY",
      "agentId": "i-49113b93",
      "assessmentRunArn": "arn:aws:inspector:us-
west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE",
      "telemetryMetadata": [
        {
          "count": 2,
          "dataSize": 345,
          "messageType": "InspectorDuplicateProcess"
        },
        {
          "count": 3,
          "dataSize": 255,
          "messageType": "InspectorTimeEventMsg"
        },
        {
          "count": 4,
          "dataSize": 1082,
          "messageType": "InspectorNetworkInterface"
        },
        {
          "count": 2,
```

```
        "dataSize": 349,  
        "messageType": "InspectorDnsEntry"  
    },  
    {  
        "count": 11,  
        "dataSize": 2514,  
        "messageType": "InspectorDirectoryInfoMsg"  
    },  
    {  
        "count": 1,  
        "dataSize": 179,  
        "messageType": "InspectorTcpV6ListeningPort"  
    },  
    {  
        "count": 101,  
        "dataSize": 10949,  
        "messageType": "InspectorTerminal"  
    },  
    {  
        "count": 26,  
        "dataSize": 5916,  
        "messageType": "InspectorUser"  
    },  
    {  
        "count": 282,  
        "dataSize": 32148,  
        "messageType": "InspectorDynamicallyLoadedCodeModule"  
    },  
    {  
        "count": 18,  
        "dataSize": 10172,  
        "messageType": "InspectorCreateProcess"  
    },  
    {  
        "count": 3,  
        "dataSize": 8001,  
        "messageType": "InspectorProcessPerformance"  
    },  
    {  
        "count": 1,  
        "dataSize": 360,  
        "messageType": "InspectorOperatingSystem"  
    },  
    {
```

```
        "count": 6,  
        "dataSize": 546,  
        "messageType": "InspectorStopProcess"  
    },  
    {  
        "count": 1,  
        "dataSize": 1553,  
        "messageType": "InspectorInstanceMetaData"  
    },  
    {  
        "count": 2,  
        "dataSize": 434,  
        "messageType": "InspectorTcpV4Connection"  
    },  
    {  
        "count": 474,  
        "dataSize": 2960322,  
        "messageType": "InspectorPackageInfo"  
    },  
    {  
        "count": 3,  
        "dataSize": 2235,  
        "messageType": "InspectorSystemPerformance"  
    },  
    {  
        "count": 105,  
        "dataSize": 46048,  
        "messageType": "InspectorCodeModule"  
    },  
    {  
        "count": 1,  
        "dataSize": 182,  
        "messageType": "InspectorUdpV6ListeningPort"  
    },  
    {  
        "count": 2,  
        "dataSize": 371,  
        "messageType": "InspectorUdpV4ListeningPort"  
    },  
    {  
        "count": 18,  
        "dataSize": 8362,  
        "messageType": "InspectorKernelModule"  
    },  
    },
```

```
{
  "count": 29,
  "dataSize": 48788,
  "messageType": "InspectorConfigurationInfo"
},
{
  "count": 1,
  "dataSize": 79,
  "messageType": "InspectorMonitoringStart"
},
{
  "count": 5,
  "dataSize": 0,
  "messageType": "InspectorSplitMsgBegin"
},
{
  "count": 51,
  "dataSize": 4593,
  "messageType": "InspectorGroup"
},
{
  "count": 1,
  "dataSize": 184,
  "messageType": "InspectorTcpV4ListeningPort"
},
{
  "count": 1159,
  "dataSize": 3146579,
  "messageType": "Total"
},
{
  "count": 5,
  "dataSize": 0,
  "messageType": "InspectorSplitMsgEnd"
},
{
  "count": 1,
  "dataSize": 612,
  "messageType": "InspectorLoadImageInProgress"
}
]
}
```

```
}
```

詳細については、[AWS 「Amazon Inspector ユーザーガイド」の「エージェント」](#)を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssessmentRunAgents](#)」の「」を参照してください。

list-assessment-runs

次の例は、list-assessment-runs を使用する方法を説明しています。

AWS CLI

評価実行を一覧表示するには

次のlist-assessment-runsコマンドは、既存の評価実行をすべて一覧表示します。

```
aws inspector list-assessment-runs
```

出力:

```
{
  "assessmentRunArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-MKkpXXPE",
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw/run/0-v5D6fI3v"
  ]
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector 評価テンプレートと評価実行](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssessmentRuns](#)」の「」を参照してください。

list-assessment-targets

次の例は、list-assessment-targets を使用する方法を説明しています。

AWS CLI

評価ターゲットを一覧表示するには

次の`list-assessment-targets`コマンドは、既存のすべての評価ターゲットを一覧表示します。

```
aws inspector list-assessment-targets
```

出力:

```
{
  "assessmentTargetArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq"
  ]
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価ターゲット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssessmentTargets](#)」の「」を参照してください。

`list-assessment-templates`

次の例は、`list-assessment-templates` を使用する方法を説明しています。

AWS CLI

評価テンプレートを一覧表示するには

次の`list-assessment-templates`コマンドは、既存のすべての評価テンプレートを一覧表示します。

```
aws inspector list-assessment-templates
```

出力:

```
{
  "assessmentTemplateArns": [
```

```
"arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
template/0-4r1V2mAw",
  "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-
Uza6ihLh"
]
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAssessmentTemplates](#)」の「」を参照してください。

list-coverage-statistics

次のコード例は、list-coverage-statistics を使用する方法を示しています。

AWS CLI

例 1: グループ別にカバレッジ統計を一覧表示するには

次のlist-coverage-statistics例では、AWS 環境のカバレッジ統計をグループ別に一覧表示します。

```
aws inspector2 list-coverage-statistics \
  --group-by RESOURCE_TYPE
```

出力:

```
{
  "countsByGroup": [
    {
      "count": 56,
      "groupKey": "AWS_LAMBDA_FUNCTION"
    },
    {
      "count": 27,
      "groupKey": "AWS_ECR_REPOSITORY"
    },
    {
      "count": 18,
      "groupKey": "AWS_EC2_INSTANCE"
    }
  ]
}
```



```
    },
    {
      "count": 3,
      "groupKey": "AWS_ECR_CONTAINER_IMAGE"
    },
    {
      "count": 1,
      "groupKey": "AWS_ACCOUNT"
    }
  ],
  "totalCounts": 105
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の AWS 「[環境の Amazon Inspector カバレッジの評価](#)」を参照してください。Amazon Inspector

例 2: リソースタイプ別にカバレッジ統計を一覧表示するには

次のlist-coverage-statistics例では、AWS 環境のカバレッジ統計をリソースタイプ別に一覧表示します。

```
aws inspector2 list-coverage-statistics
  --filter-criteria '{"resourceType":
  [{"comparison": "EQUALS", "value": "AWS_ECR_REPOSITORY"}]}'
  --group-by SCAN_STATUS_REASON
```

出力:

```
{
  "countsByGroup": [
    {
      "count": 27,
      "groupKey": "SUCCESSFUL"
    }
  ],
  "totalCounts": 27
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の AWS 「[環境の Amazon Inspector カバレッジの評価](#)」を参照してください。Amazon Inspector

例 3: ECRリポジトリ名でカバレッジ統計を一覧表示するには

次のlist-coverage-statistics例では、AWS 環境のカバレッジ統計をECRリポジトリ名別に一覧表示します。

```
aws inspector2 list-coverage-statistics
  --filter-criteria '{"ecrRepositoryName":
[{"comparison": "EQUALS", "value": "debian"}]}'
  --group-by SCAN_STATUS_REASON
```

出力:

```
{
  "countsByGroup": [
    {
      "count": 3,
      "groupKey": "SUCCESSFUL"
    }
  ],
  "totalCounts": 3
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)のAWS「[環境の Amazon Inspector カバレッジの評価](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI「[コマンドリファレンスListCoverageStatistics](#)」の「」を参照してください。

list-coverage

次のコード例は、list-coverage を使用する方法を示しています。

AWS CLI

例 1: 環境に関するカバレッジの詳細を一覧表示するには

次のlist-coverage例では、環境のカバレッジの詳細を一覧表示します。

```
aws inspector2 list-coverage
```

出力:

```
{
```

```

"coveredResources": [
  {
    "accountId": "123456789012",
    "lastScannedAt": "2024-05-20T16:23:20-07:00",
    "resourceId": "i-EXAMPLE555555555555",
    "resourceMetadata": {
      "ec2": {
        "amiId": "ami-EXAMPLE6666666666",
        "platform": "LINUX"
      }
    },
    "resourceType": "AWS_EC2_INSTANCE",
    "scanStatus": {
      "reason": "SUCCESSFUL",
      "statusCode": "ACTIVE"
    },
    "scanType": "PACKAGE"
  }
]
}

```

例 2: Lambda 関数リソースタイプに関するカバレッジの詳細を一覧表示するには
次のlist-coverage例では、Lambda 関数のリソースタイプの詳細を一覧表示します。

```

aws inspector2 list-coverage
  --filter-criteria '{"resourceType":
[{"comparison": "EQUALS", "value": "AWS_LAMBDA_FUNCTION"}]}'

```

出力:

```

{
  "coveredResources": [
    {
      "accountId": "123456789012",
      "resourceId": "arn:aws:lambda:us-west-2:123456789012:function:Eval-
container-scan-results:$LATEST",
      "resourceMetadata": {
        "lambdaFunction": {
          "functionName": "Eval-container-scan-results",
          "functionTags": {},
          "layers": [],
          "runtime": "PYTHON_3_7"
        }
      }
    }
  ]
}

```

```
    }
  },
  "resourceType": "AWS_LAMBDA_FUNCTION",
  "scanStatus": {
    "reason": "SUCCESSFUL",
    "statusCode": "ACTIVE"
  },
  "scanType": "CODE"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCoverage](#)」の「」を参照してください。

list-delegated-admin-accounts

次の例は、list-delegated-admin-accounts を使用する方法を説明しています。

AWS CLI

組織の委任管理者アカウントに関する情報を一覧表示するには

次のlist-delegated-admin-accounts例は、組織の委任管理者アカウントに関する情報を示しています。

```
aws inspector2 list-delegated-admin-accounts
```

出力:

```
{
  "delegatedAdminAccounts": [
    {
      "accountId": "123456789012",
      "status": "ENABLED"
    }
  ]
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector の委任管理者の指定](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDelegatedAdminAccounts](#)」の「」を参照してください。

list-event-subscriptions

次の例は、list-event-subscriptions を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを一覧表示するには

次のlist-event-subscriptionsコマンドは、の評価テンプレートのすべてのイベントサブスクリプションを ARN の で一覧表示しますarn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0。

```
aws inspector list-event-subscriptions --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0
```

出力:

```
{
  "subscriptions": [
    {
      "eventSubscriptions": [
        {
          "event": "ASSESSMENT_RUN_COMPLETED",
          "subscribedAt": 1459455440.867
        }
      ],
      "resourceArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0",
      "topicArn": "arn:aws:sns:us-west-2:123456789012:exampletopic"
    }
  ]
}
```

詳細については、「Amazon Inspector ガイド」のAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEventSubscriptions](#)」の「」を参照してください。

list-filters

次の例は、list-filters を使用する方法を説明しています。

AWS CLI

Amazon Inspector のアクティブ化に使用したアカウントに関連付けられたフィルターを一覧表示するには

次のlist-filters例では、Amazon Inspector のアクティブ化に使用したアカウントに関連付けられたフィルターを一覧表示します。

```
aws inspector2 list-filters
```

出力:

```
{
  "filters": [
    {
      "action": "SUPPRESS",
      "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE4444444444",
      "createdAt": "2024-05-15T21:11:08.602000+00:00",
      "criteria": {
        "resourceType": [
          {
            "comparison": "EQUALS",
            "value": "AWS_EC2_INSTANCE"
          }
        ]
      },
      "description": "This suppression rule omits EC2 instance type findings",
      "name": "ExampleSuppressionRuleEC2",
      "ownerId": "o-EXAMPLE222",
      "tags": {},
      "updatedAt": "2024-05-15T21:11:08.602000+00:00"
    },
    {
      "action": "SUPPRESS",
      "arn": "arn:aws:inspector2:us-east-1:813737243517:owner/o-EXAMPLE222/filter/EXAMPLE4444444444",
      "createdAt": "2024-05-15T21:28:27.054000+00:00",
      "criteria": {
```

```
        "resourceType": [
            {
                "comparison": "EQUALS",
                "value": "AWS_ECR_INSTANCE"
            }
        ],
        "description": "This suppression rule omits ECR instance type findings",
        "name": "ExampleSuppressionRuleECR",
        "ownerId": "o-EXAMPLE222",
        "tags": {},
        "updatedAt": "2024-05-15T21:28:27.054000+00:00"
    }
]
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector の検出結果のフィルタリング](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFilters](#)」の「」を参照してください。

list-findings

次のコード例は、list-findings を使用する方法を示しています。

AWS CLI

結果を一覧表示するには

次のlist-findingsコマンドは、生成されたすべての検出結果を一覧表示します。

```
aws inspector list-findings
```

出力:

```
{
    "findingArns": [
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
        template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4",
        "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/
        template/0-4r1V2mAw/run/0-v5D6fI3v/finding/0-tyvmqBly"
    ]
}
```

```
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector の検出結果」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFindings](#)」の「」を参照してください。

list-rules-packages

次の例は、list-rules-packages を使用する方法を説明しています。

AWS CLI

ルールパッケージを一覧表示するには

次のlist-rules-packagesコマンドは、使用可能なすべての Inspector ルールパッケージを一覧表示します。

```
aws inspector list-rules-packages
```

出力:

```
{
  "rulesPackageArns": [
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-H5hpSawc",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-JJ0tZiqQ",
    "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-vg5GGHSD"
  ]
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector ルールパッケージとルール」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRulesPackages](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resourceコマンドは、 の評価テンプレートに関連付けられたすべてのタグを ARNの で一覧表示しますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu。

```
aws inspector list-tags-for-resource --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu
```

出力:

```
{
  "tags": [
    {
      "key": "Name",
      "value": "Example"
    }
  ]
}
```

詳細については、「Amazon Inspector ガイド」のAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-usage-totals

次のコード例は、list-usage-totals を使用する方法を示しています。

AWS CLI

過去 30 日間の使用状況の合計を一覧表示するには

次のlist-usage-totals例では、過去 30 日間の使用状況の合計を一覧表示します。

```
aws inspector2 list-usage-totals
```

出力:

```
{
  "totals": [
    {
      "accountId": "123456789012",
      "usage": [
        {
          "currency": "USD",
          "estimatedMonthlyCost": 4.6022044647,
          "total": 1893.4784083333334,
          "type": "EC2_AGENTLESS_INSTANCE_HOURS"
        },
        {
          "currency": "USD",
          "estimatedMonthlyCost": 18.892449279,
          "total": 10882.050784722222,
          "type": "EC2_INSTANCE_HOURS"
        },
        {
          "currency": "USD",
          "estimatedMonthlyCost": 5.4525363736,
          "total": 6543.043648333333,
          "type": "LAMBDA_FUNCTION_CODE_HOURS"
        },
        {
          "currency": "USD",
          "estimatedMonthlyCost": 3.9064080309,
          "total": 9375.379274166668,
          "type": "LAMBDA_FUNCTION_HOURS"
        },
        {
          "currency": "USD",
          "estimatedMonthlyCost": 0.06,
          "total": 6.0,
          "type": "ECR_RESCAN"
        },
        {
          "currency": "USD",
          "estimatedMonthlyCost": 0.09,
          "total": 1.0,
          "type": "ECR_INITIAL_SCAN"
        }
      ]
    }
  ]
}
```

```
]
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector での使用状況とコストのモニタリング](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「[コマンドリファレンスListUsageTotals](#)」の「」を参照してください。

preview-agents

次の例は、preview-agents を使用する方法を説明しています。

AWS CLI

エージェントをプレビューするには

次のpreview-agentsコマンドは、 の評価ターゲットの一部であるEC2インスタンスにインストールされたエージェントを ARNの でプレビューしますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq。

```
aws inspector preview-agents --preview-agents-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

出力:

```
{
  "agentPreviews": [
    {
      "agentId": "i-49113b93"
    }
  ]
}
```

詳細については、[Amazon Inspector ガイド](#)の「[Amazon Inspector Assessment Targets](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスPreviewAgents](#)」の「」を参照してください。

register-cross-account-access-role

次のコード例は、register-cross-account-access-role を使用する方法を示しています。

AWS CLI

クロスアカウントアクセスロールを登録するには

次のregister-cross-account-access-roleコマンドは、preview-agents コマンドを呼び出すときに、評価実行の開始時にEC2インスタンスを一覧表示するために Amazon Inspector ARNarn:aws:iam::123456789012:role/inspectorが使用する IAMロールを登録します。

```
aws inspector register-cross-account-access-role --role-arn arn:aws:iam::123456789012:role/inspector
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector のセットアップ」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterCrossAccountAccessRole](#)」の「」を参照してください。

remove-attributes-from-findings

次のコード例は、remove-attributes-from-findings を使用する方法を示しています。

AWS CLI

検出結果から属性を削除するには

次のremove-attributes-from-findingコマンドは、のキーExampleと の値を持つ属性exampleを、ARNの を持つ結果から削除しますarn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU。

```
aws inspector remove-attributes-from-findings --finding-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attribute-keys key=Example,value=example
```

出力:

```
{
```

```
"failedItems": {}  
}
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector の検出結果」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveAttributesFromFindings](#)」の「」を参照してください。

set-tags-for-resource

次のコード例は、set-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを設定するには

次のset-tags-for-resourceコマンドは、のキーExampleと の値を持つ タグexampleを、ARNの を持つ評価テンプレートに設定しますarn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0。

```
aws inspector set-tags-for-resource --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0 --  
tags key=Example,value=example
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetTagsForResource](#)」の「」を参照してください。

start-assessment-run

次のコード例は、start-assessment-run を使用する方法を示しています。

AWS CLI

評価実行を開始するには

次のstart-assessment-runコマンドは、の評価テンプレートexamplerunを使用して、ARNという名前の評価実行を開始しますarn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T。

```
aws inspector start-assessment-run --assessment-run-name examplerun --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

出力:

```
{
  "assessmentRunArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY"
}
```

詳細については、「Amazon Inspector ガイド」のAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartAssessmentRun](#)」の「」を参照してください。

stop-assessment-run

次のコード例は、stop-assessment-run を使用する方法を示しています。

AWS CLI

評価実行を停止するには

次のstop-assessment-runコマンドは、ARNの を使用して評価実行を停止しますarn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY。

```
aws inspector stop-assessment-run --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY
```

詳細については、「Amazon Inspector ガイド」のAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopAssessmentRun](#)」の「」を参照してください。

subscribe-to-event

次の例は、subscribe-to-event を使用する方法を説明しています。

AWS CLI

イベントをサブスクライブするには

次の例では、を使用してASSESSMENT_RUN_COMPLETED、イベントに関する Amazon SNS通知をトピックに送信するプロセスを有効にしますARN。 `arn:aws:sns:us-west-2:123456789012:exampletopic`

```
aws inspector subscribe-to-event \  
  --event ASSESSMENT_RUN_COMPLETED \  
  --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/  
template/0-7sbz2Kz0 \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:exampletopic
```

このコマンドでは何も出力されません。

詳細については、[Amazon Inspector ガイドの「Amazon Inspector 評価テンプレートと評価実行」](#)を参照してください。 Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[SubscribeToEvent](#)」の「」を参照してください。

unsubscribe-from-event

次の例は、unsubscribe-from-event を使用する方法を説明しています。

AWS CLI

イベントからサブスクライブを解除するには

次のunsubscribe-from-eventコマンドは、ARNのを使用してASSESSMENT_RUN_COMPLETED、イベントに関する Amazon SNS通知をトピックに送信するプロセスを無効にしますarn:aws:sns:us-west-2:123456789012:exampletopic。

```
aws inspector unsubscribe-from-event --event ASSESSMENT_RUN_COMPLETED --resource-  
arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0  
  --topic arn:aws:sns:us-west-2:123456789012:exampletopic
```

詳細については、Amazon Inspector ガイドのAmazon Inspector 評価テンプレートと評価実行」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UnsubscribeFromEvent](#)」の「」を参照してください。

update-assessment-target

次のコード例は、update-assessment-target を使用する方法を示しています。

AWS CLI

評価ターゲットを更新するには

次のupdate-assessment-targetコマンドは、評価ターゲットを ARN で更新 `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX` し、 の名前を `Example` に、リソースグループを ARN で更新します `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt`。

```
aws inspector update-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX --assessment-target-name Example --resource-group-arn arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt
```

詳細については、「Amazon Inspector ガイド」の「Amazon Inspector 評価ターゲット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAssessmentTarget](#)」の「」を参照してください。

update-filter

次の例は、update-filter を使用する方法を説明しています。

AWS CLI

フィルターを更新するには

次のupdate-filter例では、ECRインスタンスの検出結果の代わりに Lambda の検出結果を省略するようにフィルターを更新します。

```
aws inspector2 update-filter \  
  --filter-arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE444444444" \  
  --name "ExampleSuppressionRuleLambda" \  
  --
```



```
--description "This suppression rule omits Lambda instance findings" \  
--reason "Updating filter to omit Lambda instance findings instead of ECR  
instance findings"
```

出力:

```
{  
  "filters": [  
    {  
      "action": "SUPPRESS",  
      "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/  
filter/EXAMPLE4444444444",  
      "createdAt": "2024-05-15T21:28:27.054000+00:00",  
      "criteria": {  
        "resourceType": [  
          {  
            "comparison": "EQUALS",  
            "value": "AWS_ECR_INSTANCE"  
          }  
        ]  
      },  
      "description": "This suppression rule omits Lambda instance findings",  
      "name": "ExampleSuppressionRuleLambda",  
      "ownerId": "o-EXAMPLE222",  
      "reason": "Updating filter to omit Lambda instance findings instead of  
ECR instance findings",  
      "tags": {},  
      "updatedAt": "2024-05-15T22:23:13.665000+00:00"  
    }  
  ]  
}
```

詳細については、[Amazon Inspector ユーザーガイド](#)の「[Amazon Inspector での検出結果の管理](#)」を参照してください。Amazon Inspector

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFilter](#)」の「」を参照してください。

AWS IoT を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-certificate-transfer

次の例は、`accept-certificate-transfer` を使用する方法を説明しています。

AWS CLI

別の AWS アカウントから転送されたデバイス証明書を受け入れるには

次の `accept-certificate-transfer` 例では、別の AWS アカウントから転送されたデバイス証明書を受け入れます。証明書は ID で識別されます。

```
aws iot accept-certificate-transfer \  
  --certificate-  
  id 488b6a7f2acdeb00a77384e63c4e40b18bEXAMPLEe57b7272ba44c45e3448142
```

このコマンドでは、出力が生成されません。

詳細については、「IoT Core デベロッパーガイド」の「[証明書を別のアカウントに転送する](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [AcceptCertificateTransfer](#)」の「」を参照してください。

add-thing-to-billing-group

次の例は、`add-thing-to-billing-group` を使用する方法を説明しています。

AWS CLI

例 1: 請求グループにモノを名前追加するには

次のadd-thing-to-billing-group例では、 という名前のモノMyLightBulbを という名前の請求グループに追加しますGroupOne。

```
aws iot add-thing-to-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyLightBulb
```

このコマンドでは何も出力されません。

例 2: ARN請求グループにモノを追加するには

次のadd-thing-to-billing-group例では、指定された を持つモノARNを、指定された を持つ請求グループに追加しますARN。の指定ARNは、複数の AWS リージョンまたはアカウントを使用する場合に役立ちます。これにより、 を適切なリージョンとアカウントに追加できます。

```
aws iot add-thing-to-thing-group \  
  --billing-group-arn "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne" \  
  --thing-arn "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb"
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの「[請求グループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[AddThingToBillingGroup](#)」の「」を参照してください。

add-thing-to-thing-group

次のコード例は、add-thing-to-thing-group を使用する方法を示しています。

AWS CLI

モノをグループに追加するには

次のadd-thing-to-thing-group例では、指定されたモノを指定されたモノグループに追加します。

```
aws iot add-thing-to-thing-group \  
  --thing-name MyLightBulb \  
  --thing-group-name LightBulbs
```

このコマンドでは何も出力されません。

詳細については、[「IoT デベロッパーガイド」の「モノのグループAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddThingToThingGroup](#)」の「」を参照してください。

associate-targets-with-job

次のコード例は、associate-targets-with-job を使用する方法を示しています。

AWS CLI

モノのグループを連続ジョブに関連付けるには

次のassociate-targets-with-job例では、指定されたモノのグループを指定された連続ジョブに関連付けます。

```
aws iot associate-targets-with-job \  
  --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --job-id "example-job-04"
```

出力:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",  
  "jobId": "example-job-04",  
  "description": "example continuous job"  
}
```

詳細については、AWS IoT デベロッパーガイドの[「ジョブの作成と管理 \(CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateTargetsWithJob](#)」の「」を参照してください。

attach-policy

次の例は、attach-policy を使用する方法を説明しています。

AWS CLI

例 1: モノのグループにポリシーをアタッチするには

次のattach-policy例では、指定されたポリシーを によって識別されるモノのグループにアタッチしますARN。

```
aws iot attach-policy \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --policy-name "UpdateDeviceCertPolicy"
```

このコマンドでは、出力が生成されません。

詳細については、[「IoT デベロッパーガイド」の「モノのグループAWS IoT」](#)を参照してください。

例 2: ポリシーを証明書にアタッチするには

次のattach-policy例では、証明書で指定されたプリンシパルUpdateDeviceCertPolicyにポリシーをアタッチします。

```
aws iot attach-policy \  
  --policy-name UpdateDeviceCertPolicy \  
  --target "arn:aws:iot:us-west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

このコマンドでは、出力が生成されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「[デバイス証明書に IoT ポリシーをアタッチする IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスAttachPolicy](#)」の「」を参照してください。

attach-security-profile

次のコード例は、attach-security-profile を使用する方法を示しています。

AWS CLI

セキュリティプロファイルを登録されていないすべてのデバイスに関連付けるには

次のattach-security-profile例では、 という名前の AWS IoT Device Defender セキュリティプロファイルを、この AWS アカウントの us-west-2リージョン内のすべての未登録デバイスTestprofileと関連付けます。

```
aws iot attach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/  
unregistered-things"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachSecurityProfile](#)」の「」を参照してください。

attach-thing-principal

次の例は、attach-thing-principal を使用する方法を説明しています。

AWS CLI

モノに証明書をアタッチするには

次のattach-thing-principal例では、MyTemperatureSensor モノに証明書をアタッチします。証明書は によって識別されますARN。AWS IoT コンソールで証明書ARNの を見つけることができます。

```
aws iot attach-thing-principal \  
  --thing-name MyTemperatureSensor \  
  --principal arn:aws:iot:us-  
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

このコマンドでは何も出力されません。

詳細については、「AWS IoT デベロッパーガイド」の「[レジストリによるモノの管理方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachThingPrincipal](#)」の「」を参照してください。

cancel-audit-mitigation-actions-task

次のコード例は、cancel-audit-mitigation-actions-task を使用する方法を示しています。

AWS CLI

監査緩和アクションタスクをキャンセルするには

次のcancel-audit-mitigations-action-task例では、指定されたタスクの緩和アクションの適用をキャンセルします。既に完了したタスクはキャンセルできません。

```
aws iot cancel-audit-mitigation-actions-task
  --task-id "myActionsTaskId"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの[CancelAuditMitigationActionsTask](#)「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CancelAuditMitigationActionsTask](#)」の「」を参照してください。

cancel-audit-task

次のコード例は、cancel-audit-task を使用する方法を示しています。

AWS CLI

監査タスクをキャンセルするには

次のcancel-audit-task例では、指定されたタスク ID を持つ監査タスクをキャンセルします。完了したタスクをキャンセルすることはできません。

```
aws iot cancel-audit-task \
  --task-id a3aea009955e501a31b764abe1bebd3d
```

このコマンドでは何も出力されません。

詳細については、AWS「IoT デベロッパーガイド」の[「監査コマンド」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CancelAuditTask](#)」の「」を参照してください。

cancel-certificate-transfer

次の例は、cancel-certificate-transfer を使用する方法を説明しています。

AWS CLI

別の AWS アカウントへの証明書の移管をキャンセルするには

次のcancel-certificate-transfer例では、指定された証明書転送の転送をキャンセルします。証明書は証明書 ID で識別されます。AWS IoT コンソールで証明書の ID を確認できます。

```
aws iot cancel-certificate-transfer \  
  --certificate-  
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78add5e605d630e05c7fc8
```

このコマンドでは何も出力されません。

詳細については、[「IoT Core デベロッパーガイド」の「証明書を別のアカウントに転送するAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelCertificateTransfer](#)」の「」を参照してください。

cancel-job-execution

次のコード例は、cancel-job-execution を使用する方法を示しています。

AWS CLI

デバイスでジョブ実行をキャンセルするには

次のcancel-job-execution例では、デバイスで指定されたジョブの実行をキャンセルします。ジョブが QUEUED状態にない場合は、--forceパラメータを追加する必要があります。

```
aws iot cancel-job-execution \  
  --job-id "example-job-03" \  
  --thing-name "MyRPi"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの[「ジョブの作成と管理 \(CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelJobExecution](#)」の「」を参照してください。

cancel-job

次のコード例は、cancel-job を使用する方法を示しています。

AWS CLI

ジョブをキャンセルするには

次のcancel-job例では、指定されたジョブをキャンセルします。

```
aws iot cancel-job \  
  --job-id "example-job-03"
```

出力:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-03",  
  "jobId": "example-job-03",  
  "description": "example job test"  
}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelJob](#)」の「」を参照してください。

clear-default-authorizer

次のコード例は、clear-default-authorizer を使用する方法を示しています。

AWS CLI

デフォルトのオーソライザーをクリアするには

次のclear-default-authorizer例では、現在設定されているデフォルトのカスタムオーソライザーをクリアします。このコマンドを実行すると、デフォルトのオーソライザーはありません。カスタムオーソライザーを使用する場合は、HTTPリクエストヘッダーに名前を指定する必要があります。

```
aws iot clear-default-authorizer
```

このコマンドでは何も出力されません。

詳細については、AWS IoT APIリファレンスの[ClearDefaultAuthorizer](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ClearDefaultAuthorizer](#)」の「」を参照してください。

confirm-topic-rule-destination

次の例は、confirm-topic-rule-destination を使用する方法を説明しています。

AWS CLI

トピックルールの送信先を確認するには

次のconfirm-topic-rule-destination例では、HTTPエンドポイントで受信した確認トークンを使用してトピックルールの送信先を確認します。

```
aws iot confirm-topic-rule-destination \  
  --confirmation-token "AYADeIcmtq-  
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmXPYy1rZXkAREFXY1E0UmLGeDg0V21BZWZ1VjZtZWFRVUJJUkt  
aywpPqg8YEsa1LD4B40aJ2s1wEHKMybiF1Ro0ZzYisI0IvsLzQY5UmCkq3tV-3f7-  
nKfosgIAAAAADAAAEEAAAAAAAAAAAAAAAAAAAAAaI9RMgy-  
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAAAAAEAAAAB1hw4SokgUcxiJ3gT06n50NLJVpzyQR1UmPIj5sShqXEQGcC  
iufgrzTePl8RZY0Wr006Aj9DiVzJZx-1iD6Pu-  
G6PUw1ka07Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRmhKqntd_c6Kgrow3bMUDbVnQo2qZr8Z8Jm2rzgseR0LAn  
PIetJ803Z4IILIF8xXlCdPGP-PV1d0XFemyL8g"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[トピックルールの送信先の確認](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ConfirmTopicRuleDestination](#)」の「」を参照してください。

create-audit-suppression

次の例は、create-audit-suppression を使用する方法を説明しています。

AWS CLI

監査結果の抑制を作成するには

次のcreate-audit-suppression例では、virtualMachinePolicy「」という名前のポリシーに対して、過度に許容されているというフラグが付けられた監査結果の抑制を作成します。

```
aws iot create-audit-suppression \
  --check-name IOT_POLICY_OVERLY_PERMISSIVE_CHECK \
  --resource-identifier
  policyVersionIdentifier={"policyName"="virtualMachinePolicy","policyVersionId"="1"}
  \
  --no-suppress-indefinitely \
  --expiration-date 2020-10-20
```

このコマンドでは何も出力されません。

詳細については、[「IoT デベロッパーガイド」の「監査結果の抑制AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateAuditSuppression](#)」の「」を参照してください。

create-authorizer

次のコード例は、create-authorizer を使用する方法を示しています。

AWS CLI

カスタムオーソライザーを作成するには

次のcreate-authorizer例では、カスタム認証サービスの一部として指定された Lambda 関数を使用するカスタムオーソライザーを作成します。

```
aws iot create-authorizer \
  --authorizer-name "CustomAuthorizer" \
  --authorizer-function-arn "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction" \
  --token-key-name "MyAuthToken" \
  --status ACTIVE \
  --token-signing-public-keys FIRST_KEY="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEA1uJ0B41QPgG/1M6ZfIwo
Z+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb503kRSmyIwTzwzm/f4Gf0Y
ZUloJ+t3PUUwHrmbYTAgrCUgRFyggjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
zw0BKPeic0asNJpqT8PkBbRaKylJh5oo81NDHmVtbBm5A5YiJjqYXLaVAowKzZ
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfP0HDJQgID0XnZwAlNnZc0hCwIx
50g2LW20y9R/dmqtDmJiVP97Z4GykxPvwLYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1
```

```
1QIDAQAB
-----END PUBLIC KEY-----"
```

出力:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

詳細については、AWS IoT APIリファレンスの[CreateAuthorizer](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateAuthorizer](#)」の「」を参照してください。

create-billing-group

次の例は、create-billing-group を使用する方法を説明しています。

AWS CLI

請求グループを作成するには

次のcreate-billing-group例では、`GroupOne`という名前の単純な請求グループを作成します。

```
aws iot create-billing-group \
  --billing-group-name GroupOne
```

出力:

```
{
  "billingGroupName": "GroupOne",
  "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
  "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562"
}
```

詳細については、IoT デベロッパーガイドの「[請求グループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[CreateBillingGroup](#)」の「」を参照してください。

create-certificate-from-csr

次のコード例は、create-certificate-from-csr を使用する方法を示しています。

AWS CLI

証明書署名リクエストからデバイス証明書を作成するには (CSR)

次のcreate-certificate-from-csr例では、 からデバイス証明書を作成します CSR。 openssl コマンドを使用して を作成できます CSR。

```
aws iot create-certificate-from-csr \  
  --certificate-signing-request=file://certificate.csr
```

出力:

```
{  
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/  
c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificateId":  
  "c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificatePem": "<certificate-text>"  
}
```

詳細については、AWS IoT APIリファレンスの[CreateCertificateFromCSR](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCertificateFromCsr](#)」の「」を参照してください。

create-custom-metric

次の例は、create-custom-metric を使用する方法を説明しています。

AWS CLI

デバイスが Device Defender に発行するカスタムメトリクスを作成するには

次のcreate-custom-metric例では、バッテリーの割合を測定するカスタムメトリクスを作成します。

```
aws iot create-custom-metric \  

```

```
--metric-name "batteryPercentage" \  
--metric-type "number" \  
--display-name "Remaining battery percentage." \  
--region us-east-1 \  
--client-request-token "02ccb92b-33e8-4dfa-a0c1-35b181ed26b0"
```

出力:

```
{  
  "metricName": "batteryPercentage",  
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/  
batteryPercentage"  
}
```

詳細については、「IoT Core デベロッパーガイド」の[「カスタムメトリクス」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomMetric](#)」の「」を参照してください。

create-dimension

次のコード例は、create-dimension を使用する方法を示しています。

AWS CLI

ディメンションを作成するには

以下では、という 1 つのトピックフィルターでディメンション create-dimension を作成します TopicFilterForAuthMessages。

```
aws iot create-dimension \  
  --name TopicFilterForAuthMessages \  
  --type TOPIC_FILTER \  
  --string-values device/+/auth
```

出力:

```
{  
  "name": "TopicFilterForAuthMessages",
```

```
"arn": "arn:aws:iot:eu-west-2:123456789012:dimension/TopicFilterForAuthMessages"
}
```

詳細については、AWS IoT デベロッパーガイドの「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDimension](#)」の「」を参照してください。

create-domain-configuration

次のコード例は、create-domain-configuration を使用する方法を示しています。

AWS CLI

ドメイン設定を作成するには

次のcreate-domain-configuration例では、サービスタイプが の AWS マネージドドメイン設定を作成します DATA。

```
aws iot create-domain-configuration \
  --domain-configuration-name "additionalDataDomain" \
  --service-type "DATA"
```

出力:

```
{
  "domainConfigurationName": "additionalDataDomain",
  "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

詳細については、AWS IoT デベロッパーガイドの「[設定可能なエンドポイント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDomainConfiguration](#)」の「」を参照してください。

create-dynamic-thing-group

次の例は、create-dynamic-thing-group を使用する方法を説明しています。

AWS CLI

動的モノのグループを作成するには

次のcreate-dynamic-thing-group例では、60 度を超える温度属性を持つモノを含む動的なモノグループを作成します。動的モノのグループを使用する前に、AWS IoT フリートのインデックス作成を有効にする必要があります。

```
aws iot create-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm" \  
  --query-string "attributes.temperature>60"
```

出力:

```
{  
  "thingGroupName": "RoomTooWarm",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",  
  "thingGroupId": "9d52492a-fc87-43f4-b6e2-e571d2ffcad1",  
  "indexName": "AWS_Things",  
  "queryString": "attributes.temperature>60",  
  "queryVersion": "2017-09-30"  
}
```

詳細については、IoT デベロッパーガイドの「[Dynamic Thing Groups](#)」を参照してください。

AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDynamicThingGroup](#)」の「」を参照してください。

create-job

次のコード例は、create-job を使用する方法を示しています。

AWS CLI

例 1: ジョブを作成するには

次のcreate-job例では、MyRaspberryPiデバイスにJSONドキュメントを送信するシンプルなAWS IoT ジョブを作成します。

```
aws iot create-job \  

```



```
--job-id "example-job-01" \  
--targets "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi" \  
--document file://example-job.json \  
--description "example job test" \  
--target-selection SNAPSHOT
```

出力:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",  
  "jobId": "example-job-01",  
  "description": "example job test"  
}
```

例 2: 連続ジョブを作成するには

次のcreate-job例では、ターゲットとして指定されたモノがジョブを完了した後も引き続き実行されるジョブを作成します。この例では、ターゲットはモノのグループであるため、新しいデバイスがグループに追加されると、それらの新しいモノに対して連続ジョブが実行されます。

```
aws iot create-job --job-id 「example-job-04」 --targets 「arn:aws:iot:us-west-2:123456789012:thinggroup/DeadBulbs」 --document file://example-job.json --description 「連続ジョブの例」 --target-selection CONTINUOUS
```

出力:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",  
  "jobId": "example-job-04",  
  "description": "example continuous job"  
}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateJob](#)」の「」を参照してください。

create-keys-and-certificate

次のコード例は、create-keys-and-certificate を使用する方法を示しています。

AWS CLI

RSAキーペアを作成して X.509 証明書を発行するには

以下は、2048 ビットのRSAキーペア `create-keys-and-certificate` を作成し、発行されたパブリックキーを使用して X.509 証明書を発行します。IoT がこの証明書のプライベートキー AWS IoT を提供するのはいずれのみですので、必ず安全な場所に保管してください。

```
aws iot create-keys-and-certificate \
  --certificate-pem-outfile "myTest.cert.pem" \
  --public-key-outfile "myTest.public.key" \
  --private-key-outfile "myTest.private.key"
```

出力:

```
{
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificateId":
    "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMC
VVMxCzAJBgNVBAGYEXAMPLEAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDAASBgNVBAwTC01BTSEXAMPLE2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMaKGA1UEBhMCEXAMPLEJBgNVBAgTA1dBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEXAMPLE251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvYsWtC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZEXAMPLEELG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qEXAMPLEyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEEXAMPLEBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQUFAAQ8AMIIBCGKCAQEAEXAMPLE1nnyJwKSMHw4h\nMMEXAMPLEEuuN/
dMAS3fyce8DW/4+EXAMPLEYjmoF/YVF/gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqX0gHh/
xJTtwo
+sGpWEXAMPLEDz18x0d2ka4tCzuWEXAMPLEEahJbYkCPUBSU8opVkr7qkEXAMPLE1DR6sx2Hoc1i00Ltu6Fkw91swQWEX"
```

```

\GB3ZPrNh0PzQYvjUStZeccyNCx2EXAMPLEVp9mQ0UXP6p1fgxwKRX2fEXAMPLEDa
\nhJLXkX3rHU2xbxJSq7D+XEXAMPLECw+LyFhI5mgFR188eGdsAEXAMPLElnI9EesG\nFQIDAQAB\n-----
END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omitted for security
reasons\n-----END RSA PRIVATE KEY-----\n"
  }
}

```

詳細については、[AWS「IoT デベロッパーガイド」の「IoT デバイス証明書の作成と登録」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[CreateKeysAndCertificate](#)」の「」を参照してください。

create-mitigation-action

次のコード例は、create-mitigation-action を使用する方法を示しています。

AWS CLI

緩和アクションを作成するには

次のcreate-mitigation-action例では、という名前の緩和アクションを定義します。この緩和アクションを適用するAddThingsToQuarantineGroup1Actionと、モノはという名前のモノのグループに移動しますQuarantineGroup1。このアクションは、動的なモノのグループを上書きします。

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

params.json の内容:

```

{
  "actionName": "AddThingsToQuarantineGroup1Action",
  "actionParams": {
    "addThingsToThingGroupParams": {
      "thingGroupNames": [
        "QuarantineGroup1"
      ],
      "overrideDynamicGroups": true
    }
  },
}

```

```
"roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole"
}
```

出力:

```
{
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroup1Action",
  "actionId": "992e9a63-a899-439a-aa50-4e20c52367e1"
}
```

詳細については、AWS IoT デベロッパーガイドの[CreateMitigationAction](#)「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateMitigationAction](#)」の「」を参照してください。

create-ota-update

次のコード例は、create-ota-update を使用する方法を示しています。

AWS CLI

Amazon Free で使用するOTA更新を作成するにはRTOS

次のcreate-ota-update例では、モノまたはグループのOTAUpdateターゲットグループにAWS IoTを作成します。これは Amazon FreeRTOS over-the-air 更新の一部であり、単一のデバイスまたはデバイスのグループに新しいファームウェアイメージをデプロイできます。

```
aws iot create-ota-update \
  --cli-input-json file://create-ota-update.json
```

create-ota-update.json の内容:

```
{
  "otaUpdateId": "ota12345",
  "description": "A critical update needed right away.",
  "targets": [
    "device1",
```

```

        "device2",
        "device3",
        "device4"
    ],
    "targetSelection": "SNAPSHOT",
    "awsJobExecutionsRolloutConfig": {
        "maximumPerMinute": 10
    },
    "files": [
        {
            "fileName": "firmware.bin",
            "fileLocation": {
                "stream": {
                    "streamId": "004",
                    "fileId": 123
                }
            },
            "codeSigning": {
                "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
            }
        }
    ]
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
}

```

出力:

```

{
    "otaUpdateId": "ota12345",
    "awsIotJobId": "job54321",
    "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",
    "awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/itsajob",
    "otaUpdateStatus": "CREATE_IN_PROGRESS"
}

```

詳細については、IoT リファレンスの「[CreateOTAUpdate](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateOtaUpdate](#)」の「」を参照してください。

create-policy-version

次のコード例は、create-policy-version を使用する方法を示しています。

AWS CLI

ポリシーを新しいバージョンで更新するには

次のcreate-policy-version例では、ポリシー定義を更新し、新しいポリシーバージョンを作成します。この例では、新しいバージョンもデフォルトになります。

```
aws iot create-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-document file://policy.json \  
  --set-as-default
```

policy.json の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iot:UpdateCertificate",  
      "Resource": "*"  
    }  
  ]  
}
```

出力:

```
{  
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",  
  "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
  \"Allow\", \"Action\": \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",  
  "policyVersionId": "2",  
  "isDefaultVersion": true  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT ポリシー-AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePolicyVersion](#)」の「」を参照してください。

create-policy

次の例は、create-policy を使用する方法を説明しています。

AWS CLI

AWS IoT ポリシーを作成するには

次のcreate-policy例では、`TemperatureSensorPolicy` という名前の AWS IoT ポリシーを作成します。policy.json ファイルには、AWS IoT ポリシーアクションを許可するステートメントが含まれています。

```
aws iot create-policy \  
  --policy-name TemperatureSensorPolicy \  
  --policy-document file://policy.json
```

policy.json の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Publish",  
        "iot:Receive"  
      ],  
      "Resource": [  
        "arn:aws:iot:us-west-2:123456789012:topic/topic_1",  
        "arn:aws:iot:us-west-2:123456789012:topic/topic_2"  
      ]  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Subscribe"  
      ],  
      "Resource": [  
        "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1",  
        "arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2"  
      ]  
    }  
  ],  
  {  
    {
```

```

    "Effect": "Allow",
    "Action": [
        "iot:Connect"
    ],
    "Resource": [
        "arn:aws:iot:us-west-2:123456789012:client/basicPubSub"
    ]
  }
]
}

```

出力:

```

{
  "policyName": "TemperatureSensorPolicy",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
  "policyDocument": "{
    \"Version\": \"2012-10-17\",
    \"Statement\": [
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Publish\",
          \"iot:Receive\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
        ]
      },
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Subscribe\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1\",
          \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2\"
        ]
      },
      {
        \"Effect\": \"Allow\",

```



```

        \ "Action\": [
            \ "iot:Connect\"
        ],
        \ "Resource\": [
            \ "arn:aws:iot:us-west-2:123456789012:client/basicPubSub\"
        ]
    }
]
}],
"policyVersionId": "1"
}

```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT ポリシーAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePolicy](#)」の「」を参照してください。

create-provisioning-claim

次のコード例は、create-provisioning-claim を使用する方法を示しています。

AWS CLI

プロビジョニングクレームを作成するには

次のcreate-provisioning-claim例では、プロビジョニングテンプレートからプロビジョニングクレームを作成します。

```

aws iot create-provisioning-claim \
  --template-name MyTestProvisioningTemplate

```

出力:

```

{
  "certificateId":
    "78de02184b2ce80cf8fb709bda59e62b19fb83513590483eb0434589476ab09f",
  "certificatePem": "-----BEGIN CERTIFICATE-----\nMIIDdzCCA1
+gAwIBAgIUXSZheBLztMLZ2fHG
14gV0NymYY0wDQYJKoZIhvcNAQEL
\nBQAwfjELMAkGA1UEBhMCVVMxEzARBgNVBAgMC1dhc2hpbmd0b24xEDA0Bg
VBAcM\nB1NlYXR0bGUxGDAWBgNVBAoMD0FtYXpvcvi5jb20gSW5jLjEgMB4GA1UECwwXQW1h
\nem9uIElvcCBQcm9

```

```

2aXNpb25pbmcxDDAKBgNVBAUTAzEuMDAeFw0yMDA3MjgxnjQ0\nMDZaFw0yMDA3MjgxnjUxMDZaMEsxsTBHBgNVB
AMMQDFhNDEyM2VkNmIxYjU3MzE3\nZTgzMTJmY2MzN2FiNTdhY2MzYTZkZGVjOGQ5OGY3NzUwMWRlMjc0YjhmYTQ
xN2Iw\nggEiMA0GCSqGSIb3EXAMPLEAA4IBDwAwggEKAoIBAQBhKI94ktKLqTwnj+ay0q1\nTAJt/
N6s6IJDZv1
rYjkC0E7wzaeY3TprWk03S29vUzVuE0XHXQXZbihgpg2m6fza\nkwm9/
wpjzE9ny5+xkPGVH4Wnwz7yK5m8S0agL
T96cRBSWnWmon0WdY0GKVzni0CA\n+iyGudgrFKm7Eae/
v18oXrf82Kt0AG04xG0KE2WKYHsT1fx3c9xZh1XP/eX
Lhv00\n+1Gp0WVw9PbhKfrxliKJ5q6sL5nVUaUHq6h1QPYwsATe0vAp3u0ak5zgTyL0fg7Y
\nPyKk6VYwLW62r+V
YBSForEM0Ahkq3LsP/rjxpEKmi2W41PVS6oFZRKcD+H1Kyil5\nAgMBAAGjIDAeMAwGA1UdEwEB/
wQCMAAwDgYDV
R0PAQH/BAQDAgeAMA0GCSqGSIb3\nDQEBcWUAA4IBAQAQgix2k6nVqbZFKq97/fZBzLGS0dyz5rT/
E41cDIRX+1j
EPW41\nnw0D+2sXheCZLZZnSkvIiP74IToNeXDrjdcaodeGFVHIElRjhMIq+4ZebPbRLtidF
\nRc2hfcTAlqq9Z6v
5Vk6BeM1tu0RqH1wPoVUccLPya8EjNCbnJZUmGd0frN/Y9pho\n5ikV+HPeZhG/k6dhE2GsQJyKfVHL/
uBgKSily
1bRyWU1r6qcpWBNBHjUoD7Hg0wD
\nnzMh4XRb2FQDsqFalkCSYmeL8IVC49sgPD90typ5uteGMTy62usAAUQdq/f
ZvrWg\n0kFpwMVnGKVKT7Kq0kK0LzKW0BB2Jm4/gmrJ\n-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCg
KCAQEAwYSiPeJLSi6k8J4/msjq
\nUwCbfzer0iCQ2b5a2I5AtB08M2nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYN
pun8\n2pFpvf8KY8xPZ8ufsZDx1R+Fp8M+8iuZvEtGoC0/enEQUl1pqJzlnWNBilc54tA
\nngPoshrnYKxSpuxGn
v79fKF63/NirTgBjuMRtChNlimEXAMPLE3PcWYZVz/3ly4b9\nNPPRqdFlcPT24Sn68ZYiieaurC
+Z1VG1B6uoZU
D2MLAE3jrwKd7tGp0c4E8i9H40\n2D8ip0lWMC1utq/
lWAUhaKxDDgIZKty7D/648aRCpotluJT1UuqBWUSnA/h9
Ssop\nEQIDAQAB\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAwYSiPeJLSi6k8J4/
msjqtUwCbfzer0iCQ2b5a2I5AtB08M2n
\nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYNpun82pFpvf8KY8xPZ8ufsZ
Dx1R+Fp8M+\n8iuZvEtGoC0/enEQUl1pqJzlnWNBilc54tAgPoshrnYKxSpuxGnv79fKF63/Nir
\nTgBjuMRtCh
NlimB7E9X8d3PcWYZVz/3ly4b9NPPRqdFlcPT24Sn68ZYiieaurC+Z
\n1VG1B6uoZUD2MLAE3jrwKd7tGp0c4E8i
9H402D8ip0lWMC1utq/lWAUhaKxDDgIZ\nKty7D/648aRCpotluJT1UuqBWUSnA/
h9SsopeQIDAQABaoIBAEaybN

```

```

QUtx9T2/nK\nztZ2pA4iugecxI4dz+DmT0XVXs5VJmrx/
nBSq6ejXExEpSIM04RY7LE3ZdJcnd56\nF7tQkkY7yR
VzfxHeXFU1kr0IPuxWebN0rRoPZr+1Rser+ww2aBC525+88pVuR6tM
\nm3pgkrR2ycCj9Fd0UoQxdjHBHaM5PDMj
9aSxCKdg3nReepeGwsR2TQA+m2vVxWk7\nou0+91eTOP+/QfP7P8Zj0Ik02Xiv1RcVDyN/
E4QXPkuIkM/8vS8VK+
E9pATQ0MtB\n2lw8R/YU5AJd6jlEXAMPLEGU2UzRzInNWiLtkPPPqgqXXhx0f+mxByjcMa1VJk0L
\nh0G2R0UCgY
EA+R0cHNHy/XbsP7Fih0hEh+6Q2QxQ2ncBUPYbBazrR8Hn+7SCICQK
\nVyYfd8Ajfq3e7RsKVL5S1MBp7S1idxak
bIn28fKfPn62DaemGCIOyDgLf+eUxBx
\nngzbCiBZga8brfurza43UZjKZLpg3hq721+FeAiXi1Nma4Yr9YWEHEN
8CgYEAxuwT\npzdWwmsiFzfsAw0sy9ySDA/xr5WRWzJyAqUsjsks6rxNzWebpufnYHcmtW7pLdqM
\nkboHwN2pXa
kmZvrk2nKkEMq5brBYGDxuxDe+V369Bianx8aZFYIsckA70wXW1w1h
\nngRC5rQ4X0gp3+Jmw7eA08LRYDjaN846+
Qbt02KcCgYAWS0UL51bijQR0ZwI0dz27\nnFQVuCAYsp748aurcRTACCj8jbnK/
QbqTNlxWsaH7ssBjZKo2D5sAqY
BRtASW0Dab\naHXsDhVm2Jye+ESLoHMaCLoyCkT3118yqXIcEDStM07f01Ryag164EiJvSIrMfny\nnNL/
fXVjCSH
/udCxdzPt+7QKBgQC+LAD7rxdr4J9538hTqpc4XK9vxRbrMXEH55XH
\nHbMa2x0NZXpmeTgEQBukyohCVceyRhK9
i0e6irZTjVXgh0eoTpC8VXkzcnzouTiQ
\nfeFQQSGfnp7Ioe6UIz23715pKdudzSNkMSKrG924ktv7CyDBF1gBQI5g
aDoHnddJBJ\nnPRtIZQKBgA8MASxtTxQntRwXXzR92U0vAighiuRkB/mx9jQpUcK1qiqHbkAMqgNF
\nPFCBYIUbFT
iYKKKeJNbyJQvjfsJcKAnaFJ+RnTxk0Q6Wjm20peJ/ii4QiDdnigoE\nnvd1c5cFQewWb4/
zqAtPdinkPLN94ileI
79XQdc7R1J0jpgTimL+V\n-----END RSA PRIVATE KEY-----\n"
    },
    "expiration": 1595955066.0
}

```

詳細については、[「IoT Core デベロッパーガイド」の「信頼できるユーザーによるプロビジョニングAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProvisioningClaim](#)」の「」を参照してください。

create-provisioning-template-version

次の例は、create-provisioning-template-version を使用する方法を説明しています。

AWS CLI

プロビジョニングテンプレートバージョンを作成するには

次の例では、指定されたプロビジョニングテンプレートのバージョンを作成します。新しいバージョンの本文は、ファイルで提供されます `template.json`。

```
aws iot create-provisioning-template-version \  
  --template-name widget-template \  
  --template-body file://template.json
```

`template.json` の内容:

```
{  
  "Parameters" : {  
    "DeviceLocation": {  
      "Type": "String"  
    }  
  },  
  "Mappings": {  
    "LocationTable": {  
      "Seattle": {  
        "LocationUrl": "https://example.aws"  
      }  
    }  
  },  
  "Resources" : {  
    "thing" : {  
      "Type" : "AWS::IoT::Thing",  
      "Properties" : {  
        "AttributePayload" : {  
          "version" : "v1",  
          "serialNumber" : "serialNumber"  
        },  
        "ThingName" : {"Fn::Join":["",["ThingPrefix_",  
{"Ref":"SerialNumber"}]]},  
        "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",  
{"Ref":"SerialNumber"}]]},  
        "ThingGroups" : ["widgets", "WA"],  
        "BillingGroup": "BillingGroup"  
      },  
      "OverrideSettings" : {  
        "AttributePayload" : "MERGE",
```

```

        "ThingTypeName" : "REPLACE",
        "ThingGroups" : "DO_NOTHING"
    }
},
"certificate" : {
    "Type" : "AWS::IoT::Certificate",
    "Properties" : {
        "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
        "Status" : "Active"
    }
},
"policy" : {
    "Type" : "AWS::IoT::Policy",
    "Properties" : {
        "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement": [{
                "Effect": "Allow",
                "Action":["iot:Publish"],
                "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/foo/
bar"]
            }]
        }
    }
},
"DeviceConfiguration": {
    "FallbackUrl": "https://www.example.com/test-site",
    "LocationUrl": {
        "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
"LocationUrl"]}
    }
}
}

```

出力:

```

{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "versionId": 2,
    "isDefaultVersion": false
}

```

```
}
```

詳細については、[AWS 「IoT Core デベロッパーガイド」の「IoT Secure Tunneling」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateProvisioningTemplateVersion](#)」の「」を参照してください。

create-provisioning-template

次の例は、create-provisioning-template を使用する方法を説明しています。

AWS CLI

プロビジョニングテンプレートを作成するには

次のcreate-provisioning-template例では、ファイルで定義されるプロビジョニングテンプレートを作成しますtemplate.json。

```
aws iot create-provisioning-template \  
  --template-name widget-template \  
  --description "A provisioning template for widgets" \  
  --provisioning-role-arn arn:aws:iam::123456789012:role/Provision_role \  
  --template-body file://template.json
```

template.json の内容:

```
{  
  "Parameters" : {  
    "DeviceLocation": {  
      "Type": "String"  
    }  
  },  
  "Mappings": {  
    "LocationTable": {  
      "Seattle": {  
        "LocationUrl": "https://example.aws"  
      }  
    }  
  },  
  "Resources" : {  
    "thing" : {  
      "Type" : "AWS::IoT::Thing",
```

```

    "Properties" : {
      "AttributePayload" : {
        "version" : "v1",
        "serialNumber" : "serialNumber"
      },
      "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
      "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
      "ThingGroups" : ["widgets", "WA"],
      "BillingGroup": "BillingGroup"
    },
    "OverrideSettings" : {
      "AttributePayload" : "MERGE",
      "ThingTypeName" : "REPLACE",
      "ThingGroups" : "DO_NOTHING"
    }
  },
  "certificate" : {
    "Type" : "AWS::IoT::Certificate",
    "Properties" : {
      "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
      "Status" : "Active"
    }
  },
  "policy" : {
    "Type" : "AWS::IoT::Policy",
    "Properties" : {
      "PolicyDocument" : {
        "Version": "2012-10-17",
        "Statement": [{
          "Effect": "Allow",
          "Action":["iot:Publish"],
          "Resource": ["arn:aws:iot:us-east-1:504350838278:topic/foo/
bar"]
        }]
      }
    }
  },
  "DeviceConfiguration": {
    "FallbackUrl": "https://www.example.com/test-site",
    "LocationUrl": {

```

```
        "Fn::FindInMap": ["LocationTable", {"Ref": "DeviceLocation"},
        "LocationUrl"]}
    }
}
```

出力:

```
{
  "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
  template",
  "templateName": "widget-template",
  "defaultVersionId": 1
}
```

詳細については、[AWS 「IoT Core デベロッパーガイド」の「IoT Secure Tunneling」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProvisioningTemplate](#)」の「」を参照してください。

create-role-alias

次のコード例は、`create-role-alias` を使用する方法を示しています。

AWS CLI

ロールエイリアスを作成するには

次の`create-role-alias`例では、指定されたロール`LightBulbRole`に というロールエイリアスを作成します。

```
aws iot create-role-alias \  
  --role-alias LightBulbRole \  
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

出力:

```
{
  "roleAlias": "LightBulbRole",
  "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
```



```
}
```

詳細については、AWS IoT APIリファレンスの[CreateRoleAlias](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateRoleAlias](#)」の「」を参照してください。

create-scheduled-audit

次の例は、create-scheduled-audit を使用する方法を説明しています。

AWS CLI

スケジュールされた監査を作成するには

次のcreate-scheduled-audit例では、CA 証明書またはデバイス証明書の有効期限が切れているかどうかを確認するために、毎週水曜日に実行されるスケジュールされた監査を作成します。

```
aws iot create-scheduled-audit \  
  --scheduled-audit-name WednesdayCertCheck \  
  --frequency WEEKLY \  
  --day-of-week WED \  
  --target-check-  
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK
```

出力:

```
{  
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/  
WednesdayCertCheck"  
}
```

詳細については、AWS「IoT デベロッパーガイド」の「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateScheduledAudit](#)」の「」を参照してください。

create-security-profile

次の例は、create-security-profile を使用する方法を説明しています。

AWS CLI

セキュリティプロファイルを作成するには

次のcreate-security-profile例では、セルラー帯域幅がしきい値を超えているか、または5分以内に10件を超える認証エラーが発生したかを確認するセキュリティプロファイルを作成します。

```
aws iot create-security-profile \
  --security-profile-name PossibleIssue \
  --security-profile-description "Check to see if authorization fails 10 times in 5 minutes or if cellular bandwidth exceeds 128" \
  --behaviors "[{\\"name\\":\\"CellularBandwidth\\",\\"metric\\":\\"aws:message-byte-size\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":128},\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}},{\\"name\\":\\"Authorization\\",\\"metric\\":\\"aws:num-authorization-failures\\",\\"criteria\\":{\\"comparisonOperator\\":\\"less-than\\",\\"value\\":{\\"count\\":10},\\"durationSeconds\\":300,\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}}]"
```

出力:

```
{
  "securityProfileName": "PossibleIssue",
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/PossibleIssue"
}
```

詳細については、AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateSecurityProfile](#)」の「」を参照してください。

create-stream

次の例は、create-stream を使用する方法を説明しています。

AWS CLI

1 つ以上の大きなファイルをチャンクで配信するためのストリームを作成するには MQTT

次のcreate-stream例では、 を介してチャンクで1つ以上の大きなファイルを配信するためのストリームを作成しますMQTT。ストリームは、S3 などのソースからのMQTTメッセージとして

パッケージ化されたチャンクまたはブロックでデータバイトを転送します。1 つ以上のファイルをストリームに関連付けることができます。

```
aws iot create-stream \  
  --cli-input-json file://create-stream.json
```

create-stream.json の内容:

```
{  
  "streamId": "stream12345",  
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
  "files": [  
    {  
      "fileId": 123,  
      "s3Location": {  
        "bucket": "codesign-ota-bucket",  
        "key": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"  
      }  
    }  
  ],  
  "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"  
}
```

出力:

```
{  
  "streamId": "stream12345",  
  "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",  
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
  "streamVersion": "1"  
}
```

詳細については、AWS IoT APIリファレンスの[CreateStream](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateStream](#)」の「」を参照してください。

create-thing-group

次のコード例は、create-thing-group を使用する方法を示しています。

AWS CLI

例 1: モノのグループを作成するには

次のcreate-thing-group例では、説明と 2 つの属性LightBulbsを持つ という名前のモノのグループを作成します。

```
aws iot create-thing-group \  
  --thing-group-name LightBulbs \  
  --thing-group-properties "thingGroupDescription=\"Generic bulb group\"",  
  attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

出力:

```
{  
  "thingGroupName": "LightBulbs",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",  
  "thingGroupId": "9198bf9f-1e76-4a88-8e8c-e7140142c331"  
}
```

例 2: 親グループの一部であるモノのグループを作成するには

次に、 という名前HalogenBulbsの親モノグループを持つ という名前のモノグループcreate-thing-groupを作成しますLightBulbs。

```
aws iot create-thing-group \  
  --thing-group-name HalogenBulbs \  
  --parent-group-name LightBulbs
```

出力:

```
{  
  "thingGroupName": "HalogenBulbs",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",  
  "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e"  
}
```

詳細については、[「IoT デベロッパーガイド」の「モノのグループAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateThingGroup](#)」の「」を参照してください。

create-thing-type

次のコード例は、create-thing-type を使用する方法を示しています。

AWS CLI

モノのタイプを定義するには

次のcreate-thing-type例では、モノのタイプと関連する属性を定義します。

```
aws iot create-thing-type \  
  --thing-type-name "LightBulb" \  
  --thing-type-properties "thingTypeDescription=light bulb type,  
  searchableAttributes=wattage,model"
```

出力:

```
{  
  "thingTypeName": "LightBulb",  
  "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",  
  "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190"  
}
```

詳細については、IoT デベロッパーガイドの「[モノのタイプ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateThingType](#)」の「」を参照してください。

create-thing

次の例は、create-thing を使用する方法を説明しています。

AWS CLI

例 1: レジストリにモノのレコードを作成するには

次のcreate-thing例では、AWS IoT モノのレジストリにデバイスのエントリを作成します。

```
aws iot create-thing \  

```

```
--thing-name SampleIoTThing
```

出力:

```
{
  "thingName": "SampleIoTThing",
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/SampleIoTThing",
  "thingId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE "
}
```

例 2: モノのタイプに関連付けられているモノを定義するには

次のcreate-thing例では、指定されたモノのタイプとその属性を持つモノを作成します。

```
aws iot create-thing \
  --thing-name "MyLightBulb" \
  --thing-type-name "LightBulb" \
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

出力:

```
{
  "thingName": "MyLightBulb",
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
  "thingId": "40da2e73-c6af-406e-b415-15acae538797"
}
```

詳細については、IoT デベロッパーガイドの「[レジストリとモノのタイプでモノを管理する方法](https://docs.aws.amazon.com/iot/latest/developerguide/thing-types.html)」を参照してください。 <https://docs.aws.amazon.com/iot/latest/developerguide/thing-types.html> AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateThing](#)」の「」を参照してください。

create-topic-rule-destination

次の例は、create-topic-rule-destination を使用する方法を説明しています。

AWS CLI

トピックルールの送信先を作成するには

次のcreate-topic-rule-destination例では、HTTPエンドポイントのトピックルールの送信先を作成します。

```
aws iot create-topic-rule-destination \  
  --destination-configuration httpUrlConfiguration={confirmationUrl=https://  
example.com}
```

出力:

```
{  
  "topicRuleDestination": {  
    "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "status": "IN_PROGRESS",  
    "statusReason": "Awaiting confirmation. Confirmation message sent on  
2020-07-09T22:47:54.154Z; no response received from the endpoint.",  
    "httpUrlProperties": {  
      "confirmationUrl": "https://example.com"  
    }  
  }  
}
```

詳細については、AWS IoT デベロッパーガイドの「[トピックルールの送信先の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTopicRuleDestination](#)」の「」を参照してください。

create-topic-rule

次のコード例は、create-topic-rule を使用する方法を示しています。

AWS CLI

Amazon SNSアラートを送信するルールを作成するには

次のcreate-topic-rule例では、デバイスシャドウにある土壌湿度レベルの測定値が低い場合に Amazon SNS メッセージを送信するルールを作成します。

```
aws iot create-topic-rule \  
  --rule-name "LowMoistureRule" \  
  --topic-rule-destination-arn arn:aws:iot:us-west-2:123456789012:ruledestination/http/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

```
--topic-rule-payload file://plant-rule.json
```

この例では、次のJSONコードを という名前のファイルに保存する必要がありますplant-rule.json。

```
{
  "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE
state.reported.moisture = 'low'\n",
  "description": "Sends an alert whenever soil moisture level readings are too
low.",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
    "sns": {
      "targetArn": "arn:aws:sns:us-
west-2:123456789012:MyRPiLowMoistureTopic",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/
MyRPiLowMoistureTopicRole",
      "messageFormat": "RAW"
    }
  ]
}
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT ルールの作成AWS IoT」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTopicRule](#)」の「」を参照してください。

delete-account-audit-configuration

次の例は、delete-account-audit-configuration を使用する方法を説明しています。

AWS CLI

AWS アカウントのすべての監査チェックを無効にするには

次のdelete-account-audit-configuration例では、このアカウントの AWS IoT Device Defender のデフォルト設定を復元し、すべての監査チェックを無効にして設定データを消去しま

す。また、このアカウントのスケジュールされた監査も削除されます。このコマンドは慎重に使用してください。

```
aws iot delete-account-audit-configuration \  
  --delete-scheduled-audits
```

このコマンドでは何も出力されません。

詳細については、AWS 「IoT デベロッパーガイド」の「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccountAuditConfiguration](#)」の「」を参照してください。

delete-audit-suppression

次のコード例は、delete-audit-suppression を使用する方法を示しています。

AWS CLI

監査結果の抑制を削除するには

次のdelete-audit-suppression例では、DEVICE_CERTIFICATE_EXPIRING_ の監査結果抑制を削除しますCHECK。

```
aws iot delete-audit-suppression \  
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \  
  --resource-identifier deviceCertificateId="c7691e<shortened>"
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの「[監査結果の抑制](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAuditSuppression](#)」の「」を参照してください。

delete-authorizer

次の例は、delete-authorizer を使用する方法を説明しています。

AWS CLI

カスタムオーソライザーを削除するには

次のdelete-authorizer例では、という名前のオーソライザーを削除しますCustomAuthorizer。カスタムオーソライザーは、削除する前に INACTIVE状態にある必要があります。

```
aws iot delete-authorizer \  
  --authorizer-name CustomAuthorizer
```

このコマンドでは何も出力されません。

詳細については、AWS 「IoT デベロッパーガイド[DeleteAuthorizer](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAuthorizer](#)」の「」を参照してください。

delete-billing-group

次のコード例は、delete-billing-group を使用する方法を示しています。

AWS CLI

請求グループを削除するには

次のdelete-billing-group例では、指定された請求グループを削除します。請求グループには 1 つ以上のモノが含まれている場合でも、削除できます。

```
aws iot delete-billing-group \  
  --billing-group-name BillingGroupTwo
```

このコマンドでは、出力が生成されません。

詳細については、IoT デベロッパーガイドの「[請求グループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBillingGroup](#)」の「」を参照してください。

delete-ca-certificate

次の例は、delete-ca-certificate を使用する方法を説明しています。

AWS CLI

CA 証明書を削除するには

次のdelete-ca-certificate例では、指定された証明書 ID を持つ CA 証明書を削除します。

```
aws iot delete-ca-certificate \  
  --certificate-  
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

このコマンドでは何も出力されません。

詳細については、IoT リファレンスの「[DeleteCACertificate](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCaCertificate](#)」の「」を参照してください。

delete-certificate

次の例は、delete-certificate を使用する方法を説明しています。

AWS CLI

デバイス証明書を削除するには

次のdelete-certificate例では、指定された ID のデバイス証明書を削除します。

```
aws iot delete-certificate \  
  --certificate-  
  id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9
```

このコマンドでは何も出力されません。

詳細については、AWS IoT APIリファレンスの[DeleteCertificate](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCertificate](#)」の「」を参照してください。

delete-custom-metric

次の例は、delete-custom-metric を使用する方法を説明しています。

AWS CLI

カスタムメトリクスを削除するには

次のdelete-custom-metric例では、カスタムメトリクスを削除します。

```
aws iot delete-custom-metric \  
  --metric-name batteryPercentage \  
  --region us-east-1
```

出力:

```
HTTP 200
```

詳細については、「IoT Core デベロッパーガイド」の「[カスタムメトリクス](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCustomMetric](#)」の「」を参照してください。

delete-dimension

次の例は、delete-dimension を使用する方法を説明しています。

AWS CLI

ディメンションを削除するには

次のdelete-dimension例では、`TopicFilterForAuthMessages` というディメンションを削除します。

```
aws iot delete-dimension \  
  --name TopicFilterForAuthMessages
```

このコマンドでは何も出力されません。

詳細については、「AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDimension](#)」の「」を参照してください。

delete-domain-configuration

次の例は、delete-domain-configuration を使用する方法を説明しています。

AWS CLI

ドメイン設定を削除するには

次のdelete-domain-configuration例では、 という名前のドメイン設定を AWS アカウントadditionalDataDomainから削除します。

```
aws iot delete-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --domain-configuration-status "OK"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの [「設定可能なエンドポイント」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDomainConfiguration](#)」の「」を参照してください。

delete-dynamic-thing-group

次の例は、delete-dynamic-thing-group を使用する方法を説明しています。

AWS CLI

動的モノのグループを削除するには

次のdelete-dynamic-thing-group例では、指定された動的モノのグループを削除します。

```
aws iot delete-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm"
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの [「Dynamic Thing Groups」](#) を参照してください。

AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDynamicThingGroup](#)」の「」を参照してください。

delete-job-execution

次のコード例は、delete-job-execution を使用する方法を示しています。

AWS CLI

ジョブ実行を削除するには

次のdelete-job-execution例では、デバイス上の指定されたジョブのジョブ実行を削除します。describe-job-execution を使用して実行番号を取得します。

```
aws iot delete-job-execution
  --job-id "example-job-02"
  --thing-name "MyRaspberryPi"
  --execution-number 1
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteJobExecution](#)」の「」を参照してください。

delete-job

次の例は、delete-job を使用する方法を説明しています。

AWS CLI

ジョブを削除するには

次のdelete-job例では、指定されたジョブを削除します。--force オプションを指定することで、ステータスが IN_PROGRESS であってもジョブは削除されます。

```
aws iot delete-job \
  --job-id "example-job-04" \
  --force
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteJob](#)」の「」を参照してください。

delete-mitigation-action

次のコード例は、delete-mitigation-action を使用する方法を示しています。

AWS CLI

緩和アクションを削除するには

次のdelete-mitigation-action例では、指定された緩和アクションを削除します。

```
aws iot delete-mitigation-action \  
  --action-name AddThingsToQuarantineGroup1Action
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの[DeleteMitigationAction](#) 「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMitigationAction](#)」の「」を参照してください。

delete-ota-update

次のコード例は、delete-ota-update を使用する方法を示しています。

AWS CLI

OTA 更新を削除するには

次のdelete-ota-update例では、指定されたOTA更新を削除します。

```
aws iot delete-ota-update \  
  --ota-update-id ota12345 \  
  --delete-stream \  
  --force-delete-aws-job
```

このコマンドでは何も出力されません。

詳細については、IoT リファレンスの「[DeleteOTAUpdate](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteOtaUpdate](#)」の「」を参照してください。

delete-policy-version

次の例は、delete-policy-version を使用する方法を説明しています。

AWS CLI

ポリシーのバージョンを削除するには

次のdelete-policy-version例では、指定されたポリシーのバージョン 2 を AWS アカウントから削除します。

```
aws iot delete-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT ポリシー」を参照してください。

AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicyVersion](#)」の「」を参照してください。

delete-policy

次のコード例は、delete-policy を使用する方法を示しています。

AWS CLI

ポリシーを削除するには

次のdelete-policy例では、指定されたポリシーを AWS アカウントから削除します。

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```


このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT ポリシーAWS IoT」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicy](#)」の「」を参照してください。

delete-provisioning-template-version

次のコード例は、delete-provisioning-template-version を使用する方法を示しています。

AWS CLI

プロビジョニングテンプレートのバージョンを削除するには

次のdelete-provisioning-template-version例では、指定されたプロビジョニングテンプレートのバージョン 2 を削除します。

```
aws iot delete-provisioning-template-version \  
  --version-id 2 \  
  --template-name "widget-template"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT Core デベロッパーガイド」](#)の「IoT Secure Tunneling」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProvisioningTemplateVersion](#)」の「」を参照してください。

delete-provisioning-template

次の例は、delete-provisioning-template を使用する方法を説明しています。

AWS CLI

プロビジョニングテンプレートを削除するには

次のdelete-provisioning-template例では、指定されたプロビジョニングテンプレートを削除します。

```
aws iot delete-provisioning-template \  
  --template-name widget-template
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT Core デベロッパーガイド」の「IoT Secure Tunneling」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProvisioningTemplate](#)」の「」を参照してください。

delete-registration-code

次の例は、delete-registration-code を使用する方法を説明しています。

AWS CLI

登録コードを削除するには

次のdelete-registration-code例では、AWS IoT アカウント固有の登録コードを削除します。

```
aws iot delete-registration-code
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの[「独自の証明書を使用する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRegistrationCode](#)」の「」を参照してください。

delete-role-alias

次の例は、delete-role-alias を使用する方法を説明しています。

AWS CLI

AWS IoT ロールエイリアスを削除するには

次のdelete-role-alias例では、`LightBulbRole` という名前の AWS IoT ロールエイリアスを削除します。

```
aws iot delete-role-alias \  
  --role-alias LightBulbRole
```

このコマンドでは何も出力されません。

詳細については、AWS「IoT デベロッパーガイド」の[AWS「サービスへの直接呼び出しの承認」](#)を参照してください。

- API 詳細については、「コマンドリファレンス[DeleteRoleAlias](#)」の「」を参照してください。
AWS CLI

delete-scheduled-audit

次の例は、delete-scheduled-audit を使用する方法を説明しています。

AWS CLI

スケジュールされた監査を削除するには

次のdelete-scheduled-audit例では、という名前の AWS IoT Device Defender スケジュールされた監査を削除しますAWSIoTDeviceDefenderDailyAudit。

```
aws iot delete-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

このコマンドでは何も出力されません。

詳細については、AWS「IoT デベロッパーガイド」の[「監査コマンド」](#)を参照してください。

- API 詳細については、「コマンドリファレンス[DeleteScheduledAudit](#)」の「」を参照してください。
AWS CLI

delete-security-profile

次の例は、delete-security-profile を使用する方法を説明しています。

AWS CLI

セキュリティプロファイルを削除するには

次のdelete-security-profile例では、という名前のセキュリティプロファイルを削除しますPossibleIssue。

```
aws iot delete-security-profile \  
  --security-profile-name PossibleIssue
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSecurityProfile](#)」の「」を参照してください。

delete-stream

次の例は、delete-stream を使用する方法を説明しています。

AWS CLI

ストリームを削除するには

次のdelete-stream例では、指定されたストリームを削除します。

```
aws iot delete-stream \  
  --stream-id stream12345
```

このコマンドでは何も出力されません。

詳細については、AWS IoT APIリファレンスの[DeleteStream](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStream](#)」の「」を参照してください。

delete-thing-group

次のコード例は、delete-thing-group を使用する方法を示しています。

AWS CLI

モノのグループを削除するには

次のdelete-thing-group例では、指定されたモノのグループを削除します。モノの子グループが含まれている場合、モノのグループを削除することはできません。

```
aws iot delete-thing-group \  
  --thing-group-name stream12345
```

```
--thing-group-name DefectiveBulbs
```

このコマンドでは何も出力されません。

詳細については、[「IoT デベロッパーガイド」の「モノのグループAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteThingGroup](#)」の「」を参照してください。

delete-thing-type

次の例は、delete-thing-type を使用する方法を説明しています。

AWS CLI

例 1: モノのタイプを削除するには

次のdelete-thing-type例では、非推奨のモノのタイプを削除します。

```
aws iot delete-thing-type \  
  --thing-type-name "obsoleteThingType"
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの[「モノのタイプ」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteThingType](#)」の「」を参照してください。

delete-thing

次のコード例は、delete-thing を使用する方法を示しています。

AWS CLI

モノに関する詳細情報を表示するには

次のdelete-thing例では、AWS アカウントの AWS IoT レジストリからモノを削除します。

```
aws iot delete-thing --thing-nameFourthBulb 「」
```

このコマンドでは何も出力されません。

詳細については、「AWS IoT デベロッパーガイド」の「[レジストリによるモノの管理方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteThing](#)」の「」を参照してください。

delete-topic-rule-destination

次のコード例は、delete-topic-rule-destination を使用する方法を示しています。

AWS CLI

トピックルールの送信先を削除するには

次のdelete-topic-rule-destination例では、指定されたトピックルールの送信先を削除します。

```
aws iot delete-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[トピックルールの送信先の削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTopicRuleDestination](#)」の「」を参照してください。

delete-topic-rule

次の例は、delete-topic-rule を使用する方法を説明しています。

AWS CLI

ルールを削除するには

次のdelete-topic-rule例では、指定されたルールを削除します。

```
aws iot delete-topic-rule \  
  --rule-name "LowMoistureRule"
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの「[ルール削除](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTopicRule](#)」の「」を参照してください。

delete-v2-logging-level

次の例は、delete-v2-logging-level を使用する方法を説明しています。

AWS CLI

モノのグループのログ記録レベルを削除するには

次のdelete-v2-logging-level例では、指定されたモノのグループのログ記録レベルを削除します。

```
aws iot delete-v2-logging-level \  
  --target-type THING_GROUP \  
  --target-name LightBulbs
```

このコマンドでは何も出力されません。

- API 詳細については、「コマンドリファレンス」の[DeleteV2LoggingLevel](#)」を参照してください。AWS CLI

deprecate-thing-type

次の例は、deprecate-thing-type を使用する方法を説明しています。

AWS CLI

例 1: モノのタイプを非推奨にするには

次のdeprecate-thing-type例では、ユーザーが新しいモノを関連付けられないように、モノのタイプを廃止しています。

```
aws iot deprecate-thing-type \  
  --thing-type-name "obsoleteThingType"
```

このコマンドでは何も出力されません。

例 2: モノのタイプの非推奨を元に戻すには

次のdeprecate-thing-type例では、モノタイプの非推奨化を取り消し、ユーザーが新しいモノを再度関連付けることができます。

```
aws iot deprecate-thing-type \  
  --thing-type-name "obsoleteThingType" \  
  --undo-deprecate
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの「[モノのタイプ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeprecateThingType](#)」の「」を参照してください。

describe-account-audit-configuration

次のコード例は、describe-account-audit-configuration を使用する方法を示しています。

AWS CLI

現在の監査設定を表示するには

次のdescribe-account-audit-configuration例では、AWS IoT Device Defender 監査設定の現在の設定を一覧表示します。

```
aws iot describe-account-audit-configuration
```

出力:

```
{  
  "roleArn": "arn:aws:iam::123456789012:role/service-role/  
AWSIoTDeviceDefenderAudit_1551201085996",  
  "auditNotificationTargetConfigurations": {  
    "SNS": {  
      "targetArn": "arn:aws:sns:us-west-2:123456789012:ddauidits",  
      "roleArn": "arn:aws:iam::123456789012:role/service-role/  
AWSIoTDeviceDefenderAudit",  
      "enabled": true  
    }  
  }  
}
```



```
  },
  "auditCheckConfigurations": {
    "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
      "enabled": true
    },
    "CA_CERTIFICATE_EXPIRING_CHECK": {
      "enabled": true
    },
    "CONFLICTING_CLIENT_IDS_CHECK": {
      "enabled": true
    },
    "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
      "enabled": true
    },
    "DEVICE_CERTIFICATE_SHARED_CHECK": {
      "enabled": true
    },
    "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
      "enabled": true
    },
    "LOGGING_DISABLED_CHECK": {
      "enabled": true
    },
    "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
      "enabled": true
    },
    "REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK": {
      "enabled": true
    },
    "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
      "enabled": true
    }
  }
}
```

詳細については、AWS IoT デベロッパーガイドの「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountAuditConfiguration](#)」の「」を参照してください。

describe-audit-finding

次の例は、describe-audit-finding を使用する方法を説明しています。

AWS CLI

監査結果の詳細を一覧表示するには

次のdescribe-audit-finding例では、指定された AWS IoT Device Defender 監査結果の詳細を一覧表示します。監査は複数の検出結果を生成できます。list-audit-findings コマンドを使用して、監査から検出結果のリストを取得し、 を取得しますfindingId。

```
aws iot describe-audit-finding \  
  --finding-id "ef4826b8-e55a-44b9-b460-5c485355371b"
```

出力:

```
{  
  "finding": {  
    "findingId": "ef4826b8-e55a-44b9-b460-5c485355371b",  
    "taskId": "873ed69c74a9ec8fa9b8e88e9abc4661",  
    "checkName": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",  
    "taskStartTime": 1576012045.745,  
    "findingTime": 1576012046.168,  
    "severity": "CRITICAL",  
    "nonCompliantResource": {  
      "resourceType": "IOT_POLICY",  
      "resourceIdentifier": {  
        "policyVersionIdentifier": {  
          "policyName": "smp-ggrass-group_Core-policy",  
          "policyVersionId": "1"  
        }  
      }  
    },  
    "reasonForNonCompliance": "Policy allows broad access to IoT data plane  
actions: [iot:Subscribe, iot:Connect, iot:GetThingShadow, iot>DeleteThingShadow,  
iot:UpdateThingShadow, iot:Publish].",  
    "reasonForNonComplianceCode":  
    "ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS"  
  }  
}
```

詳細については、AWS IoT デベロッパーガイドの「[監査結果の確認 \(監査コマンド\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAuditFinding](#)」の「」を参照してください。

describe-audit-mitigation-actions-task

次のコード例は、describe-audit-mitigation-actions-task を使用する方法を示しています。

AWS CLI

監査緩和アクションタスクの詳細を表示するには

次のdescribe-audit-mitigation-actions-task例は、 が検出結果に適用ResetPolicyVersionActionされた、指定されたタスクの詳細を示しています。結果には、タスクの開始と終了のタイミング、ターゲットとなった結果の数(および結果)、このタスクの一部として適用されるアクションの定義が含まれます。

```
aws iot describe-audit-mitigation-actions-task \  
  --task-id ResetPolicyTask01
```

出力:

```
{  
  "taskStatus": "COMPLETED",  
  "startTime": "2019-12-10T15:13:19.457000-08:00",  
  "endTime": "2019-12-10T15:13:19.947000-08:00",  
  "taskStatistics": {  
    "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {  
      "totalFindingsCount": 1,  
      "failedFindingsCount": 0,  
      "succeededFindingsCount": 1,  
      "skippedFindingsCount": 0,  
      "canceledFindingsCount": 0  
    }  
  },  
  "target": {  
    "findingIds": [  
      "ef4826b8-e55a-44b9-b460-5c485355371b"  
    ]  
  },  
  "auditCheckToActionsMapping": {  
    "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": [  
      "ResetPolicyVersionAction"  
    ]  
  },  
  "actionsDefinition": [  
    {  
      "actionName": "ResetPolicyVersionAction",  
      "actionType": "ResetPolicyVersionAction",  
      "actionParameters": {}  
    }  
  ]  
}
```

```
{
  "name": "ResetPolicyVersionAction",
  "id": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",
  "roleArn": "arn:aws:iam::123456789012:role/service-role/
ReplacePolicyVersionRole",
  "actionParams": {
    "replaceDefaultPolicyVersionParams": {
      "templateName": "BLANK_POLICY"
    }
  }
}
```

詳細については、AWS IoT デベロッパーガイドの[DescribeAuditMitigationActionsTask](#)「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeAuditMitigationActionsTask](#)の「」を参照してください。

describe-audit-suppression

次の例は、describe-audit-suppression を使用する方法を説明しています。

AWS CLI

監査結果の抑制に関する詳細を取得するには

次のdescribe-audit-suppression例では、監査結果の抑制の詳細を一覧表示します。

```
aws iot describe-audit-task \
  --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

出力:

```
{
  "taskStatus": "COMPLETED",
  "taskType": "SCHEDULED_AUDIT_TASK",
  "taskStartTime": 1596168096.157,
  "taskStatistics": {
    "totalChecks": 1,
    "inProgressChecks": 0,
    "waitingForDataCollectionChecks": 0,
```

```
    "compliantChecks": 0,
    "nonCompliantChecks": 1,
    "failedChecks": 0,
    "canceledChecks": 0
  },
  "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
  "auditDetails": {
    "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
      "checkRunStatus": "COMPLETED_NON_COMPLIANT",
      "checkCompliant": false,
      "totalResourcesCount": 195,
      "nonCompliantResourcesCount": 2
    }
  }
}
```

詳細については、[「IoT デベロッパーガイド」の「監査結果の抑制AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAuditSuppression](#)」の「」を参照してください。

describe-audit-task

次のコード例は、describe-audit-task を使用する方法を示しています。

AWS CLI

監査インスタンスに関する情報を取得するには

次のdescribe-audit-task例では、AWS IoT Device Defender 監査のインスタンスに関する情報を取得します。監査が完了すると、実行のサマリー統計が結果に含まれます。

```
aws iot describe-audit-task \
  --task-id a3aea009955e501a31b764abe1bebd3d
```

出力:

```
{
  "taskStatus": "COMPLETED",
  "taskType": "ON_DEMAND_AUDIT_TASK",
  "taskStartTime": 1560356923.434,
```

```
"taskStatistics": {
  "totalChecks": 3,
  "inProgressChecks": 0,
  "waitingForDataCollectionChecks": 0,
  "compliantChecks": 3,
  "nonCompliantChecks": 0,
  "failedChecks": 0,
  "canceledChecks": 0
},
"auditDetails": {
  "CA_CERTIFICATE_EXPIRING_CHECK": {
    "checkRunStatus": "COMPLETED_COMPLIANT",
    "checkCompliant": true,
    "totalResourcesCount": 0,
    "nonCompliantResourcesCount": 0
  },
  "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
    "checkRunStatus": "COMPLETED_COMPLIANT",
    "checkCompliant": true,
    "totalResourcesCount": 6,
    "nonCompliantResourcesCount": 0
  },
  "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
    "checkRunStatus": "COMPLETED_COMPLIANT",
    "checkCompliant": true,
    "totalResourcesCount": 0,
    "nonCompliantResourcesCount": 0
  }
}
}
```

詳細については、AWS「IoT デベロッパーガイド」の「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeAuditTask](#)」の「」を参照してください。

describe-authorizer

次の例は、describe-authorizer を使用する方法を説明しています。

AWS CLI

カスタムオーソライザーに関する情報を取得するには

次のdescribe-authorizer例では、指定されたカスタムオーソライザーの詳細を表示します。

```
aws iot describe-authorizer \
  --authorizer-name CustomAuthorizer
```

出力:

```
{
  "authorizerDescription": {
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer",
    "authorizerFunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction",
    "tokenKeyName": "MyAuthToken",
    "tokenSigningPublicKeys": {
      "FIRST_KEY": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAAOCAQ8AMIIBCgKCAQEA1uJ0B4lQPgG/lM6ZfIwo
\nZ+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0UEq9xb503kRSmyIwTzwzm/f4Gf0Y
\nZUloJ+t3PUUwHrmbYTAGTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
\nzw0BKPeic0asNjppqT8PkBbRaKylEJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ\n
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfPOHDJQgID0XnZwAlNnZc0hCwIx\n50g2LW20y9R/
dmqtDmJiVP97Z4GyKxPvw1YHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1\nlQIDAQAB\n-----END PUBLIC
KEY-----"
    },
    "status": "ACTIVE",
    "creationDate": 1571245658.069,
    "lastModifiedDate": 1571245658.069
  }
}
```

詳細については、AWS IoT APIリファレンスの[DescribeAuthorizer](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeAuthorizer](#)」の「」を参照してください。

describe-billing-group

次の例は、describe-billing-group を使用する方法を説明しています。

AWS CLI

請求グループに関する情報を取得するには

次のdescribe-billing-group例では、指定された請求グループの情報を取得します。

```
aws iot describe-billing-group --billing-group-name GroupOne
```

出力:

```
{
  "billingGroupName": "GroupOne",
  "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562",
  "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
  "version": 1,
  "billingGroupProperties": {},
  "billingGroupMetadata": {
    "creationDate": 1560199355.378
  }
}
```

詳細については、IoT デベロッパーガイドの「[請求グループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeBillingGroup](#)」の「」を参照してください。

describe-ca-certificate

次の例は、describe-ca-certificate を使用する方法を説明しています。

AWS CLI

CA 証明書の詳細を取得するには

次のdescribe-ca-certificate例では、指定された CA 証明書の詳細を表示します。

```
aws iot describe-ca-certificate \
  --certificate-
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

出力:

```
{
  "certificateDescription": {
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
```



```

    "certificateId":
      "f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "status": "INACTIVE",
      "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIICzzCCAbegEXAMPLEJANVEPW18taPMA0GCSqGSIb3DQEBBQUAMB4xCzAJBgNV
\nBAYTA1VMTQ8wDQYDVQQKDAZBbWF6b24wHhcNMtkwOTI0MjEzMTU1WhcNMjkwOTIx
\nMjEzMTU1WjAeMQswCQYDVQQGEwJVUzEPMA0GA1UECgwW1hem9uMIIIBIjANBgkq
\nhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAZd3R3ioa1CS0MhFWfBrVGR036EK07UAF
\nVdz9EXAMPLE1VczICbADnATK522kEIB51/18Vz1FtAhQL5V5eybXKnB7QebNer5m
\n4Yibx7shR5oqNzFsrXWxuugN5+w5gEfqNMaw0jhF4Lscu1KG49yuqjcDU19/13ua
\n3B2gxs1Pe7TiWwvUskzxb01F2WCshbEjvqY8fIwtGYCjTeJAgQ9hvZx/69XhKen
\nwV9LJw0QxrsUS0Ty8IHwbB8fRy72VM3u7fJoaU+n04jd5cqaoEPtzoEUFEXAMPLE
\nyVAJpqHwgbYbcUfn7V+AB6yh1+0Fa1rEQGuZDPGyJslxwr5vh8nRewIDAQABoxAw
\nDjAMBgNVHRMBETADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA+3a5CV3IJg0nd0AgI
\nBgVMtmYzTvqAngx26aG9/spvCjXckh2SBF+EcB1CFwH1yakwjJL1dR4yarnrfxgI
\nEqP4A0YVimAVoQ5FBwnloHe16+3qtDib1U9DeXBUCtS55EcfrEXAMPLEYtXdqU5C
\nU9ia4KAjV0dxW1+EFYmWx5eGeb0gDTNHBy1V6B/f0SZiQAwDYp4x3B+gAP+a/bWB
\nlum0qtBdWe6L6/83L+JhaTByqV25iVJ4c/UZUnG8926wU1DM9zQvEXuEVvzZ7+m\n4PSNqst/
\nV0vnLpoG4e0WgcJgANuB33CSwtjWSuYsbhmqQRknGhREXAMPLEZT4fm\nfo0e\n-----END
CERTIFICATE-----\n",
      "ownedBy": "123456789012",
      "creationDate": 1569365372.053,
      "autoRegistrationStatus": "DISABLE",
      "lastModifiedDate": 1569365372.053,
      "customerVersion": 1,
      "generationId": "c5c2eb95-140b-4f49-9393-6aaac85b2a90",
      "validity": {
        "notBefore": 1569360675.0,
        "notAfter": 1884720675.0
      }
    }
  }
}

```

詳細については、IoT リファレンスの「[DescribeCACertificate](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCaCertificate](#)」の「」を参照してください。

describe-certificate

次の例は、describe-certificate を使用する方法を説明しています。

AWS CLI

証明書に関する情報を取得するには

次のdescribe-certificate例では、指定された証明書の詳細を表示します。

```
aws iot describe-certificate \  
  --certificate-  
id "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

出力:

```
{  
  "certificateDescription": {  
    "certificateArn": "arn:aws:iot:us-  
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",  
    "certificateId":  
    "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",  
    "status": "ACTIVE",  
    "certificatePem": "-----BEGIN CERTIFICATE-----  
MIICiTEXAMPLEQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC  
VVMxCzAJBgNVBEXAMPLEMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAsTC0lBTSBDEXAMPLElMRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAd  
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5EXAMPLEcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNEXAMPLEdBMRAwDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBEXAMPLEz  
b2xEXAMPLEYDVQQDEwLUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft  
YXpvbi5jb20wgZ8EXAMPLEZIHvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySWtC2XADZ4nB+BLYEXAMPLEpiwsZ3G93vUEI03IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7EXAMPLEGBzZswY6786m86gpE  
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFEXAMPLEAtCu4  
nUhVVxYUnEXAMPLE8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJI1J00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=  
-----END CERTIFICATE-----",  
    "ownedBy": "123456789012",  
    "creationDate": 1541022751.983,  
    "lastModifiedDate": 1541022751.983,  
    "customerVersion": 1,  
    "transferData": {},  
    "generationId": "6974fbcd-2e61-4114-bc5e-4204cc79b045",  
    "validity": {  
      "notBefore": 1541022631.0,  

```

```
        "notAfter": 2524607999.0
      }
    }
  }
```

詳細については、AWS IoT APIリファレンスの[DescribeCertificate](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeCertificate](#)」の「」を参照してください。

describe-custom-metric

次のコード例は、describe-custom-metric を使用する方法を示しています。

AWS CLI

Device Defender カスタムメトリクスに関する情報を取得するには

次のdescribe-custom-metric例では、`myCustomMetric` という名前のカスタムメトリクスに関する情報を取得します。

```
aws iot describe-custom-metric \
  --metric-name myCustomMetric
```

出力:

```
{
  "metricName": "myCustomMetric",
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/myCustomMetric",
  "metricType": "number",
  "displayName": "My custom metric",
  "creationDate": 2020-11-17T23:02:12.879000-09:00,
  "lastModifiedDate": 2020-11-17T23:02:12.879000-09:00
}
```

詳細については、「IoT Core デベロッパーガイド」の[「カスタムメトリクス」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeCustomMetric](#)」の「」を参照してください。

describe-default-authorizer

次の例は、`describe-default-authorizer` を使用する方法を説明しています。

AWS CLI

デフォルトのカスタムオーソライザーに関する情報を取得するには

次の`describe-default-authorizer`例では、デフォルトのカスタムオーソライザーの詳細を表示します。

```
aws iot describe-default-authorizer
```

出力:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

詳細については、AWS IoT APIリファレンスの[DescribeDefaultAuthorizer](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDefaultAuthorizer](#)」の「」を参照してください。

describe-dimension

次の例は、`describe-dimension` を使用する方法を説明しています。

AWS CLI

ディメンションに関する情報を取得するには

次の`describe-dimension`例では、`TopicFilterForAuthMessages` という名前のディメンションに関する情報を取得します。

```
aws iot describe-dimension \
  --name TopicFilterForAuthMessages
```

出力:

```
{
  "name": "TopicFilterForAuthMessages",
  "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
  "type": "TOPIC_FILTER",
  "stringValues": [
    "device/+/auth"
  ],
  "creationDate": 1578620223.255,
  "lastModifiedDate": 1578620223.255
}
```

詳細については、AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDimension](#)」の「」を参照してください。

describe-domain-configuration

次のコード例は、describe-domain-configuration を使用する方法を示しています。

AWS CLI

ドメイン設定を記述するには

次のdescribe-domain-configuration例では、指定されたドメイン設定の詳細を表示します。

```
aws iot describe-domain-configuration \
  --domain-configuration-name "additionalDataDomain"
```

出力:

```
{
  "domainConfigurationName": "additionalDataDomain",
  "domainConfigurationArn": "arn:aws:iot:us-
east-1:758EXAMPLE143:domainconfiguration/additionalDataDomain/norpw",
  "domainName": "d055exampleed74y71zfd-ats.beta.us-east-1.iot.amazonaws.com",
  "serverCertificates": [],
  "domainConfigurationStatus": "ENABLED",
  "serviceType": "DATA",
  "domainType": "AWS_MANAGED",
```

```
"lastStatusChangeDate": 1601923783.774
}
```

詳細については、AWS IoT デベロッパーガイドの「[設定可能なエンドポイント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDomainConfiguration](#)」の「」を参照してください。

describe-endpoint

次の例は、describe-endpoint を使用する方法を説明しています。

AWS CLI

例 1: 現在の AWS エンドポイントを取得するには

次のdescribe-endpoint例では、すべてのコマンドが適用されるデフォルトの AWS エンドポイントを取得します。

```
aws iot describe-endpoint
```

出力:

```
{
  "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"
}
```

詳細については、AWS 「IoT デベロッパーガイド[DescribeEndpoint](#)」の「」を参照してください。

例 2: ATS エンドポイントを取得するには

次のdescribe-endpoint例では、Amazon Trust Services (ATS) エンドポイントを取得します。

```
aws iot describe-endpoint \
  --endpoint-type iot:Data-ATS
```

出力:

```
{
  "endpointAddress": "abc123defghijk-ats.iot.us-west-2.amazonaws.com"
}
```

詳細については、AWS IoT デベロッパーガイドの「[X.509 証明書と AWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEndpoint](#)」の「」を参照してください。

describe-event-configurations

次の例は、describe-event-configurations を使用する方法を説明しています。

AWS CLI

公開されるイベントタイプを表示するには

次のdescribe-event-configurations例では、何かを追加、更新、または削除したときに生成されるイベントを制御する設定を一覧表示します。

```
aws iot describe-event-configurations
```

出力:

```
{
  "eventConfigurations": {
    "CA_CERTIFICATE": {
      "Enabled": false
    },
    "CERTIFICATE": {
      "Enabled": false
    },
    "JOB": {
      "Enabled": false
    },
    "JOB_EXECUTION": {
      "Enabled": false
    },
    "POLICY": {
      "Enabled": false
    },
  },
}
```

```
    "THING": {
      "Enabled": false
    },
    "THING_GROUP": {
      "Enabled": false
    },
    "THING_GROUP_HIERARCHY": {
      "Enabled": false
    },
    "THING_GROUP_MEMBERSHIP": {
      "Enabled": false
    },
    "THING_TYPE": {
      "Enabled": false
    },
    "THING_TYPE_ASSOCIATION": {
      "Enabled": false
    }
  }
}
```

詳細については、AWS IoT デベロッパーガイドの「[イベントメッセージ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEventConfigurations](#)」の「」を参照してください。

describe-index

次の例は、describe-index を使用する方法を説明しています。

AWS CLI

モノインデックスの現在のステータスを取得するには

次のdescribe-index例では、モノインデックスの現在のステータスを取得します。

```
aws iot describe-index \  
  --index-name "AWS_Things"
```

出力:

```
{
```



```
"indexName": "AWS_Things",
"indexStatus": "ACTIVE",
"schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

詳細については、AWS IoT デベロッパーガイドの「[モノのインデックス作成の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeIndex](#)」の「」を参照してください。

describe-job-execution

次の例は、describe-job-execution を使用する方法を説明しています。

AWS CLI

デバイス上のジョブの実行の詳細を取得するには

次のdescribe-job-execution例では、指定されたジョブの実行の詳細を取得します。

```
aws iot describe-job-execution \
  --job-id "example-job-01" \
  --thing-name "MyRaspberryPi"
```

出力:

```
{
  "execution": {
    "jobId": "example-job-01",
    "status": "QUEUED",
    "statusDetails": {},
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi",
    "queuedAt": 1560787023.636,
    "lastUpdatedAt": 1560787023.636,
    "executionNumber": 1,
    "versionNumber": 1
  }
}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeJobExecution](#)」の「」を参照してください。

describe-job

次の例は、describe-job を使用する方法を説明しています。

AWS CLI

ジョブの詳細なステータスを取得するには

次のdescribe-job例では、ID が のジョブの詳細ステータスを取得しますexample-job-01。

```
aws iot describe-job \  
  --job-id "example-job-01"
```

出力:

```
{  
  "job": {  
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",  
    "jobId": "example-job-01",  
    "targetSelection": "SNAPSHOT",  
    "status": "IN_PROGRESS",  
    "targets": [  
      "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"  
    ],  
    "description": "example job test",  
    "presignedUrlConfig": {},  
    "jobExecutionsRolloutConfig": {},  
    "createdAt": 1560787022.733,  
    "lastUpdatedAt": 1560787026.294,  
    "jobProcessDetails": {  
      "numberOfCanceledThings": 0,  
      "numberOfSucceededThings": 0,  
      "numberOfFailedThings": 0,  
      "numberOfRejectedThings": 0,  
      "numberOfQueuedThings": 1,  
      "numberOfInProgressThings": 0,  
      "numberOfRemovedThings": 0,  
      "numberOfTimedOutThings": 0  
    },  
    "timeoutConfig": {}  
  }  
}
```

```
}  
}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeJob](#)」の「」を参照してください。

describe-mitigation-action

次のコード例は、describe-mitigation-action を使用する方法を示しています。

AWS CLI

定義された緩和アクションの詳細を表示するには

次のdescribe-mitigation-action例では、指定された緩和アクションの詳細を表示します。

```
aws iot describe-mitigation-action \  
  --action-name AddThingsToQuarantineGroupAction
```

出力:

```
{  
  "actionName": "AddThingsToQuarantineGroupAction",  
  "actionType": "ADD_THINGS_TO_THING_GROUP",  
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/  
AddThingsToQuarantineGroupAction",  
  "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa",  
  "roleArn": "arn:aws:iam::123456789012:role/service-role/  
MoveThingsToQuarantineGroupRole",  
  "actionParams": {  
    "addThingsToThingGroupParams": {  
      "thingGroupNames": [  
        "QuarantineGroup1"  
      ],  
      "overrideDynamicGroups": true  
    }  
  },  
  "creationDate": "2019-12-10T11:09:35.999000-08:00",
```

```
"lastModifiedDate": "2019-12-10T11:09:35.999000-08:00"
}
```

詳細については、AWS IoT デベロッパーガイドの[DescribeMitigationAction](#)「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeMitigationAction](#)」の「」を参照してください。

describe-provisioning-template-version

次の例は、describe-provisioning-template-version を使用する方法を説明しています。

AWS CLI

プロビジョニングテンプレートのバージョンを記述するには

次のdescribe-provisioning-template-version例では、プロビジョニングテンプレートのバージョンについて説明します。

```
aws iot describe-provisioning-template-version \
  --template-name MyTestProvisioningTemplate \
  --version-id 1
```

出力:

```
{
  "versionId": 1,
  "creationDate": 1589308310.574,
  "templateBody": "{
    \"Parameters\":{
      \"SerialNumber\":{
        \"Type\": \"String\"
      },
      \"AWS::IoT::Certificate::Id\":{
        \"Type\": \"String\"
      }
    },
    \"Resources\":{
      \"certificate\":{
        \"Properties\":{
          \"CertificateId\":{
```

```

        \ "Ref\":"\ "AWS::IoT::Certificate::Id\"
      },
      \ "Status\":"\ "Active\"
    },
    \ "Type\":"\ "AWS::IoT::Certificate\"
  },
  \ "policy\":{
    \ "Properties\":{
      \ "PolicyName\":"\ "MyIotPolicy\"
    },
    \ "Type\":"\ "AWS::IoT::Policy\"
  },
  \ "thing\":{
    \ "OverrideSettings\":{
      \ "AttributePayload\":"\ "MERGE\",
      \ "ThingGroups\":"\ "DO_NOTHING\",
      \ "ThingTypeName\":"\ "REPLACE\"
    },
    \ "Properties\":{
      \ "AttributePayload\":{ },
      \ "ThingGroups\":[ ],
      \ "ThingName\":{
        \ "Fn::Join\":[
          \ "\",
          [
            \ "DemoGroup_\",
            { \ "Ref\":"\ "SerialNumber\" }
          ]
        ]
      },
      \ "ThingTypeName\":"\ "VirtualThings\"
    },
    \ "Type\":"\ "AWS::IoT::Thing\"
  }
}
}
}
",
"isDefaultVersion": true
}

```

詳細については、[「IoT Core デベロッパーガイド」の「フリートプロビジョニングを使用したデバイス証明書がないデバイスのプロビジョニング」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeProvisioningTemplateVersion](#)」の「」を参照してください。

describe-provisioning-template

次のコード例は、describe-provisioning-template を使用する方法を示しています。

AWS CLI

プロビジョニングテンプレートを記述するには

次のdescribe-provisioning-template例では、プロビジョニングテンプレートについて説明します。

```
aws iot describe-provisioning-template \  
--template-name MyTestProvisioningTemplate
```

出力:

```
{  
  "templateArn": "arn:aws:iot:us-west-2:57EXAMPLE833:provisioningtemplate/  
MyTestProvisioningTemplate",  
  "templateName": "MyTestProvisioningTemplate",  
  "creationDate": 1589308310.574,  
  "lastModifiedDate": 1589308345.539,  
  "defaultVersionId": 1,  
  "templateBody": "{  
    \"Parameters\":{  
      \"SerialNumber\":{  
        \"Type\":\"String\"  
      },  
      \"AWS::IoT::Certificate::Id\":{  
        \"Type\":\"String\"  
      }  
    },  
    \"Resources\":{  
      \"certificate\":{  
        \"Properties\":{  
          \"CertificateId\":{  
            \"Ref\":\"AWS::IoT::Certificate::Id\"  
          },  
          \"Status\":\"Active\"  
        },  
        \"Type\":\"AWS::IoT::Certificate\"  
      },  
      \"policy\":{
```

```

        \ "Properties\":{
            \ "PolicyName\":\ "MyIotPolicy\ "
        },
        \ "Type\":\ "AWS::IoT::Policy\ "
    },
    \ "thing\":{
        \ "OverrideSettings\":{
            \ "AttributePayload\":\ "MERGE\ ",
            \ "ThingGroups\":\ "DO_NOTHING\ ",
            \ "ThingTypeName\":\ "REPLACE\ "
        },
        \ "Properties\":{
            \ "AttributePayload\":{ },
            \ "ThingGroups\":[ ],
            \ "ThingName\":{
                \ "Fn::Join\":[
                    \ "\ ",
                    [
                        \ "DemoGroup_\ ",
                        { \ "Ref\":\ "SerialNumber\ " }
                    ]
                ]
            },
            \ "ThingTypeName\":\ "VirtualThings\ "
        },
        \ "Type\":\ "AWS::IoT::Thing\ "
    }
}
}
},
"enabled": true,
"provisioningRoleArn": "arn:aws:iam::571032923833:role/service-role/IoT_access"
}

```

詳細については、[「IoT Core デベロッパーガイド」の「フリートプロビジョニングを使用したデバイス証明書がないデバイスのプロビジョニング」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeProvisioningTemplate](#)」の「」を参照してください。

describe-role-alias

次のコード例は、describe-role-alias を使用する方法を示しています。

AWS CLI

AWS IoT ロールエイリアスに関する情報を取得するには

次のdescribe-role-alias例では、指定されたロールエイリアスの詳細を表示します。

```
aws iot describe-role-alias \  
  --role-alias LightBulbRole
```

出力:

```
{  
  "roleAliasDescription": {  
    "roleAlias": "LightBulbRole",  
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/  
LightBulbRole",  
    "roleArn": "arn:aws:iam::123456789012:role/light_bulb_role_001",  
    "owner": "123456789012",  
    "credentialDurationSeconds": 3600,  
    "creationDate": 1570558643.221,  
    "lastModifiedDate": 1570558643.221  
  }  
}
```

詳細については、AWS IoT APIリファレンスの[DescribeRoleAlias](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeRoleAlias](#)」の「」を参照してください。

describe-scheduled-audit

次のコード例は、describe-scheduled-audit を使用する方法を示しています。

AWS CLI

スケジュールされた監査に関する情報を取得するには

次のdescribe-scheduled-audit例では、という名前の AWS IOT Device Defender スケジュールされた監査に関する詳細情報を取得しますAWSIoTDeviceDefenderDailyAudit。

```
aws iot describe-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```


出力:

```
{
  "frequency": "DAILY",
  "targetCheckNames": [
    "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK",
    "CONFLICTING_CLIENT_IDS_CHECK",
    "DEVICE_CERTIFICATE_SHARED_CHECK",
    "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
    "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK",
    "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK"
  ],
  "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
AWSIoTDeviceDefenderDailyAudit"
}
```

詳細については、AWS「IoT デベロッパーガイド」の[「監査コマンド」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeScheduledAudit](#)」の「」を参照してください。

describe-security-profile

次の例は、describe-security-profile を使用する方法を説明しています。

AWS CLI

セキュリティプロファイルに関する情報を取得するには

次のdescribe-security-profile例では、という名前の AWS IoT Device Defender セキュリティプロファイルに関する情報を取得します。PossibleIssue.

```
aws iot describe-security-profile \
  --security-profile-name PossibleIssue
```

出力:

```
{
  "securityProfileName": "PossibleIssue",
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue",
}
```

```
"securityProfileDescription": "check to see if authorization fails 10 times in 5
minutes or if cellular bandwidth exceeds 128",
"behaviors": [
  {
    "name": "CellularBandwidth",
    "metric": "aws:message-byte-size",
    "criteria": {
      "comparisonOperator": "greater-than",
      "value": {
        "count": 128
      },
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1
    }
  },
  {
    "name": "Authorization",
    "metric": "aws:num-authorization-failures",
    "criteria": {
      "comparisonOperator": "greater-than",
      "value": {
        "count": 10
      },
      "durationSeconds": 300,
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1
    }
  }
],
"version": 1,
"creationDate": 1560278102.528,
"lastModifiedDate": 1560278102.528
}
```

詳細については、AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSecurityProfile](#)」の「」を参照してください。

describe-stream

次の例は、describe-stream を使用する方法を説明しています。

AWS CLI

ストリームに関する情報を取得するには

次のdescribe-stream例では、指定されたストリームの詳細を表示します。

```
aws iot describe-stream \  
  --stream-id stream12345
```

出力:

```
{  
  "streamInfo": {  
    "streamId": "stream12345",  
    "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",  
    "streamVersion": 1,  
    "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",  
    "files": [  
      {  
        "fileId": "123",  
        "s3Location": {  
          "bucket": "codesign-ota-bucket",  
          "key": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"  
        }  
      }  
    ],  
    "createdAt": 1557863215.995,  
    "lastUpdatedAt": 1557863215.995,  
    "roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_stream_role"  
  }  
}
```

詳細については、AWS IoT APIリファレンスの[DescribeStream](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeStream](#)」の「」を参照してください。

describe-thing-group

次の例は、describe-thing-group を使用する方法を説明しています。

AWS CLI

モノのグループに関する情報を取得するには

次のdescribe-thing-group例では、 という名前のモノのグループに関する情報を取得しますHalogenBulbs。

```
aws iot describe-thing-group \  
  --thing-group-name HalogenBulbs
```

出力:

```
{  
  "thingGroupName": "HalogenBulbs",  
  "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",  
  "version": 1,  
  "thingGroupProperties": {},  
  "thingGroupMetadata": {  
    "parentGroupName": "LightBulbs",  
    "rootToParentThingGroups": [  
      {  
        "groupName": "LightBulbs",  
        "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/  
LightBulbs"  
      }  
    ],  
    "creationDate": 1559927609.897  
  }  
}
```

詳細については、[「IoT デベロッパーガイド」の「モノのグループAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeThingGroup](#)」の「」を参照してください。

describe-thing-type

次の例は、describe-thing-type を使用する方法を説明しています。

AWS CLI

モノのタイプに関する情報を取得するには

次のdescribe-thing-type例では、AWS アカウントで定義されている指定されたモノのタイプに関する情報を表示します。

```
aws iot describe-thing-type \  
  --thing-type-name "LightBulb"
```

出力:

```
{  
  "thingTypeName": "LightBulb",  
  "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190",  
  "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",  
  "thingTypeProperties": {  
    "thingTypeDescription": "light bulb type",  
    "searchableAttributes": [  
      "model",  
      "wattage"  
    ]  
  },  
  "thingTypeMetadata": {  
    "deprecated": false,  
    "creationDate": 1559772562.498  
  }  
}
```

詳細については、IoT デベロッパーガイドの「[モノのタイプ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeThingType](#)」の「」を参照してください。

describe-thing

次のコード例は、describe-thing を使用する方法を示しています。

AWS CLI

モノに関する詳細情報を表示するには

次のdescribe-thing例では、AWS アカウントの AWS IoT レジストリで定義されているモノ (デバイス) に関する情報を表示します。

```
aws iot describe-thing --thing-nameMyLightBulb 「」
```

出力:

```
{
  "defaultClientId": "MyLightBulb",
  "thingName": "MyLightBulb",
  "thingId": "40da2e73-c6af-406e-b415-15acae538797",
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
  "thingTypeName": "LightBulb",
  "attributes": {
    "model": "123",
    "wattage": "75"
  },
  "version": 1
}
```

詳細については、「AWS IoT デベロッパーガイド」の「[レジストリによるモノの管理方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeThing](#)」の「」を参照してください。

detach-policy

次のコード例は、detach-policy を使用する方法を示しています。

AWS CLI

例 1: IoT AWS IoT ポリシーをモノのグループからデタッチするには

次のdetach-policy例では、指定されたポリシーをモノのグループからデタッチし、さらにそのグループ内のすべてのモノとグループの子グループからデタッチします。

```
aws iot detach-policy \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --policy-name "MyFirstGroup_Core-policy"
```

このコマンドでは何も出力されません。

詳細については、[「IoT デベロッパーガイド」の「モノのグループAWS IoT」](#)を参照してください。

例 2: デバイス証明書から AWS IoT ポリシーをデタッチするには

次のdetach-policy例では、 によって識別されたデバイス証明書から TemperatureSensorPolicy ポリシーをデタッチしますARN。

```
aws iot detach-policy \  
  --policy-name TemperatureSensorPolicy \  
  --target arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachPolicy](#)」の「」を参照してください。

detach-security-profile

次のコード例は、detach-security-profile を使用する方法を示しています。

AWS CLI

ターゲットからセキュリティプロファイルの関連付けを解除するには

次のdetach-security-profile例では、 という名前の AWS IoT Device Defender セキュリティプロファイルTestprofileと、登録されたすべてのモノのターゲットとの関連付けを削除します。

```
aws iot detach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/registered-things"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachSecurityProfile](#)」の「」を参照してください。

detach-thing-principal

次のコード例は、detach-thing-principal を使用する方法を示しています。

AWS CLI

モノから証明書/プリンシパルをデタッチするには

次のdetach-thing-principal例では、指定されたモノからプリンシパルを表す証明書を削除します。

```
aws iot detach-thing-principal \  
  --thing-name "MyLightBulb" \  
  --principal "arn:aws:iot:us-  
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36"
```

このコマンドでは何も出力されません。

詳細については、「AWS IoT デベロッパーガイド」の「[レジストリによるモノの管理方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachThingPrincipal](#)」の「」を参照してください。

disable-topic-rule

次のコード例は、disable-topic-rule を使用する方法を示しています。

AWS CLI

トピックルールを無効にするには

次のdisable-topic-rule例では、指定されたトピックルールを無効にします。

```
aws iot disable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[ルールの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableTopicRule](#)」の「」を参照してください。

enable-topic-rule

次の例は、enable-topic-rule を使用する方法を説明しています。

AWS CLI

トピックルールを有効にするには

次のenable-topic-rule例では、指定されたトピックルールを有効化 (または再有効化) します。

```
aws iot enable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[ルールの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableTopicRule](#)」の「」を参照してください。

get-behavior-model-training-summaries

次の例は、get-behavior-model-training-summaries を使用する方法を説明しています。

AWS CLI

Device Defender の ML Detect Security Profile トレーニングモデルのステータスを一覧表示するには

次のget-behavior-model-training-summaries例では、選択したセキュリティプロファイルで設定された動作のモデルトレーニングステータスを一覧表示します。動作ごとに、名前、モデルステータス、収集されたデータポイントの割合が一覧表示されます。

```
aws iot get-behavior-model-training-summaries \  
  --security-profile-name MySecuirtyProfileName
```

出力:

```
{  
  "summaries": [  
    {  
      "name": "MySecurityProfile",  
      "status": "Training",  
      "percentage": 100  
    }  
  ]  
}
```

```
{
  "securityProfileName": "MySecuirtyProfileName",
  "behaviorName": "Messages_sent_ML_behavior",
  "modelStatus": "PENDING_BUILD",
  "datapointsCollectionPercentage": 0.0
},
{
  "securityProfileName": "MySecuirtyProfileName",
  "behaviorName": "Messages_received_ML_behavior",
  "modelStatus": "PENDING_BUILD",
  "datapointsCollectionPercentage": 0.0
},
{
  "securityProfileName": "MySecuirtyProfileName",
  "behaviorName": "Authorization_failures_ML_behavior",
  "modelStatus": "PENDING_BUILD",
  "datapointsCollectionPercentage": 0.0
},
{
  "securityProfileName": "MySecuirtyProfileName",
  "behaviorName": "Message_size_ML_behavior",
  "modelStatus": "PENDING_BUILD",
  "datapointsCollectionPercentage": 0.0
},
{
  "securityProfileName": "MySecuirtyProfileName",
  "behaviorName": "Connection_attempts_ML_behavior",
  "modelStatus": "PENDING_BUILD",
  "datapointsCollectionPercentage": 0.0
},
{
  "securityProfileName": "MySPNoALerts",
  "behaviorName": "Disconnects_ML_behavior",
  "modelStatus": "PENDING_BUILD",
  "datapointsCollectionPercentage": 0.0
}
]
}
```

詳細については、AWS IoT デベロッパガイドの[GetBehaviorModelTrainingSummaries](#) 「(コマンドの検出)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBehaviorModelTrainingSummaries](#)」の「」を参照してください。

get-cardinality

次のコード例は、get-cardinality を使用する方法を示しています。

AWS CLI

クエリに一致する一意の値の概算数を返すには

次のセットアップスクリプトを使用して、10 個の温度センサーを表す 10 個のモノを作成できます。新しいモノには 3 つの属性があります。

```
# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
Racks=(Rack1 Rack1 Rack2 Rack2 Rack3 Rack4 Rack5 Rack6 Rack6 Rack6)
IsNormal=(true true true true true true false false false false)
for ((i=0; i<10 ; i++))
do
  thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
  attributes="{temperature=${Temperatures[i]},rackId=${Racks[i]},stateNormal=
  ${IsNormal[i]}}")
  aws iot describe-thing --thing-name "TempSensor$i"
done
```

セットアップスクリプトの出力例 :

```
{
  "version": 1,
  "thingName": "TempSensor0",
  "defaultClientId": "TempSensor0",
  "attributes": {
    "rackId": "Rack1",
    "stateNormal": "true",
    "temperature": "70"
  },
  "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
  "thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

次のget-cardinality例では、セットアップスクリプトによって作成された 10 個のセンサーをクエリし、異常な温度値をレポートする温度センサーを持つラックの数を返します。温度値が 60 未満または 80 を超える場合、温度センサーは異常な状態になります。

```
aws iot get-cardinality \  
  --aggregation-field "attributes.rackId" \  
  --query-string "thingName:TempSensor* AND attributes.stateNormal:false"
```

出力:

```
{  
  "cardinality": 2  
}
```

詳細については、AWS IoT デベロッパーガイドの「集計データのクエリ <<https://docs.aws.amazon.com/iot/latest/developerguide/index-aggregate.html>>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetCardinality](#)」の「」を参照してください。

get-effective-policies

次のコード例は、get-effective-policies を使用する方法を示しています。

AWS CLI

モノに影響を与えるポリシーを一覧表示するには

次のget-effective-policies例では、指定されたモノに影響を与えるポリシーを一覧表示します。これには、それが属するすべてのグループにアタッチされたポリシーが含まれます。

```
aws iot get-effective-policies \  
  --thing-name TemperatureSensor-001 \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

出力:

```
{  
  "effectivePolicies": [  
    {  
      "policyName": "TemperatureSensorPolicy",  
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/  
TemperatureSensorPolicy",  
      "policyDocument": "{
```

```

    \"Version\": \"2012-10-17\",
    \"Statement\": [
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Publish\",
          \"iot:Receive\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
          \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
        ]
      },
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Subscribe\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_1\",
          \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_2\"
        ]
      },
      {
        \"Effect\": \"Allow\",
        \"Action\": [
          \"iot:Connect\"
        ],
        \"Resource\": [
          \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub
\"
        ]
      }
    ]
  }
}

```

詳細については、「IoT デベロッパーガイド」の [「モノの効果的なポリシーを取得する」](#) を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEffectivePolicies](#)」の「」を参照してください。

get-indexing-configuration

次の例は、get-indexing-configuration を使用する方法を説明しています。

AWS CLI

モノのインデックス設定を取得するには

次のget-indexing-configuration例では、AWS IoT フリートインデックス作成の現在の設定データを取得します。

```
aws iot get-indexing-configuration
```

出力:

```
{
  "thingIndexingConfiguration": {
    "thingIndexingMode": "OFF",
    "thingConnectivityIndexingMode": "OFF"
  },
  "thingGroupIndexingConfiguration": {
    "thingGroupIndexingMode": "OFF"
  }
}
```

詳細については、IoT デベロッパーガイドの「[モノのインデックス作成の管理](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIndexingConfiguration](#)」の「」を参照してください。

get-job-document

次の例は、get-job-document を使用する方法を説明しています。

AWS CLI

ジョブのドキュメントを取得するには

次のget-job-document例では、ID が のジョブのドキュメントの詳細を表示しますexample-job-01。

```
aws iot get-job-document \  
  --job-id "example-job-01"
```

出力:

```
{  
  "document": "\n{\n  \"operation\": \"customJob\", \n  \"otherInfo\":  
  \"someValue\"\n}\n"}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetJobDocument](#)」の「」を参照してください。

get-logging-options

次のコード例は、get-logging-options を使用する方法を示しています。

AWS CLI

ログ記録オプションを取得するには

次のget-logging-options例では、AWS アカウントの現在のログ記録オプションを取得します。

```
aws iot get-logging-options
```

出力:

```
{  
  "roleArn": "arn:aws:iam::123456789012:role/service-role/iotLoggingRole",  
  "logLevel": "ERROR"  
}
```

詳細については、AWS 「IoT デベロッパーガイド」の「タイトル」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLoggingOptions](#)」の「」を参照してください。

get-ota-update

次の例は、get-ota-update を使用する方法を説明しています。

AWS CLI

OTA 更新に関する情報を取得するには

次のget-ota-update例では、指定されたOTA更新の詳細を表示します。

```
aws iot get-ota-update \  
  --ota-update-id ota12345
```

出力:

```
{  
  "otaUpdateInfo": {  
    "otaUpdateId": "ota12345",  
    "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",  
    "creationDate": 1557863215.995,  
    "lastModifiedDate": 1557863215.995,  
    "description": "A critical update needed right away.",  
    "targets": [  
      "device1",  
      "device2",  
      "device3",  
      "device4"  
    ],  
    "targetSelection": "SNAPSHOT",  
    "protocols": ["HTTP"],  
    "awsJobExecutionsRolloutConfig": {  
      "maximumPerMinute": 10  
    },  
    "otaUpdateFiles": [  
      {  
        "fileName": "firmware.bin",  
        "fileLocation": {  
          "stream": {  
            "streamId": "004",
```



```

        "fileId":123
      }
    },
    "codeSigning": {
      "awsSignerJobId": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
    }
  }
],
"roleArn": "arn:aws:iam:123456789012:role/service-role/my_ota_role"
"otaUpdateStatus": "CREATE_COMPLETE",
"awsIotJobId": "job54321",
"awsIotJobArn": "arn:aws:iot:us-west-2:123456789012:job/job54321",
"errorInfo": {
}
}
}

```

詳細については、IoT リファレンスの「[GetOTAUpdate](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI「コマンドリファレンス[GetOtaUpdate](#)」の「」を参照してください。

get-percentiles

次の例は、get-percentiles を使用する方法を説明しています。

AWS CLI

クエリに一致する集計値をパーセンタイルグループにグループ化するには

次のセットアップスクリプトを使用して、10 個の温度センサーを表す 10 個のモノを作成できます。新しいモノには 1 つの属性があります。

```

# Bash script. If in other shells, type `bash` before running
Temperatures=(70 71 72 73 74 75 47 97 98 99)
for ((i=0; i<10 ; i++))
do
  thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
  attributes="{temperature=${Temperatures[i]}}")
  aws iot describe-thing --thing-name "TempSensor$i"
done

```

セットアップスクリプトの出力例：

```
{
  "version": 1,
  "thingName": "TempSensor0",
  "defaultClientId": "TempSensor0",
  "attributes": {
    "temperature": "70"
  },
  "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/TempSensor0",
  "thingId": "example1-90ab-cdef-fedc-ba987example"
}
```

次のget-percentiles例では、セットアップスクリプトによって作成された 10 個のセンサーをクエリし、指定された各パーセンタイルグループの値を返します。パーセンタイルグループ「10」には、クエリに一致する値の約 10% で発生する集計フィールド値が含まれます。次の出力では、{「パーセント」: 10.0、「値」: 67.7} は、温度値の約 10.0% が 67.7 未満であることを意味します。

```
aws iot get-percentiles \
  --aggregation-field "attributes.temperature" \
  --query-string "thingName:TempSensor*" \
  --percents 10 25 50 75 90
```

出力:

```
{
  "percentiles": [
    {
      "percent": 10.0,
      "value": 67.7
    },
    {
      "percent": 25.0,
      "value": 71.25
    },
    {
      "percent": 50.0,
      "value": 73.5
    },
    {
      "percent": 75.0,
      "value": 91.5
    }
  ]
}
```

```
    },
    {
      "percent": 90.0,
      "value": 98.1
    }
  ]
}
```

詳細については、AWS IoT デベロッパーガイドの「[集計データのクエリ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPercentiles](#)」の「」を参照してください。

get-policy-version

次のコード例は、get-policy-version を使用する方法を示しています。

AWS CLI

ポリシーの特定のバージョンに関する情報を取得するには

次のget-policy-version例では、指定されたポリシーの最初のバージョンに関する情報を取得します。

```
aws iot get-policy \
  --policy-name UpdateDeviceCertPolicy
  --policy-version-id "1"
```

出力:

```
{
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",
  "policyName": "UpdateDeviceCertPolicy",
  "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow\", \"Action\": \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",
  "policyVersionId": "1",
  "isDefaultVersion": false,
  "creationDate": 1559925941.924,
  "lastModifiedDate": 1559926175.458,
  "generationId":
  "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"
```

```
}
```

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT ポリシーAWS IoT」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicyVersion](#)」の「」を参照してください。

get-policy

次のコード例は、get-policy を使用する方法を示しています。

AWS CLI

ポリシーのデフォルトバージョンに関する情報を取得するには

次のget-policy例では、指定されたポリシーのデフォルトバージョンに関する情報を取得します。

```
aws iot get-policy \  
  --policy-name UpdateDeviceCertPolicy
```

出力:

```
{  
  "policyName": "UpdateDeviceCertPolicy",  
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy",  
  "policyDocument": "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\":  
  \"Allow\", \"Action\": \"iot:UpdateCertificate\", \"Resource\": \"*\" } ] }",  
  "defaultVersionId": "2",  
  "creationDate": 1559925941.924,  
  "lastModifiedDate": 1559925941.924,  
  "generationId":  
  "5066f1b6712ce9d2a1e56399771649a272d6a921762fead080e24fe52f24e042"  
}
```

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT ポリシーAWS IoT」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicy](#)」の「」を参照してください。

get-registration-code

次のコード例は、get-registration-code を使用する方法を示しています。

AWS CLI

AWS アカウント固有の登録コードを取得するには

次のget-registration-code例では、AWS アカウント固有の登録コードを取得します。

```
aws iot get-registration-code
```

出力:

```
{
  "registrationCode":
  "15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"
}
```

詳細については、AWS IoT デベロッパーガイドの「[独自の証明書を使用する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRegistrationCode](#)」の「」を参照してください。

get-statistics

次の例は、get-statistics を使用する方法を説明しています。

AWS CLI

デバイスインデックスで集計データを検索するには

次のget-statistics例では、というプロパティがデバイスシャドウで connectivity.connected に設定されているモノの数 false (つまり、接続されていないデバイスの数) を返します。

```
aws iot get-statistics \
  --index-name AWS_Things \
  --query-string "connectivity.connected:false"
```

出力:

```
{
  "statistics": {
    "count": 6
  }
}
```

詳細については、AWS IoT デベロッパーガイドの「[デバイスフリートに関する統計の取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStatistics](#)」の「」を参照してください。

get-topic-rule-destination

次の例は、get-topic-rule-destination を使用する方法を説明しています。

AWS CLI

トピックルールの送信先を取得するには

次のget-topic-rule-destination例では、トピックルールの送信先に関する情報を取得します。

```
aws iot get-topic-rule-destination \
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

出力:

```
{
  "topicRuleDestination": {
    "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "status": "DISABLED",
    "httpUrlProperties": {
      "confirmationUrl": "https://example.com"
    }
  }
}
```

詳細については、AWS IoT デベロッパーガイドの「[トピックルールの送信先の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTopicRuleDestination](#)」の「」を参照してください。

get-topic-rule

次のコード例は、get-topic-rule を使用する方法を示しています。

AWS CLI

ルールに関する情報を取得するには

次のget-topic-rule例では、指定されたルールに関する情報を取得します。

```
aws iot get-topic-rule \  
  --rule-name MyRPiLowMoistureAlertRule
```

出力:

```
{  
  "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/MyRPiLowMoistureAlertRule",  
  "rule": {  
    "ruleName": "MyRPiLowMoistureAlertRule",  
    "sql": "SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE  
state.reported.moisture = 'low'\n          ",  
    "description": "Sends an alert whenever soil moisture level readings are too  
low.",  
    "createdAt": 1558624363.0,  
    "actions": [  
      {  
        "sns": {  
          "targetArn": "arn:aws:sns:us-  
west-2:123456789012:MyRPiLowMoistureTopic",  
          "roleArn": "arn:aws:iam::123456789012:role/service-role/  
MyRPiLowMoistureTopicRole",  
          "messageFormat": "RAW"  
        }  
      }  
    ],  
    "ruleDisabled": false,  
    "awsIotSqlVersion": "2016-03-23"  
  }  
}
```

```
}  
}
```

詳細については、[「IoT デベロッパーガイド」の「ルールを表示AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTopicRule](#)」の「」を参照してください。

get-v2-logging-options

次の例は、get-v2-logging-options を使用する方法を説明しています。

AWS CLI

現在のログ記録オプションを一覧表示するには

次のget-v2-logging-options例では、AWS IoT の現在のログ記録オプションを一覧表示します。

```
aws iot get-v2-logging-options
```

出力:

```
{  
  "roleArn": "arn:aws:iam::094249569039:role/service-role/iotLoggingRole",  
  "defaultLogLevel": "WARN",  
  "disableAllLogs": false  
}
```

詳細については、AWS 「IoT デベロッパーガイド」の「タイトル」を参照してください。

- API 詳細については、「コマンドリファレンス」の[GetV2LoggingOptions](#)を参照してください。AWS CLI

list-active-violations

次の例は、list-active-violations を使用する方法を説明しています。

AWS CLI

アクティブな違反を一覧表示するには

次のlist-active-violations例では、指定されたセキュリティプロファイルに対するすべての違反を一覧表示します。

```
aws iot list-active-violations \  
--security-profile-name Testprofile
```

出力:

```
{  
  "activeViolations": [  
    {  
      "violationId": "174db59167fa474c80a652ad1583fd44",  
      "thingName": "iotconsole-1560269126751-1",  
      "securityProfileName": "Testprofile",  
      "behavior": {  
        "name": "Authorization",  
        "metric": "aws:num-authorization-failures",  
        "criteria": {  
          "comparisonOperator": "greater-than",  
          "value": {  
            "count": 10  
          }  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1  
      }  
    },  
    "lastViolationValue": {  
      "count": 0  
    },  
    "lastViolationTime": 1560293700.0,  
    "violationStartTime": 1560279000.0  
  },  
  {  
    "violationId": "c8a9466a093d3b7b35cd44ca58bdbbeab",  
    "thingName": "TvnQoEoU",  
    "securityProfileName": "Testprofile",  
    "behavior": {  
      "name": "CellularBandwidth",  
      "metric": "aws:message-byte-size",  
      "criteria": {  
        "comparisonOperator": "greater-than",  
        "value": {
```

```
        "count": 128
      },
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1
    }
  },
  "lastViolationValue": {
    "count": 110
  },
  "lastViolationTime": 1560369000.0,
  "violationStartTime": 1560276600.0
},
{
  "violationId": "74aa393adea02e6648f3ac362beed55e",
  "thingName": "iotconsole-1560269232412-2",
  "securityProfileName": "Testprofile",
  "behavior": {
    "name": "Authorization",
    "metric": "aws:num-authorization-failures",
    "criteria": {
      "comparisonOperator": "greater-than",
      "value": {
        "count": 10
      }
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  }
},
  "lastViolationValue": {
    "count": 0
  },
  "lastViolationTime": 1560276600.0,
  "violationStartTime": 1560276600.0
},
{
  "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
  "thingName": "TvnQoEoU",
  "securityProfileName": "Testprofile",
  "behavior": {
    "name": "Authorization",
    "metric": "aws:num-authorization-failures",
    "criteria": {
      "comparisonOperator": "greater-than",
```

```

        "value": {
            "count": 10
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    }
},
"lastViolationValue": {
    "count": 0
},
"lastViolationTime": 1560369000.0,
"violationStartTime": 1560276600.0
}
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListActiveViolations](#)」の「」を参照してください。

list-attached-policies

次のコード例は、list-attached-policies を使用する方法を示しています。

AWS CLI

例 1: グループにアタッチされたポリシーを一覧表示するには

次のlist-attached-policies例では、指定されたグループにアタッチされているポリシーを一覧表示します。

```

aws iot list-attached-policies \
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"

```

出力:

```

{
  "policies": [
    {
      "policyName": "UpdateDeviceCertPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy"
    }
  ]
}

```

```
    }  
  ]  
}
```

詳細については、IoT デベロッパーガイドの「[モノのグループ](#)」を参照してください。AWS IoT

例 2: デバイス証明書にアタッチされたポリシーを一覧表示するには

次のlist-attached-policies例では、デバイス証明書にアタッチされた AWS IoT ポリシーを一覧表示します。証明書は `arn` によって識別されます。

```
aws iot list-attached-policies \  
  --target arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

出力:

```
{  
  "policies": [  
    {  
      "policyName": "TemperatureSensorPolicy",  
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/  
TemperatureSensorPolicy"  
    }  
  ]  
}
```

詳細については、「IoT デベロッパーガイド」の「[モノのグループ AWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAttachedPolicies](#)」の「」を参照してください。

list-audit-findings

次のコード例は、list-audit-findings を使用する方法を示しています。

AWS CLI

例 1: 監査のすべての結果を一覧表示するには

次のlist-audit-findings例では、指定されたタスク ID を持つ AWS IoT Device Defender 監査のすべての検出結果を一覧表示します。

```
aws iot list-audit-findings \  
  --task-id a3aea009955e501a31b764abe1bebd3d
```

出力:

```
{  
  "findings": []  
}
```

例 2: 監査チェックタイプの結果を一覧表示するには

次のlist-audit-findings例は、デバイスがデバイス証明書を共有している 2019 年 6 月 5 日から 2019 年 6 月 19 日の間に実行された AWS IoT Device Defender 監査の結果を示しています。チェック名を指定するときは、開始時刻と終了時刻を指定する必要があります。

```
aws iot list-audit-findings \  
  --check-name DEVICE_CERTIFICATE_SHARED_CHECK \  
  --start-time 1559747125 \  
  --end-time 1560962028
```

出力:

```
{  
  "findings": [  
    {  
      "taskId": "eeef61068b0eb03c456d746c5a26ee04",  
      "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",  
      "taskStartTime": 1560161017.172,  
      "findingTime": 1560161017.592,  
      "severity": "CRITICAL",  
      "nonCompliantResource": {  
        "resourceType": "DEVICE_CERTIFICATE",  
        "resourceIdentifier": {  
          "deviceCertificateId":  
            "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"  
        }  
      },  
      "relatedResources": [  
        {
```

```
        "resourceType": "CLIENT_ID",
        "resourceIdentifier": {
            "clientId": "ZipxgAIl"
        },
        "additionalInfo": {
            "CONNECTION_TIME": "1560086374068"
        }
    },
    {
        "resourceType": "CLIENT_ID",
        "resourceIdentifier": {
            "clientId": "ZipxgAIl"
        },
        "additionalInfo": {
            "CONNECTION_TIME": "1560081552187",
            "DISCONNECTION_TIME": "1560086371552"
        }
    },
    {
        "resourceType": "CLIENT_ID",
        "resourceIdentifier": {
            "clientId": "ZipxgAIl"
        },
        "additionalInfo": {
            "CONNECTION_TIME": "1559289863631",
            "DISCONNECTION_TIME": "1560081532716"
        }
    }
],
"reasonForNonCompliance": "Certificate shared by one or more devices.",
"reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
},
{
    "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
    "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
    "taskStartTime": 1559988217.27,
    "findingTime": 1559988217.655,
    "severity": "CRITICAL",
    "nonCompliantResource": {
        "resourceType": "DEVICE_CERTIFICATE",
        "resourceIdentifier": {
            "deviceCertificateId":
                "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
        }
    }
}
```

```

    },
    "relatedResources": [
      {
        "resourceType": "CLIENT_ID",
        "resourceIdentifier": {
          "clientId": "xShGENLW"
        },
        "additionalInfo": {
          "CONNECTION_TIME": "1559972350825"
        }
      },
      {
        "resourceType": "CLIENT_ID",
        "resourceIdentifier": {
          "clientId": "xShGENLW"
        },
        "additionalInfo": {
          "CONNECTION_TIME": "1559255062002",
          "DISCONNECTION_TIME": "1559972350616"
        }
      }
    ],
    "reasonForNonCompliance": "Certificate shared by one or more devices.",
    "reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
  },
  {
    "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
    "checkName": "DEVICE_CERTIFICATE_SHARED_CHECK",
    "taskStartTime": 1559901817.31,
    "findingTime": 1559901817.767,
    "severity": "CRITICAL",
    "nonCompliantResource": {
      "resourceType": "DEVICE_CERTIFICATE",
      "resourceIdentifier": {
        "deviceCertificateId":
        "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b"
      }
    },
    "relatedResources": [
      {
        "resourceType": "CLIENT_ID",
        "resourceIdentifier": {
          "clientId": "TvnQoEoU"
        },
      },
    ]
  }

```

```

        "additionalInfo": {
            "CONNECTION_TIME": "1559826729768"
        }
    },
    {
        "resourceType": "CLIENT_ID",
        "resourceIdentifier": {
            "clientId": "TvnQoEoU"
        },
        "additionalInfo": {
            "CONNECTION_TIME": "1559345920964",
            "DISCONNECTION_TIME": "1559826728402"
        }
    }
],
"reasonForNonCompliance": "Certificate shared by one or more devices.",
"reasonForNonComplianceCode": "CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES"
}
]
}

```

詳細については、AWS IoT デベロッパーガイドの「[Audit Commands](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAuditFindings](#)」の「」を参照してください。

list-audit-mitigation-actions-executions

次の例は、list-audit-mitigation-actions-executions を使用する方法を説明しています。

AWS CLI

監査緩和アクションの実行の詳細を一覧表示するには

監査緩和アクションタスクは、AWS IoT Device Defender 監査からの 1 つ以上の検出結果に緩和アクションを適用します。次のlist-audit-mitigation-actions-executions例では、taskIdと が指定された結果の緩和アクションタスクの詳細を一覧表示します。

```

aws iot list-audit-mitigation-actions-executions \
  --task-id myActionsTaskId \
  --finding-id 0edbaec-2fe1-4cf5-abc9-d4c3e51f7464

```


出力:

```
{
  "actionsExecutions": [
    {
      "taskId": "myActionsTaskId",
      "findingId": "0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464",
      "actionName": "ResetPolicyVersionAction",
      "actionId": "1ea0b415-bef1-4a01-bd13-72fb63c59afb",
      "status": "COMPLETED",
      "startTime": "2019-12-10T15:19:13.279000-08:00",
      "endTime": "2019-12-10T15:19:13.337000-08:00"
    }
  ]
}
```

詳細については、AWS IoT デベロッパガイドの[ListAuditMitigationActionsExecutions](#)「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListAuditMitigationActionsExecutions](#)」の「」を参照してください。

list-audit-mitigation-actions-tasks

次のコード例は、list-audit-mitigation-actions-tasks を使用する方法を示しています。

AWS CLI

監査緩和アクションタスクを一覧表示するには

次のlist-audit-mitigation-actions-tasks例では、指定された期間内に結果に適用された緩和アクションを一覧表示します。

```
aws iot list-audit-mitigation-actions-tasks \
  --start-time 1594157400 \
  --end-time 1594157430
```

出力:

```
{
  "tasks": [
```

```
{
  "taskId": "0062f2d6-3999-488f-88c7-bef005414103",
  "startTime": "2020-07-07T14:30:15.172000-07:00",
  "taskStatus": "COMPLETED"
}
]
```

詳細については、AWS IoT デベロッパーガイドの[ListAuditMitigationActionsTasks](#)「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListAuditMitigationActionsTasks](#)」の「」を参照してください。

list-audit-suppressions

次の例は、list-audit-suppressions を使用する方法を説明しています。

AWS CLI

すべての監査結果の抑制を一覧表示するには

次のlist-audit-suppressions例では、アクティブな監査結果の抑制をすべて一覧表示します。

```
aws iot list-audit-suppressions
```

出力:

```
{
  "suppressions": [
    {
      "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
      "resourceIdentifier": {
        "deviceCertificateId": "c7691e<shortened>"
      },
      "expirationDate": 1597881600.0,
      "suppressIndefinitely": false
    }
  ]
}
```

詳細については、[「IoT デベロッパーガイド」の「監査結果の抑制AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAuditSuppressions](#)」の「」を参照してください。

list-audit-tasks

次のコード例は、list-audit-tasks を使用する方法を示しています。

AWS CLI

監査のすべての結果を一覧表示するには

次のlist-audit-tasks例では、2019年6月5日から2019年6月12日の間に実行された監査タスクを一覧表示します。

```
aws iot list-audit-tasks \  
  --start-time 1559747125 \  
  --end-time 1560357228
```

出力:

```
{  
  "tasks": [  
    {  
      "taskId": "a3aea009955e501a31b764abe1bebd3d",  
      "taskStatus": "COMPLETED",  
      "taskType": "ON_DEMAND_AUDIT_TASK"  
    },  
    {  
      "taskId": "f76b4b5102b632cd9ae38a279c266da1",  
      "taskStatus": "COMPLETED",  
      "taskType": "SCHEDULED_AUDIT_TASK"  
    },  
    {  
      "taskId": "51d9967d9f9ff4d26529505f6d2c444a",  
      "taskStatus": "COMPLETED",  
      "taskType": "SCHEDULED_AUDIT_TASK"  
    },  
    {  
      "taskId": "eef61068b0eb03c456d746c5a26ee04",
```

```
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "041c49557b7c7b04c079a49514b55589",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "82c7f2afac1562d18a4560be73998acc",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "ac9086b7222a2f5e2e17bb6fd30b3aeb",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  }
]
}
```

詳細については、AWS「IoT デベロッパーガイド」の「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスListAuditTasks](#)」の「」を参照してください。

list-authorizers

次のコード例は、list-authorizers を使用する方法を示しています。

AWS CLI

カスタムオーソライザーを一覧表示するには

次のlist-authorizers例では、AWS アカウントのカスタムオーソライザーを一覧表示します。

```
aws iot list-authorizers
```

出力:

```
{
  "authorizers": [
    {
      "authorizerName": "CustomAuthorizer",
      "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer"
    },
    {
      "authorizerName": "CustomAuthorizer2",
      "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer2"
    },
    {
      "authorizerName": "CustomAuthorizer3",
      "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer3"
    }
  ]
}
```

詳細については、AWS IoT APIリファレンスの[ListAuthorizers](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListAuthorizers](#)」の「」を参照してください。

list-billing-groups

次の例は、list-billing-groups を使用する方法を説明しています。

AWS CLI

AWS アカウントとリージョンの請求グループを一覧表示するには

次のlist-billing-groups例では、AWS アカウントと AWS リージョンに定義されているすべての請求グループを一覧表示します。

```
aws iot list-billing-groups
```

出力:

```
{
  "billingGroups": [
    {
      "groupName": "GroupOne",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne"
    }
  ]
}
```

詳細については、IoT デベロッパーガイドの「[請求グループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBillingGroups](#)」の「」を参照してください。

list-ca-certificates

次のコード例は、list-ca-certificates を使用する方法を示しています。

AWS CLI

AWS アカウントに登録されている CA 証明書を一覧表示するには

次のlist-ca-certificates例では、AWS アカウントに登録されている CA 証明書を一覧表示します。

```
aws iot list-ca-certificates
```

出力:

```
{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "certificateId":
      "f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "status": "INACTIVE",
      "creationDate": 1569365372.053
    }
  ]
}
```

```
    }
  ]
}
```

詳細については、AWS IoT デベロッパーガイドの「[独自の証明書を使用する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCaCertificates](#)」の「」を参照してください。

list-certificates-by-ca

次のコード例は、list-certificates-by-ca を使用する方法を示しています。

AWS CLI

CA 証明書で署名されたすべてのデバイス証明書を一覧表示するには

次のlist-certificates-by-ca例では、指定された CA 証明書で署名された AWS アカウント内のすべてのデバイス証明書を一覧表示します。

```
aws iot list-certificates-by-ca \
  --ca-certificate-
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

出力:

```
{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "certificateId":
"488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "status": "ACTIVE",
      "creationDate": 1569363250.557
    }
  ]
}
```

詳細については、IoT リファレンスの[ListCertificatesBy「CA」](#)を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCertificatesByCa](#)」の「」を参照してください。

list-certificates

次のコード例は、list-certificates を使用する方法を示しています。

AWS CLI

例 1: AWS アカウントに登録されている証明書を一覧表示するには

次のlist-certificates例では、アカウントに登録されているすべての証明書を一覧表示します。デフォルトのページング制限が 25 を超える場合は、このコマンドのnextMarkerレスポンス値を使用して次のコマンドに指定し、結果の次のバッチを取得できます。が値なしでnextMarker を返すまで繰り返します。

```
aws iot list-certificates
```

出力:

```
{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
      "certificateId":
      "604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36",
      "status": "ACTIVE",
      "creationDate": 1556810537.617
    },
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
      "certificateId":
      "262a1ac8a7d8aa72f6e96e365480f7313aa9db74b8339ec65d34dc3074e1c31e",
      "status": "ACTIVE",
      "creationDate": 1546447050.885
    },
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",

```



```
    "certificateId":
      "b193ab7162c0fadca83246d24fa090300a1236fe58137e121b011804d8ac1d6b",
      "status": "ACTIVE",
      "creationDate": 1546292258.322
    },
    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
      "certificateId":
        "7aebeea3845d14a44ec80b06b8b78a89f3f8a706974b8b34d18f5adf0741db42",
        "status": "ACTIVE",
        "creationDate": 1541457693.453
      },
      {
        "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
        "certificateId":
          "54458aa39ebb3eb39c91ffbbdcc3a6ca1c7c094d1644b889f735a6fc2cd9a7e3",
          "status": "ACTIVE",
          "creationDate": 1541113568.611
        },
        {
          "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
          "certificateId":
            "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
            "status": "ACTIVE",
            "creationDate": 1541022751.983
          }
        ]
      ]
    }
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCertificates](#)」の「」を参照してください。

list-custom-metrics

次の例は、list-custom-metrics を使用する方法を説明しています。

AWS CLI

カスタムメトリクスを一覧表示するには

次のlist-custom-metrics例では、すべてのカスタムメトリクスを一覧表示します。

```
aws iot list-custom-metrics \  
  --region us-east-1
```

出力:

```
{  
  "metricNames": [  
    "batteryPercentage"  
  ]  
}
```

詳細については、「IoT Core デベロッパーガイド」の「[カスタムメトリクス](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCustomMetrics](#)」の「」を参照してください。

list-dimensions

次の例は、list-dimensions を使用する方法を説明しています。

AWS CLI

AWS アカウントのディメンションを一覧表示するには

次のlist-dimensions例では、AWS アカウントで定義されているすべての AWS IoT Device Defender ディメンションを一覧表示します。

```
aws iot list-dimensions
```

出力:

```
{  
  "dimensionNames": [  
    "TopicFilterForAuthMessages",  
    "TopicFilterForActivityMessages"  
  ]  
}
```

詳細については、AWS IoT デベロッパガイドの「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDimensions](#)」の「」を参照してください。

list-domain-configurations

次のコード例は、list-domain-configurations を使用する方法を示しています。

AWS CLI

ドメイン設定を一覧表示するには

次のlist-domain-configurations例では、指定されたサービスタイプを持つ AWS アカウントのドメイン設定を一覧表示します。

```
aws iot list-domain-configurations \  
  --service-type "DATA"
```

出力:

```
{  
  "domainConfigurations":  
  [  
    {  
      "domainConfigurationName": "additionalDataDomain",  
      "domainConfigurationArn": "arn:aws:iot:us-  
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh",  
      "serviceType": "DATA"  
    },  
    {  
      "domainConfigurationName": "iot:Jobs",  
      "domainConfigurationArn": "arn:aws:iot:us-  
west-2:123456789012:domainconfiguration/iot:Jobs",  
      "serviceType": "JOBS"  
    },  
    {  
      "domainConfigurationName": "iot:Data-ATS",  
      "domainConfigurationArn": "arn:aws:iot:us-  
west-2:123456789012:domainconfiguration/iot:Data-ATS",  
      "serviceType": "DATA"  
    },  
  ],  
}
```

```
{
  "domainConfigurationName": "iot:CredentialProvider",
  "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/iot:CredentialProvider",
  "serviceType": "CREDENTIAL_PROVIDER"
}
]
```

詳細については、AWS IoT デベロッパーガイドの「[設定可能なエンドポイント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDomainConfigurations](#)」の「」を参照してください。

list-indices

次のコード例は、list-indices を使用する方法を示しています。

AWS CLI

設定された検索インデックスを一覧表示するには

次のlist-indices例では、AWS アカウントに設定されているすべての検索インデックスを一覧表示します。モノのインデックス作成を有効にしていない場合は、インデックスがない可能性があります。

```
aws iot list-indices
```

出力:

```
{
  "indexNames": [
    "AWS_Things"
  ]
}
```

詳細については、AWS IoT デベロッパーガイドの「[モノのインデックス作成の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListIndices](#)」の「」を参照してください。

list-job-executions-for-job

次の例は、list-job-executions-for-job を使用する方法を説明しています。

AWS CLI

AWS アカウントのジョブを一覧表示するには

次のlist-job-executions-for-job例では、で指定された AWS、アカウント内のジョブのすべてのジョブ実行を一覧表示します。jobId。

```
aws iot list-job-executions-for-job \  
  --job-id my-ota-job
```

出力:

```
{  
  "executionSummaries": [  
    {  
      "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/my_thing",  
      "jobExecutionSummary": {  
        "status": "QUEUED",  
        "queuedAt": "2022-03-07T15:58:42.195000-08:00",  
        "lastUpdatedAt": "2022-03-07T15:58:42.195000-08:00",  
        "executionNumber": 1,  
        "retryAttempt": 0  
      }  
    }  
  ]  
}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobExecutionsForJob](#)」の「」を参照してください。

list-job-executions-for-thing

次のコード例は、list-job-executions-for-thing を使用する方法を示しています。

AWS CLI

モノに対して実行されたジョブを一覧表示するには

次のlist-job-executions-for-thing例では、 という名前のモノに対して実行されたすべてのジョブを一覧表示しますMyRaspberryPi。

```
aws iot list-job-executions-for-thing \  
  --thing-name "MyRaspberryPi"
```

出力:

```
{  
  "executionSummaries": [  
    {  
      "jobId": "example-job-01",  
      "jobExecutionSummary": {  
        "status": "QUEUED",  
        "queuedAt": 1560787023.636,  
        "lastUpdatedAt": 1560787023.636,  
        "executionNumber": 1  
      }  
    }  
  ]  
}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobExecutionsForThing](#)」の「」を参照してください。

list-jobs

次の例は、list-jobs を使用する方法を説明しています。

AWS CLI

AWS アカウントのジョブを一覧表示するには

次のlist-jobs例では、AWS アカウント内のすべてのジョブをジョブステータスでソートして一覧表示します。

```
aws iot list-jobs
```

出力:

```
{
  "jobs": [
    {
      "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
      "jobId": "example-job-01",
      "targetSelection": "SNAPSHOT",
      "status": "IN_PROGRESS",
      "createdAt": 1560787022.733,
      "lastUpdatedAt": 1560787026.294
    }
  ]
}
```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobs](#)」の「」を参照してください。

list-mitigation-actions

次のコード例は、list-mitigation-actions を使用する方法を示しています。

AWS CLI

定義されたすべての緩和アクションを一覧表示するには

次のlist-mitigation-actions例では、AWS アカウントとリージョンに対して定義されたすべての緩和アクションを一覧表示します。アクションごとに、名前、ARN、および作成日が一覧表示されます。

```
aws iot list-mitigation-actions
```

出力:

```
{
  "actionIdentifiers": [
```

```
{
  "actionName": "DeactivateCACertAction",
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/DeactivateCACertAction",
  "creationDate": "2019-12-10T11:12:47.574000-08:00"
},
{
  "actionName": "ResetPolicyVersionAction",
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/ResetPolicyVersionAction",
  "creationDate": "2019-12-10T11:11:48.920000-08:00"
},
{
  "actionName": "PublishFindingToSNSAction",
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/PublishFindingToSNSAction",
  "creationDate": "2019-12-10T11:10:49.546000-08:00"
},
{
  "actionName": "AddThingsToQuarantineGroupAction",
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/AddThingsToQuarantineGroupAction",
  "creationDate": "2019-12-10T11:09:35.999000-08:00"
},
{
  "actionName": "UpdateDeviceCertAction",
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/UpdateDeviceCertAction",
  "creationDate": "2019-12-10T11:08:44.263000-08:00"
},
{
  "actionName": "SampleMitigationAction",
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/SampleMitigationAction",
  "creationDate": "2019-12-10T11:03:41.840000-08:00"
}
]
```

詳細については、AWS IoT デベロッパーガイドの[ListMitigationActions](#) 「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMitigationActions](#)」の「」を参照してください。

list-mitigations-actions

次の例は、list-mitigations-actions を使用する方法を説明しています。

AWS CLI

定義されたすべての緩和アクションを一覧表示するには

次のlist-mitigations-actions例では、AWS アカウントとリージョンに対して定義されたすべての緩和アクションを一覧表示します。アクションごとに、名前、ARN、および作成日が一覧表示されます。

```
aws iot list-mitigation-actions
```

出力:

```
{
  "actionIdentifiers": [
    {
      "actionName": "DeactivateCACertAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/DeactivateCACertAction",
      "creationDate": "2019-12-10T11:12:47.574000-08:00"
    },
    {
      "actionName": "ResetPolicyVersionAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/ResetPolicyVersionAction",
      "creationDate": "2019-12-10T11:11:48.920000-08:00"
    },
    {
      "actionName": "PublishFindingToSNSAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/PublishFindingToSNSAction",
      "creationDate": "2019-12-10T11:10:49.546000-08:00"
    },
    {
      "actionName": "AddThingsToQuarantineGroupAction",
      "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/AddThingsToQuarantineGroupAction",
      "creationDate": "2019-12-10T11:09:35.999000-08:00"
    },
    {
```

```
    "actionName": "UpdateDeviceCertAction",
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
UpdateDeviceCertAction",
    "creationDate": "2019-12-10T11:08:44.263000-08:00"
  },
  {
    "actionName": "SampleMitigationAction",
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
SampleMitigationAction",
    "creationDate": "2019-12-10T11:03:41.840000-08:00"
  }
]
}
```

詳細については、AWS IoT デベロッパーガイドの[ListMitigationActions](#) 「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMitigationsActions](#)」の「」を参照してください。

list-ota-updates

次のコード例は、list-ota-updates を使用する方法を示しています。

AWS CLI

アカウントのOTA更新を一覧表示するには

次のlist-ota-updates例では、利用可能なOTA更新を一覧表示します。

```
aws iot list-ota-updates
```

出力:

```
{
  "otaUpdates": [
    {
      "otaUpdateId": "itsaupdate",
      "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/
itsaupdate",
      "creationDate": 1557863215.995
    }
  ]
}
```

```
]
}
```

詳細については、IoT リファレンスの「[ListOTAUpdates](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOtaUpdates](#)」の「」を参照してください。

list-outgoing-certificates

次のコード例は、list-outgoing-certificates を使用する方法を示しています。

AWS CLI

別の AWS アカウントに転送される証明書を一覧表示するには

次のlist-outgoing-certificates例では、transfer-certificate コマンドを使用して別の AWS アカウントに転送されているすべてのデバイス証明書を一覧表示します。

```
aws iot list-outgoing-certificates
```

出力:

```
{
  "outgoingCertificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "certificateId":
"488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "transferredTo": "030714055129",
      "transferDate": 1569427780.441,
      "creationDate": 1569363250.557
    }
  ]
}
```

詳細については、AWS IoT APIリファレンスの[ListOutgoingCertificates](#) 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOutgoingCertificates](#)」の「」を参照してください。

list-policies

次の例は、list-policies を使用する方法を説明しています。

AWS CLI

AWS アカウントで定義されているポリシーを一覧表示するには

次のlist-policies例では、AWS アカウントで定義されているすべてのポリシーを一覧表示します。

```
aws iot list-policies
```

出力:

```
{
  "policies": [
    {
      "policyName": "UpdateDeviceCertPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/UpdateDeviceCertPolicy"
    },
    {
      "policyName": "PlantIoTPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/PlantIoTPolicy"
    },
    {
      "policyName": "MyPiGroup_Core-policy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/MyPiGroup_Core-policy"
    }
  ]
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT ポリシーAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicies](#)」の「」を参照してください。

list-policy-versions

次の例は、list-policy-versions を使用する方法を説明しています。

AWS CLI

例 1: ポリシーのすべてのバージョンを表示するには

次のlist-policy-versions例では、指定されたポリシーのすべてのバージョンとその作成日を一覧表示します。

```
aws iot list-policy-versions \  
  --policy-name LightBulbPolicy
```

出力:

```
{  
  "policyVersions": [  
    {  
      "versionId": "2",  
      "isDefaultVersion": true,  
      "createDate": 1559925941.924  
    },  
    {  
      "versionId": "1",  
      "isDefaultVersion": false,  
      "createDate": 1559925941.924  
    }  
  ]  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT ポリシーAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicyVersions](#)」の「」を参照してください。

list-principal-things

次の例は、list-principal-things を使用する方法を説明しています。

AWS CLI

プリンシパルにアタッチされたモノを一覧表示するには

次のlist-principal-things例では、`arn:aws:iot:us-west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8`によって指定されたプリンシパルにアタッチされたモノを一覧表示しますARN。

```
aws iot list-principal-things \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

出力:

```
{  
  "things": [  
    "DeskLamp",  
    "TableLamp"  
  ]  
}
```

詳細については、AWS IoT APIリファレンスの[ListPrincipalThings](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListPrincipalThings](#)」の「」を参照してください。

list-provisioning-template-versions

次の例は、list-provisioning-template-versions を使用する方法を説明しています。

AWS CLI

プロビジョニングテンプレートのバージョンを一覧表示するには

次のlist-provisioning-template-versions例では、指定されたプロビジョニングテンプレートの使用可能なバージョンを一覧表示します。

```
aws iot list-provisioning-template-versions \  
  --template-name "widget-template"
```

出力:

```
{
  "versions": [
    {
      "versionId": 1,
      "creationDate": 1574800471.339,
      "isDefaultVersion": true
    },
    {
      "versionId": 2,
      "creationDate": 1574801192.317,
      "isDefaultVersion": false
    }
  ]
}
```

詳細については、[AWS 「IoT Core デベロッパークガイド」の「IoT Secure Tunneling」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProvisioningTemplateVersions](#)」の「」を参照してください。

list-provisioning-templates

次のコード例は、list-provisioning-templates を使用する方法を示しています。

AWS CLI

プロビジョニングテンプレートを一覧表示するには

次のlist-provisioning-templates例では、AWS アカウント内のすべてのプロビジョニングテンプレートを一覧表示します。

```
aws iot list-provisioning-templates
```

出力:

```
{
  "templates": [
    {
      "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-template",

```

```
        "templateName": "widget-template",
        "description": "A provisioning template for widgets",
        "creationDate": 1574800471.367,
        "lastModifiedDate": 1574801192.324,
        "enabled": false
    }
]
```

詳細については、[AWS 「IoT Core デベロッパーガイド」の「IoT Secure Tunneling」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProvisioningTemplates](#)」の「」を参照してください。

list-role-aliases

次の例は、list-role-aliases を使用する方法を説明しています。

AWS CLI

AWS アカウントの AWS IoT ロールエイリアスを一覧表示するには

次のlist-role-aliases例では、AWS アカウントの AWS IoT ロールエイリアスを一覧表示します。

```
aws iot list-role-aliases
```

出力:

```
{
  "roleAliases": [
    "ResidentAlias",
    "ElectricianAlias"
  ]
}
```

詳細については、AWS IoT APIリファレンスの[ListRoleAliases](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRoleAliases](#)」の「」を参照してください。

list-scheduled-audits

次のコード例は、list-scheduled-audits を使用する方法を示しています。

AWS CLI

AWS アカウントのスケジュールされた監査を一覧表示するには

次のlist-scheduled-audits例では、AWS アカウントにスケジュールされた監査を一覧表示します。

```
aws iot list-scheduled-audits
```

出力:

```
{
  "scheduledAudits": [
    {
      "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
      "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/AWSIoTDeviceDefenderDailyAudit",
      "frequency": "DAILY"
    },
    {
      "scheduledAuditName": "AWSDeviceDefenderWeeklyAudit",
      "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/AWSDeviceDefenderWeeklyAudit",
      "frequency": "WEEKLY",
      "dayOfWeek": "SUN"
    }
  ]
}
```

詳細については、AWS IoT デベロッパーガイド」の「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListScheduledAudits](#)」の「」を参照してください。

list-security-profiles-for-target

次の例は、list-security-profiles-for-target を使用する方法を説明しています。

AWS CLI

ターゲットにアタッチされたセキュリティプロファイルを一覧表示するには

次のlist-security-profiles-for-target例では、未登録のデバイスにアタッチされている AWS IoT Device Defender セキュリティプロファイルを一覧表示します。

```
aws iot list-security-profiles-for-target \
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
  unregistered-things"
```

出力:

```
{
  "securityProfileTargetMappings": [
    {
      "securityProfileIdentifier": {
        "name": "Testprofile",
        "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
Testprofile"
      },
      "target": {
        "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
      }
    }
  ]
}
```

詳細については、AWS IoT デベロッパガイドの「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSecurityProfilesForTarget](#)」の「」を参照してください。

list-security-profiles

次の例は、list-security-profiles を使用する方法を説明しています。

AWS CLI

AWS アカウントのセキュリティプロファイルを一覧表示するには

次のlist-security-profiles例では、AWS アカウントで定義されているすべての AWS IoT Device Defender セキュリティプロファイルを一覧表示します。

```
aws iot list-security-profiles
```

出力:

```
{
  "securityProfileIdentifiers": [
    {
      "name": "Testprofile",
      "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Testprofile"
    }
  ]
}
```

詳細については、AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSecurityProfiles](#)」の「」を参照してください。

list-streams

次の例は、list-streams を使用する方法を説明しています。

AWS CLI

アカウントのストリームを一覧表示するには

次のlist-streams例では、AWS アカウント内のすべてのストリームを一覧表示します。

```
aws iot list-streams
```

出力:

```
{
  "streams": [
    {
      "streamId": "stream12345",
      "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
      "streamVersion": 1,
      "description": "This stream is used for Amazon FreeRTOS OTA Update
12345."
    },
  ],
}
```

```
{
  "streamId": "stream54321",
  "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream54321",
  "streamVersion": 1,
  "description": "This stream is used for Amazon FreeRTOS OTA Update
54321."
}
```

詳細については、AWS IoT APIリファレンスの[ListStreams](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListStreams](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースに関連付けられたタグとその値を表示するには

次のlist-tags-for-resource例では、モノのグループに関連付けられたタグと値を表示しますLightBulbs。

```
aws iot list-tags-for-resource \
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

出力:

```
{
  "tags": [
    {
      "Key": "Assembly",
      "Value": "Fact1NW"
    },
    {
      "Key": "MyTag",
      "Value": "777"
    }
  ]
}
```

```
}
```

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT リソースのタグ付け」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-targets-for-policy

次の例は、list-targets-for-policy を使用する方法を説明しています。

AWS CLI

AWS IoT ポリシーに関連付けられているプリンシパルを一覧表示するには

次のlist-targets-for-policy例では、指定されたポリシーがアタッチされているデバイス証明書を一覧表示します。

```
aws iot list-targets-for-policy \  
  --policy-name UpdateDeviceCertPolicy
```

出力:

```
{  
  "targets": [  
    "arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",  
    "arn:aws:iot:us-west-2:123456789012:cert/  
d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be"  
  ]  
}
```

詳細については、IoT デベロッパーガイドの「[モノのグループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTargetsForPolicy](#)」の「」を参照してください。

list-targets-for-security-profile

次の例は、list-targets-for-security-profile を使用する方法を説明しています。

AWS CLI

セキュリティプロファイルが適用されるターゲットを一覧表示するには

次のlist-targets-for-security-profile例では、 という名前の AWS IoT Device Defender セキュリティプロファイルPossibleIssueが適用されるターゲットを一覧表示します。

```
aws iot list-targets-for-security-profile \  
  --security-profile-name Testprofile
```

出力:

```
{  
  "securityProfileTargets": [  
    {  
      "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"  
    },  
    {  
      "arn": "arn:aws:iot:us-west-2:123456789012:all/registered-things"  
    }  
  ]  
}
```

詳細については、AWS IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTargetsForSecurityProfile](#)」の「」を参照してください。

list-thing-groups-for-thing

次の例は、list-thing-groups-for-thing を使用する方法を説明しています。

AWS CLI

モノが属するグループを一覧表示するには

次のlist-thing-groups-for-thing例では、指定されたモノが属するグループを一覧表示します。

```
aws iot list-thing-groups-for-thing \  
  --thing-id thing-id
```

```
--thing-name MyLightBulb
```

出力:

```
{
  "thingGroups": [
    {
      "groupName": "DeadBulbs",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/DeadBulbs"
    },
    {
      "groupName": "LightBulbs",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
    }
  ]
}
```

詳細については、[「IoT デベロッパーガイド」の「モノのグループAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListThingGroupsForThing](#)」の「」を参照してください。

list-thing-groups

次の例は、list-thing-groups を使用する方法を説明しています。

AWS CLI

AWS アカウントで定義されているモノのグループを一覧表示するには

次のdescribe-thing-group例では、AWS アカウントで定義されているすべてのモノのグループを一覧表示します。

```
aws iot list-thing-groups
```

出力:

```
{
  "thingGroups": [
```

```
{
  "groupName": "HalogenBulbs",
  "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs"
},
{
  "groupName": "LightBulbs",
  "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
}
]
```

詳細については、IoT デベロッパーガイドの「[モノのグループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[ListThingGroups](#)」の「」を参照してください。

list-thing-principals

次のコード例は、list-thing-principals を使用する方法を示しています。

AWS CLI

モノに関連付けられたプリンシパルを一覧表示するには

次のlist-thing-principals例では、指定されたモノに関連付けられているプリンシパル (X.509 証明書、IAMユーザー、グループ、ロール、Amazon Cognito ID、またはフェデレーテッド ID) を一覧表示します。

```
aws iot list-thing-principals \
  --thing-name MyRaspberryPi
```

出力:

```
{
  "principals": [
    "arn:aws:iot:us-west-2:123456789012:cert/33475ac865079a5fffd5ecd44240640349293facc760642d7d8d5dbb6b4c86893"
  ]
}
```

詳細については、AWS IoT APIリファレンスの[ListThingPrincipals](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListThingPrincipals](#)」の「」を参照してください。

list-thing-types

次のコード例は、list-thing-types を使用する方法を示しています。

AWS CLI

定義されたモノのタイプを一覧表示するには

次のlist-thing-types例では、AWS アカウントで定義されているモノのタイプのリストを表示します。

```
aws iot list-thing-types
```

出力:

```
{
  "thingTypes": [
    {
      "thingTypeName": "LightBulb",
      "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/
LightBulb",
      "thingTypeProperties": {
        "thingTypeDescription": "light bulb type",
        "searchableAttributes": [
          "model",
          "wattage"
        ]
      },
      "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1559772562.498
      }
    }
  ]
}
```

詳細については、IoT デベロッパーガイドの「[モノのタイプ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListThingTypes](#)」の「」を参照してください。

list-things-in-billing-group

次の例は、list-things-in-billing-group を使用する方法を説明しています。

AWS CLI

請求グループ内のモノを一覧表示するには

次のlist-things-in-billing-group例では、指定された請求グループにあるモノを一覧表示します。

```
aws iot list-things-in-billing-group \  
  --billing-group-name GroupOne
```

出力:

```
{  
  "things": [  
    "MyOtherLightBulb",  
    "MyLightBulb"  
  ]  
}
```

詳細については、IoT デベロッパーガイドの「[請求グループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListThingsInBillingGroup](#)」の「」を参照してください。

list-things-in-thing-group

次のコード例は、list-things-in-thing-group を使用する方法を示しています。

AWS CLI

グループに属するモノを一覧表示するには

次のlist-things-in-thing-group例では、指定されたモノのグループに属するモノを一覧表示します。

```
aws iot list-things-in-thing-group \  
  --thing-group-name LightBulbs
```

出力:

```
{  
  "things": [  
    "MyLightBulb"  
  ]  
}
```

詳細については、IoT デベロッパーガイドの [「モノのグループ」](#) を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [ListThingsInThingGroup](#)」の「」を参照してください。

list-things

次のコード例は、list-things を使用する方法を示しています。

AWS CLI

例 1: レジストリ内のすべてのモノを一覧表示するには

次のlist-things例では、AWS アカウントの AWS IoT レジストリで定義されているモノ (デバイス) を一覧表示します。

```
aws iot list-things
```

出力:

```
{  
  "things": [  
    {  
      "thingName": "ThirdBulb",  
      "thingTypeName": "LightBulb",  
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/ThirdBulb",  
      "attributes": {  
        "model": "123",  
        "wattage": "75"  
      },  
      "version": 2  
    }  
  ]  
}
```

```
    },
    {
      "thingName": "MyOtherLightBulb",
      "thingTypeName": "LightBulb",
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
      "attributes": {
        "model": "123",
        "wattage": "75"
      },
      "version": 3
    },
    {
      "thingName": "MyLightBulb",
      "thingTypeName": "LightBulb",
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
      "attributes": {
        "model": "123",
        "wattage": "75"
      },
      "version": 1
    },
    {
      "thingName": "SampleIoTThing",
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/SampleIoTThing",
      "attributes": {},
      "version": 1
    }
  ]
}
```

例 2: 特定の属性を持つ定義済みのモノを一覧表示するには

次の `list-things` の例は、`wattage` という名前の属性を持つモノのリストを表示します。

```
aws iot list-things \
  --attribute-name wattage
```

出力:

```
{
  "things": [
    {
      "thingName": "MyLightBulb",
```

```
    "thingTypeName": "LightBulb",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
    "attributes": {
      "model": "123",
      "wattage": "75"
    },
    "version": 1
  },
  {
    "thingName": "MyOtherLightBulb",
    "thingTypeName": "LightBulb",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
    "attributes": {
      "model": "123",
      "wattage": "75"
    },
    "version": 3
  }
]
```

詳細については、「AWS IoT デイベロッパーガイド」の「[レジストリによるモノの管理方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListThings](#)」の「」を参照してください。

list-topic-rule-destinations

次の例は、list-topic-rule-destinations を使用する方法を説明しています。

AWS CLI

トピックルールの送信先を一覧表示するには

次のlist-topic-rule-destinations例では、現在の AWS リージョンで定義したすべてのトピックルールの送信先を一覧表示します。

```
aws iot list-topic-rule-destinations
```

出力:

```
{
```

```
    "destinationSummaries": [  
      {  
        "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
        "status": "ENABLED",  
        "httpUrlSummary": {  
          "confirmationUrl": "https://example.com"  
        }  
      }  
    ]  
  }  
}
```

詳細については、AWS IoT デベロッパーガイドの「[トピックルールの送信先の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTopicRuleDestinations](#)」の「」を参照してください。

list-topic-rules

次のコード例は、list-topic-rules を使用する方法を示しています。

AWS CLI

ルールを一覧表示するには

次のlist-topic-rules例では、定義したすべてのルールを一覧表示します。

```
aws iot list-topic-rules
```

出力:

```
{  
  "rules": [  
    {  
      "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/  
MyRPiLowMoistureAlertRule",  
      "ruleName": "MyRPiLowMoistureAlertRule",  
      "topicPattern": "$aws/things/MyRPi/shadow/update/accepted",  
      "createdAt": 1558624363.0,  
      "ruleDisabled": false  
    },  
    {
```

```
    "ruleArn": "arn:aws:iot:us-west-2:123456789012:rule/
MyPlantPiMoistureAlertRule",
    "ruleName": "MyPlantPiMoistureAlertRule",
    "topicPattern": "$aws/things/MyPlantPi/shadow/update/accepted",
    "createdAt": 1541458459.0,
    "ruleDisabled": false
  }
]
}
```

詳細については、[「IoT デベロッパーガイド」の「ルールを表示AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTopicRules](#)」の「」を参照してください。

list-v2-logging-levels

次の例は、list-v2-logging-levels を使用する方法を説明しています。

AWS CLI

ログ記録レベルを一覧表示するには

次のlist-v2-logging-levels例では、設定されたログ記録レベルを一覧表示します。ログ記録レベルが設定されていない場合、このコマンドを実行すると NotConfiguredExceptionが発生します。

```
aws iot list-v2-logging-levels
```

出力:

```
{
  "logTargetConfigurations": [
    {
      "logTarget": {
        "targetType": "DEFAULT"
      },
      "logLevel": "ERROR"
    }
  ]
}
```

- API 詳細については、「コマンドリファレンス」の[ListV2LoggingLevels](#)を参照してください。AWS CLI

list-violation-events

次のコード例は、list-violation-events を使用する方法を示しています。

AWS CLI

期間中のセキュリティプロファイル違反を一覧表示するには

次のlist-violation-events例では、現在の AWS アカウントと AWS リージョンのすべての AWS IoT Device Defender セキュリティプロファイルについて、2019 年 6 月 5 日から 2019 年 6 月 12 日の間に発生した違反を一覧表示します。

```
aws iot list-violation-events \  
  --start-time 1559747125 \  
  --end-time 1560351925
```

出力:

```
{  
  "violationEvents": [  
    {  
      "violationId": "174db59167fa474c80a652ad1583fd44",  
      "thingName": "iotconsole-1560269126751-1",  
      "securityProfileName": "Testprofile",  
      "behavior": {  
        "name": "Authorization",  
        "metric": "aws:num-authorization-failures",  
        "criteria": {  
          "comparisonOperator": "greater-than",  
          "value": {  
            "count": 10  
          },  
          "durationSeconds": 300,  
          "consecutiveDatapointsToAlarm": 1,  
          "consecutiveDatapointsToClear": 1  
        }  
      },  
      "metricValue": {  
        "count": 0  
      }  
    }  
  ]  
}
```



```
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560279000.0
  },
  {
    "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
    "thingName": "TvnQoEoU",
    "securityProfileName": "Testprofile",
    "behavior": {
      "name": "CellularBandwidth",
      "metric": "aws:message-byte-size",
      "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
          "count": 128
        },
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    },
    "metricValue": {
      "count": 110
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560276600.0
  },
  {
    "violationId": "74aa393adea02e6648f3ac362beed55e",
    "thingName": "iotconsole-1560269232412-2",
    "securityProfileName": "Testprofile",
    "behavior": {
      "name": "Authorization",
      "metric": "aws:num-authorization-failures",
      "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
          "count": 10
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    },
    "metricValue": {
```

```

        "count": 0
      },
      "violationEventType": "in-alarm",
      "violationEventTime": 1560276600.0
    },
    {
      "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
      "thingName": "TvnQoEoU",
      "securityProfileName": "Testprofile",
      "behavior": {
        "name": "Authorization",
        "metric": "aws:num-authorization-failures",
        "criteria": {
          "comparisonOperator": "greater-than",
          "value": {
            "count": 10
          }
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
      }
    },
    "metricValue": {
      "count": 0
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560276600.0
  }
]
}

```

詳細については、AWS IoT デベロッパーガイドの「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListViolationEvents](#)」の「」を参照してください。

register-ca-certificate

次のコード例は、register-ca-certificate を使用する方法を示しています。

AWS CLI

認証局 (CA) 証明書を登録するには

次のregister-ca-certificate例では、CA 証明書を登録します。コマンドは、CA 証明書と、CA 証明書に関連付けられたプライベートキーを所有していることを証明するキー検証証明書を提供します。

```
aws iot register-ca-certificate \  
  --ca-certificate file://rootCA.pem \  
  --verification-cert file://verificationCert.pem
```

出力:

```
{  
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/  
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",  
  "certificateId":  
  "f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467"  
}
```

詳細については、IoT リファレンスの「[RegisterCACertificate](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterCaCertificate](#)」の「」を参照してください。

register-certificate

次の例は、register-certificate を使用する方法を説明しています。

AWS CLI

自己署名デバイス証明書を登録するには

次のregister-certificate例では、rootCA.pemCA 証明書によって署名されたdeviceCert.pemデバイス証明書を登録します。CA 証明書は、自己署名デバイス証明書の登録に使用する前に登録する必要があります。自己署名証明書は、このコマンドに渡すのと同じCA 証明書で署名する必要があります。

```
aws iot register-certificate \  
  --certificate-pem file://deviceCert.pem \  
  --ca-certificate-pem file://rootCA.pem
```

出力:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
  "certificateId":
  "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
}
```

詳細については、AWS IoT APIリファレンスの[RegisterCertificate](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[RegisterCertificate](#)」の「」を参照してください。

register-thing

次の例は、register-thing を使用する方法を説明しています。

AWS CLI

モノを登録するには

次のregister-thing例では、プロビジョニングテンプレートを使用してモノを登録します。

```
aws iot register-thing \
  --template-body '{"Parameters":{"ThingName":
{"Type":"String"},"AWS::IoT::Certificate::Id":{"Type":"String"}},'Resources':
{"certificate":{"Properties":{"CertificateId":
{"Ref":"AWS::IoT::Certificate::Id"},"Status":"Active"},"Type":"AWS::IoT::Certificate"},"poli
{"Properties":{"PolicyName":"MyIotPolicy"},"Type":"AWS::IoT::Policy"},"thing":
{"OverrideSettings":
{"AttributePayload":"MERGE","ThingGroups":"DO_NOTHING","ThingTypeName":"REPLACE"},"Propertie
{"AttributePayload":{},"ThingGroups":[],"ThingName":
{"Ref":"ThingName"},"ThingTypeName":"VirtualThings"},"Type":"AWS::IoT::Thing"}}}' \
  --parameters '{"ThingName":"Register-thing-
trial-1","AWS::IoT::Certificate::Id":"799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e3
```

出力:

```
{
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAKGgAwIBAgIUYLk81I35cIppobpw
```

```

Hi0J2jNjboIwDQYJKoZIhvcNAQEL
\nBQAwTTFLEMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi
5jb20g\nSW5jLiBMPVNIYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIwMDcyMzE2NDUw
\n0VoXDTQ5MTIzMT
IzNTk10VowHjEcMBoGA1UEAwwTQVdTIEIvVjBBDZlXJ0aWZpY2F0\nZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBA071uADhdBajqTmqrMV5\nnmCFfBZQRMo1MdtVoZr2X+M4MzL
+RARrtUzH9a2SMAckeX8Keb1I0TKzORI
RDXnyE
\n6lV0wjgAsd0ku22rFhex4eG2ikha7pYYkvuToqA7L3TxItRvfKrxRI4ZfJoFPip4\nKqiuBJVNOGKTCQ
Hd1RN0rddwwu6kFJLeKDMEXAMPLEdUF0N+qfR9yKnZQkm
+g6Q2\nGXu7u0W3hn6n1RN8qVoka0uW12p53xM7oHVz
Gf+cxKBx1b0hGkp6yCfTskUBm3Sp\n9zLw35kiHXVm4EVpwgNlnk6XcIGIkW8a/iy4pzmVUgAANY1/uU/
zgCjymw
ZT5S30\nBV0CAwEAAANgMF4wHwYDVR0jBBgwFoAUGx0tCcU3q2n1WXAuUCv6hugXjKswHQYD
\nVR00BBYEF0VtvZ
9Aj2RYFNkX7Iu01XTRUdxgMAwGA1UdEwEB/wQCMAAwDgYDVR0P\nAQH/
BAQDAgeAMA0GCSqGSIb3DQEBcWUAA4IB
AQXCQcp0tubS5ft0sDMTcP/jNX
\nDHyaRxmjpSc2aCdm7WX591TKWyAdxGAvqaDVWqTo0oXI7tZ8w7aIN1Gi5
pXnifx\n3SBebMUoBbTktrC97yUaeL025mCFv8emDnTR/fe7PTsBKjW0g/rrfpwBxZLXDFwN
\nnqkQjy3EDfifj2
6j0xYIqqWMPogyn4sr0CKynS5wMJUQZ1HQ0nabVwnwK4Y0Mflp
\np9+4susFUR9aT3BT1AcIwqSpzh1Khh4Iz7ND
kRn4amsUT210jg/z0010w+BTHcVQ\nJly8XDU0CWSu04q6SnaBzHmlySIajxuRTP/AdfRouP10Xe
+q1bPOBcvVvF
8o\n-----END CERTIFICATE-----\n",
  "resourceArns": {
    "certificate": "arn:aws:iot:us-
west-2:571032923833:cert/799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e30404b9233c",
    "thing": "arn:aws:iot:us-west-2:571032923833:thing/Register-thing-trial-1"
  }
}

```

詳細については、[「IoT Core デベロッパーガイド」の「信頼できるユーザーによるプロビジョニングAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterThing](#)」の「」を参照してください。

reject-certificate-transfer

次のコード例は、reject-certificate-transfer を使用する方法を示しています。

AWS CLI

証明書の転送を拒否するには

次のreject-certificate-transfer例では、別の AWS アカウントからの指定されたデバイス証明書の転送を拒否します。

```
aws iot reject-certificate-transfer \  
  --certificate-  
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78add5e605d630e05c7fc8
```

このコマンドでは何も出力されません。

詳細については、[「IoT Core デベロッパーガイド」の「証明書を別のアカウントに転送するAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RejectCertificateTransfer](#)」の「」を参照してください。

remove-thing-from-billing-group

次のコード例は、remove-thing-from-billing-group を使用する方法を示しています。

AWS CLI

請求グループからモノを削除するには

次のremove-thing-from-billing-group例では、指定されたモノを請求グループから削除します。

```
aws iot remove-thing-from-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyOtherLightBulb
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの[「請求グループ」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveThingFromBillingGroup](#)」の「」を参照してください。

remove-thing-from-thing-group

次の例は、remove-thing-from-thing-group を使用する方法を説明しています。

AWS CLI

モノをモノグループから削除するには

次のremove-thing-from-thing-group例では、モノグループから指定されたモノを削除します。

```
aws iot remove-thing-from-thing-group \  
  --thing-name bulb7 \  
  --thing-group-name DeadBulbs
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「Thing Groups <<https://docs.aws.amazon.com/iot/latest/developerguide/thing-groups.html>>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveThingFromThingGroup](#)」の「」を参照してください。

replace-topic-rule

次の例は、replace-topic-rule を使用する方法を説明しています。

AWS CLI

トピックのルール定義を更新するには

次のreplace-topic-rule例では、土壌湿度レベルの測定値が低すぎるとSNSアラートを送信するように指定されたルールを更新します。

```
aws iot replace-topic-rule \  
  --rule-name MyRPiLowMoistureAlertRule \  
  --topic-rule-payload "{\"sql\": \"SELECT * FROM '$aws/things/MyRPi/shadow/  
update/accepted' WHERE state.reported.moisture = 'low'\", \"description\": \"Sends  
an alert when soil moisture level readings are too low.\", \"actions\": [{\"sns  
\": {\"targetArn\": \"arn:aws:sns:us-west-2:123456789012:MyRPiLowMoistureTopic\",  
\"roleArn\": \"arn:aws:iam::123456789012:role/service-role/MyRPiLowMoistureTopicRole  
\", \"messageFormat\": \"RAW\"}}], \"ruleDisabled\": false, \"awsIotSqlVersion\":  
\"2016-03-23\"}"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT ルールの作成」を参照してください。

AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ReplaceTopicRule](#)」の「」を参照してください。

search-index

次の例は、search-index を使用する方法を説明しています。

AWS CLI

モノのインデックスをクエリするには

次のsearch-index例では、タイプの を持つモノについてAWS_ThingsインデックスをクエリしますLightBulb。

```
aws iot search-index \  
  --index-name "AWS_Things" \  
  --query-string "thingTypeName:LightBulb"
```

出力:

```
{  
  "things": [  
    {  
      "thingName": "MyLightBulb",  
      "thingId": "40da2e73-c6af-406e-b415-15acae538797",  
      "thingTypeName": "LightBulb",  
      "thingGroupNames": [  
        "LightBulbs",  
        "DeadBulbs"  
      ],  
      "attributes": {  
        "model": "123",  
        "wattage": "75"  
      },  
      "connectivity": {  
        "connected": false  
      }  
    },  
  ],  
}
```



```
{
  "thingName": "ThirdBulb",
  "thingId": "615c8455-33d5-40e8-95fd-3ee8b24490af",
  "thingTypeName": "LightBulb",
  "attributes": {
    "model": "123",
    "wattage": "75"
  },
  "connectivity": {
    "connected": false
  }
},
{
  "thingName": "MyOtherLightBulb",
  "thingId": "6dae0d3f-40c1-476a-80c4-1ed24ba6aa11",
  "thingTypeName": "LightBulb",
  "attributes": {
    "model": "123",
    "wattage": "75"
  },
  "connectivity": {
    "connected": false
  }
}
]
```

詳細については、AWS IoT デベロッパーガイドの「[モノのインデックス作成の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchIndex](#)」の「」を参照してください。

set-default-authorizer

次のコード例は、set-default-authorizer を使用する方法を示しています。

AWS CLI

デフォルトのオーソライザーを設定するには

次のset-default-authorizer例では、 という名前のカスタムオーソライザーをデフォルトのオーソライザーCustomAuthorizerとして設定します。

```
aws iot set-default-authorizer \  
  --authorizer-name CustomAuthorizer
```

出力:

```
{  
  "authorizerName": "CustomAuthorizer",  
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/  
CustomAuthorizer"  
}
```

詳細については、AWS IoT APIリファレンスの[CreateDefaultAuthorizer](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[SetDefaultAuthorizer](#)」の「」を参照してください。

set-default-policy-version

次のコード例は、set-default-policy-version を使用する方法を示しています。

AWS CLI

ポリシーのデフォルトバージョンを設定するには

次のset-default-policy-version例では、という名前のポリシー2のデフォルトバージョンをに設定しますUpdateDeviceCertPolicy。

```
aws iot set-default-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI「コマンドリファレンス[SetDefaultPolicyVersion](#)」の「」を参照してください。

set-v2-logging-level

次の例は、set-v2-logging-level を使用する方法を説明しています。

AWS CLI

モノのグループのログ記録レベルを設定するには

次のset-v2-logging-level例では、指定したモノのグループの警告をログに記録するようにログ記録レベルを設定します。

```
aws iot set-v2-logging-level \  
  --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \  
  --log-level WARN
```

このコマンドでは何も出力されません。

- API 詳細については、「コマンドリファレンス」の[SetV2LoggingLevel](#)を参照してください。AWS CLI

set-v2-logging-options

次のコード例は、set-v2-logging-options を使用する方法を示しています。

AWS CLI

ログ記録オプションを設定するには

次のset-v2-logging-options例では、デフォルトのログ記録の詳細レベルを に設定ERRORしARN、ログ記録に使用する を指定します。

```
aws iot set-v2-logging-options \  
  --default-log-level ERROR \  
  --role-arn "arn:aws:iam::094249569039:role/service-role/iotLoggingRole"
```

このコマンドでは何も出力されません。

- API 詳細については、「コマンドリファレンス」の[SetV2LoggingOptions](#)を参照してください。AWS CLI

start-audit-mitigation-actions-task

次の例は、start-audit-mitigation-actions-task を使用する方法を説明しています。

AWS CLI

監査の結果に緩和アクションを適用するには

次のstart-audit-mitigation-actions-task例では、ResetPolicyVersionActionアクション (ポリシーをクリア) を指定された単一の検出結果に適用します。

```
aws iot start-audit-mitigation-actions-task \  
  --task-id "myActionsTaskId" \  
  --target "findingIds=[\"@edbaaec-2fe1-4cf5-abc9-d4c3e51f7464\"]" \  
  --audit-check-to-actions-mapping  
  "IOT_POLICY_OVERLY_PERMISSIVE_CHECK=[\"ResetPolicyVersionAction\"]" \  
  --client-request-token "adhadhahda"
```

出力:

```
{  
  "taskId": "myActionsTaskId"  
}
```

詳細については、AWS IoT デベロッパーガイドの[StartAuditMitigationActionsTask 「\(緩和アクションコマンド\)」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartAuditMitigationActionsTask](#)」の「」を参照してください。

start-on-demand-audit-task

次のコード例は、start-on-demand-audit-task を使用する方法を示しています。

AWS CLI

監査をすぐに開始するには

次のstart-on-demand-audit-task例では、AWS IoT Device Defender 監査を開始し、3つの証明書チェックを実行します。

```
aws iot start-on-demand-audit-task \  
  --target-check-  
  names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

出力:

```
{
  "taskId": "a3aea009955e501a31b764abe1bebd3d"
}
```

詳細については、AWS「IoT デベロッパーガイド」の「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスStartOnDemandAuditTask](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースのタグキーと値を指定するには

次のtag-resource例では、キーAssemblyと 値を持つ タグをモノのグループに適用Fact1NWしますLightBulbs。

```
aws iot tag-resource \
  --tags Key=Assembly,Value="Fact1NW" \
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT デベロッパーガイド」の「IoT リソースのタグ付け](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI「[コマンドリファレンスTagResource](#)」の「」を参照してください。

test-authorization

次の例は、test-authorization を使用する方法を説明しています。

AWS CLI

AWS IoT ポリシーをテストするには

次のtest-authorization例では、指定されたプリンシパルに関連付けられた AWS IoT ポリシーをテストします。

```
aws iot test-authorization \  
  --auth-infos actionType=CONNECT,resources=arn:aws:iot:us-  
east-1:123456789012:client/client1 \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/  
aab1068f7f43ac3e3cae4b3a8aa3f308d2a750e6350507962e32c1eb465d9775
```

出力:

```
{  
  "authResults": [  
    {  
      "authInfo": {  
        "actionType": "CONNECT",  
        "resources": [  
          "arn:aws:iot:us-east-1:123456789012:client/client1"  
        ]  
      },  
      "allowed": {  
        "policies": [  
          {  
            "policyName": "TestPolicyAllowed",  
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/  
TestPolicyAllowed"  
          }  
        ]  
      },  
      "denied": {  
        "implicitDeny": {  
          "policies": [  
            {  
              "policyName": "TestPolicyDenied",  
              "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/  
TestPolicyDenied"  
            }  
          ]  
        },  
        "explicitDeny": {  
          "policies": [  
            {  
              "policyName": "TestPolicyExplicitDenied",  
              "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/  
TestPolicyExplicitDenied"  
            }  
          ]  
        }  
      }  
    ]  
  }  
}
```

```

    ]
  }
},
"authDecision": "IMPLICIT_DENY",
"missingContextValues": []
}
]
}

```

詳細については、AWS IoT APIリファレンスの[TestAuthorization](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TestAuthorization](#)」の「」を参照してください。

test-invoke-authorizer

次のコード例は、test-invoke-authorizer を使用する方法を示しています。

AWS CLI

カスタムオーソライザーをテストするには

次のtest-invoke-authorizer例ではtestSします。

```

aws iot test-invoke-authorizer \
  --authorizer-name IoTAuthorizer \
  --token allow \
  --token-signature "mE0GvaHqy9nER/  
FdgtJX5LXYEJ3b3vE7t1gEszc0TKGgLKWXtnPkb2AbKn0AZ81GyoN5dVtWDWVmr25m7+  
+zjbYIMk2TBvyGXh0mvKFBPkdgyA43KL6SiZy0cTqLPMcQDsP7VX2rXr7CTowCxSNKphGXdQe0/  
I5dQ+J06KUaHwCmupt0/MejKtaNwiiA064j6wpr0AUwG5S1IYFuRd0X  
+wfo8pb0DubAIX1Ua705kuhRUcTx4SxUSHEYKmN4IDEvLB6FsIr0B2wvB7y4iPmcajxzG102ExvyCUNctCV9dY1RRGJj"

```

出力:

```

{
  "isAuthenticated": true,
  "principalId": "principalId",
  "policyDocuments": [
    {"Version": "2012-10-17", "Statement":
[{"Action": "iot:Publish", "Effect": "Allow", "Resource": "arn:aws:iot:us-
west-2:123456789012:topic/customauthtesting"}]}]
  },

```

```
"refreshAfterInSeconds": 600,  
"disconnectAfterInSeconds": 3600  
}
```

詳細については、AWS IoT APIリファレンスの[TestInvokeAuthorizer](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TestInvokeAuthorizer](#)」の「」を参照してください。

transfer-certificate

次のコード例は、transfer-certificate を使用する方法を示しています。

AWS CLI

デバイス証明書を別の AWS アカウントに転送するには

次のtransfer-certificate例では、デバイス証明書を別の AWS アカウントに転送します。証明書と AWS アカウントは ID で識別されます。

```
aws iot transfer-certificate \  
  --certificate-  
id 488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142 \  
  --target-aws-account 030714055129
```

出力:

```
{  
  "transferredCertificateArn": "arn:aws:iot:us-  
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"  
}
```

詳細については、「[IoT Core デベロッパーガイド](#)」の「[証明書を別のアカウントに転送AWS する IoT](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TransferCertificate](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグキーを削除するには

次の`untag-resource`例では、モノのグループ からタグMyTagとその値を削除しますLightBulbs。

```
command
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「IoT リソースのタグ付け」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-account-audit-configuration

次のコード例は、`update-account-audit-configuration` を使用する方法を示しています。

AWS CLI

例 1: 監査SNS通知の Amazon 通知を有効にするには

次の`update-account-audit-configuration`例では、AWS IoT Device Defender 監査SNS通知の Amazon 通知を有効にし、ターゲットとそのターゲットへの書き込みに使用されるロールを指定します。

```
aws iot update-account-audit-configuration \  
  --audit-notification-target-configurations "SNS={targetArn=\"arn:aws:sns:us-west-2:123456789012:ddaudits\",roleArn=\"arn:aws:iam::123456789012:role/service-role/AWSIoTDeviceDefenderAudit\",enabled=true}"
```

このコマンドでは何も出力されません。

例 2: 監査チェックを有効にするには

次の`update-account-audit-configuration`例では、という名前の AWS IoT Device Defender 監査チェックを有効にしますAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK。AWS アカウントの 1 つ

以上のスケジュールされた監査targetCheckNamesの の一部である場合、監査チェックを無効にすることはできません。

```
aws iot update-account-audit-configuration \  
  --audit-check-configurations \  
  "{\"AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK\":{\"enabled\":true}}"
```

このコマンドでは何も出力されません。

詳細については、AWS「IoT デベロッパーガイド」の「[監査コマンド](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスUpdateAccountAuditConfiguration](#)」の「」を参照してください。

update-audit-suppression

次のコード例は、update-audit-suppression を使用する方法を示しています。

AWS CLI

監査結果の抑制を更新するには

次のupdate-audit-suppression例では、監査結果抑制の有効期限を 2020-09-21 に更新します。

```
aws iot update-audit-suppression \  
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \  
  --resource-identifier deviceCertificateId=c7691e<shortened> \  
  --no-suppress-indefinitely \  
  --expiration-date 2020-09-21
```

このコマンドでは何も出力されません。

詳細については、「IoT デベロッパーガイド」の「[監査結果の抑制AWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスUpdateAuditSuppression](#)」の「」を参照してください。

update-authorizer

次のコード例は、update-authorizer を使用する方法を示しています。

AWS CLI

カスタムオーソライザーを更新するには

次のupdate-authorizer例では、の状態を CustomAuthorizer2から にしていません INACTIVE。

```
aws iot update-authorizer \  
  --authorizer-name CustomAuthorizer2 \  
  --status INACTIVE
```

出力:

```
{  
  "authorizerName": "CustomAuthorizer2",  
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/  
CustomAuthorizer2"  
}
```

詳細については、AWS IoT APIリファレンスの[UpdateAuthorizer](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateAuthorizer](#)」の「」を参照してください。

update-billing-group

次のコード例は、update-billing-group を使用する方法を示しています。

AWS CLI

請求グループに関する情報を更新するには

次のupdate-billing-group例では、指定された請求グループの説明を更新します。

```
aws iot update-billing-group \  
  --billing-group-name GroupOne \  
  --billing-group-properties "billingGroupDescription=\"Primary bulb billing group  
\""
```

出力:

```
{
```

```
"version": 2
}
```

詳細については、IoT デベロッパーガイドの「[請求グループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateBillingGroup](#)」の「」を参照してください。

update-ca-certificate

次のコード例は、update-ca-certificate を使用する方法を示しています。

AWS CLI

認証局 (CA) 証明書を更新するには

次のupdate-ca-certificate例では、指定された CA 証明書を ACTIVEステータスに設定します。

```
aws iot update-ca-certificate \  
  --certificate-  
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467 \  
  --new-status ACTIVE
```

このコマンドでは何も出力されません。

詳細については、IoT リファレンスの「[UpdateCACertificate](#)」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCaCertificate](#)」の「」を参照してください。

update-certificate

次の例は、update-certificate を使用する方法を説明しています。

AWS CLI

デバイス証明書を更新するには

次のupdate-certificate例では、指定されたデバイス証明書を INACTIVEステータスに設定します。

```
aws iot update-certificate \  
  --certificate-  
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \  
  --new-status INACTIVE
```

このコマンドでは何も出力されません。

詳細については、AWS IoT APIリファレンスの[UpdateCertificate](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateCertificate](#)」の「」を参照してください。

update-custom-metric

次のコード例は、update-custom-metric を使用する方法を示しています。

AWS CLI

カスタムメトリクスを更新するには

次のupdate-custom-metric例では、カスタムメトリクスを更新して新しい `display-name` を取得します。

```
aws iot update-custom-metric \  
  --metric-name batteryPercentage \  
  --display-name 'remaining battery percentage on device' \  
  --region us-east-1
```

出力:

```
{  
  "metricName": "batteryPercentage",  
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/  
batteryPercentage",  
  "metricType": "number",  
  "displayName": "remaining battery percentage on device",  
  "creationDate": "2020-11-17T23:01:35.110000-08:00",  
  "lastModifiedDate": "2020-11-17T23:02:12.879000-08:00"  
}
```

詳細については、「IoT Core デベロッパーガイド」の[「カスタムメトリクス」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCustomMetric](#)」の「」を参照してください。

update-dimension

次のコード例は、update-dimension を使用する方法を示しています。

AWS CLI

ディメンションを更新するには

次のupdate-dimension例では、ディメンションを更新します。

```
aws iot update-dimension \  
  --name TopicFilterForAuthMessages \  
  --string-values device/${iot:ClientId}/auth
```

出力:

```
{  
  "name": "TopicFilterForAuthMessages",  
  "lastModifiedDate": 1585866222.317,  
  "stringValues": [  
    "device/${iot:ClientId}/auth"  
  ],  
  "creationDate": 1585854500.474,  
  "type": "TOPIC_FILTER",  
  "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/  
TopicFilterForAuthMessages"  
}
```

詳細については、AWS 「IoT Core デベロッパーガイド」の「[ディメンションを使用したセキュリティプロファイルのメトリクスのスコープ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDimension](#)」の「」を参照してください。

update-domain-configuration

次の例は、update-domain-configuration を使用する方法を説明しています。

AWS CLI

ドメイン設定を更新するには

次のupdate-domain-configuration例では、指定されたドメイン設定を無効にします。

```
aws iot update-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --domain-configuration-status "DISABLED"
```

出力:

```
{  
  "domainConfigurationName": "additionalDataDomain",  
  "domainConfigurationArn": "arn:aws:iot:us-  
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"  
}
```

詳細については、AWS IoT デベロッパーガイドの「[設定可能なエンドポイント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDomainConfiguration](#)」の「」を参照してください。

update-dynamic-thing-group

次の例は、update-dynamic-thing-group を使用する方法を説明しています。

AWS CLI

動的モノのグループを更新するには

次のupdate-dynamic-thing-group例では、指定された動的モノのグループを更新します。説明を提供し、クエリ文字列を更新してグループのメンバーシップ条件を変更します。

```
aws iot update-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm"  
  --thing-group-properties "thingGroupDescription=\"This thing group contains  
rooms warmer than 65F.\"\" \  
  --query-string "attributes.temperature>65"
```

出力:

```
{
  "version": 2
}
```

詳細については、IoT デベロッパーガイドの「[Dynamic Thing Groups](#)」を参照してください。
AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDynamicThingGroup](#)」の「」を参照してください。

update-event-configurations

次の例は、update-event-configurations を使用する方法を説明しています。

AWS CLI

公開されるイベントタイプを表示するには

次のupdate-event-configurations例では、CA 証明書が追加、更新、または削除されたときにメッセージを有効にするように設定を更新します。

```
aws iot update-event-configurations \
  --event-configurations "{\"CA_CERTIFICATE\":{\"Enabled\":true}}"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[イベントメッセージ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateEventConfigurations](#)」の「」を参照してください。

update-indexing-configuration

次の例は、update-indexing-configuration を使用する方法を説明しています。

AWS CLI

モノのインデックス作成を有効にするには

次のupdate-indexing-configuration例では、_AWS Things インデックスを使用して、モノのインデックス作成がレジストリデータ、シャドウデータ、モノの接続ステータスの検索をサポートできるようにします。


```
aws iot update-indexing-configuration
  --thing-indexing-
configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

このコマンドでは何も出力されません。

詳細については、IoT デベロッパーガイドの「[モノのインデックス作成の管理](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIndexingConfiguration](#)」の「」を参照してください。

update-job

次のコード例は、update-job を使用する方法を示しています。

AWS CLI

ジョブの詳細ステータスを取得するには

次のupdate-job例では、ID が のジョブの詳細ステータスを取得しますexample-job-01。

```
aws iot describe-job \
  --job-id "example-job-01"
```

出力:

```
{
  "job": {
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
    "jobId": "example-job-01",
    "targetSelection": "SNAPSHOT",
    "status": "IN_PROGRESS",
    "targets": [
      "arn:aws:iot:us-west-2:123456789012:thing/MyRaspberryPi"
    ],
    "description": "example job test",
    "presignedUrlConfig": {},
    "jobExecutionsRolloutConfig": {},
    "createdAt": 1560787022.733,
    "lastUpdatedAt": 1560787026.294,
    "jobProcessDetails": {
```

```

        "numberOfCanceledThings": 0,
        "numberOfSucceededThings": 0,
        "numberOfFailedThings": 0,
        "numberOfRejectedThings": 0,
        "numberOfQueuedThings": 1,
        "numberOfInProgressThings": 0,
        "numberOfRemovedThings": 0,
        "numberOfTimedOutThings": 0
    },
    "timeoutConfig": {}
}

```

詳細については、AWS IoT デベロッパーガイドの「[ジョブの作成と管理 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateJob](#)」の「」を参照してください。

update-mitigation-action

次のコード例は、update-mitigation-action を使用する方法を示しています。

AWS CLI

緩和アクションを更新するには

次のupdate-mitigation-action例では、という名前の指定された緩和アクションを更新しAddThingsToQuarantineGroupAction、モノのグループ名を変更し、overrideDynamicGroups を に設定しますfalse。describe-mitigation-action コマンドを使用して変更を確認できます。

```

aws iot update-mitigation-action \
  --cli-input-json "{ \"actionName\": \"AddThingsToQuarantineGroupAction\",
  \"actionParams\": { \"addThingsToThingGroupParams\": {\"thingGroupNames\":
  [\"QuarantineGroup2\"],\"overrideDynamicGroups\": false}}}"

```

出力:

```

{
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
  AddThingsToQuarantineGroupAction",

```

```
"actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa"
}
```

詳細については、AWS IoT デベロッパーガイドの[UpdateMitigationAction](#)「(緩和アクションコマンド)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateMitigationAction](#)」の「」を参照してください。

update-provisioning-template

次の例は、update-provisioning-template を使用する方法を説明しています。

AWS CLI

プロビジョニングテンプレートを更新するには

次のupdate-provisioning-template例では、指定されたプロビジョニングテンプレートの説明とロール arn を変更し、テンプレートを有効にします。

```
aws iot update-provisioning-template \  
  --template-name widget-template \  
  --enabled \  
  --description "An updated provisioning template for widgets" \  
  --provisioning-role-arn arn:aws:iam::504350838278:role/Provision_role
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT Core デベロッパーガイド」の「IoT Secure Tunneling」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateProvisioningTemplate](#)」の「」を参照してください。

update-role-alias

次のコード例は、update-role-alias を使用する方法を示しています。

AWS CLI

ロールエイリアスを更新するには

次のupdate-role-alias例では、LightBulbRoleロールエイリアスを更新します。

```
aws iot update-role-alias \  
  --role-alias LightBulbRole \  
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

出力:

```
{  
  "roleAlias": "LightBulbRole",  
  "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"  
}
```

詳細については、AWS IoT APIリファレンスの[UpdateRoleAlias](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateRoleAlias](#)」の「」を参照してください。

update-scheduled-audit

次のコード例は、update-scheduled-audit を使用する方法を示しています。

AWS CLI

スケジュールされた監査定義を更新するには

次のupdate-scheduled-audit例では、AWS IoT Device Defender スケジュールされた監査のターゲットチェック名を変更します。

```
aws iot update-scheduled-audit \  
  --scheduled-audit-name WednesdayCertCheck \  
  --target-check-  
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

出力:

```
{  
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/  
WednesdayCertCheck"  
}
```

詳細については、AWS「IoT デベロッパーガイド」の[「監査コマンド」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateScheduledAudit](#)」の「」を参照してください。

update-security-profile

次のコード例は、update-security-profile を使用する方法を示しています。

AWS CLI

セキュリティプロファイルを変更するには

次のupdate-security-profile例では、AWS IoT Device Defender セキュリティプロファイルの説明と動作の両方を更新します。

```
aws iot update-security-profile \  
  --security-profile-name PossibleIssue \  
  --security-profile-description "Check to see if authorization fails 12 times in  
5 minutes or if cellular bandwidth exceeds 128" \  
  --behaviors "[{\\"name\\":\\"CellularBandwidth\\",\\"metric\\":\\"aws:message-byte-size  
\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":128},  
\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}},{\\"name  
\\":\\"Authorization\\",\\"metric\\":\\"aws:num-authorization-failures\\",\\"criteria\\":  
{\\"comparisonOperator\\":\\"less-than\\",\\"value\\":{\\"count\\":12},\\"durationSeconds  
\\":300,\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}]]"
```

出力:

```
{  
  "securityProfileName": "PossibleIssue",  
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/  
PossibleIssue",  
  "securityProfileDescription": "check to see if authorization fails 12 times in 5  
minutes or if cellular bandwidth exceeds 128",  
  "behaviors": [  
    {  
      "name": "CellularBandwidth",  
      "metric": "aws:message-byte-size",  
      "criteria": {  
        "comparisonOperator": "greater-than",  
        "value": {  
          "count": 128  
        }  
      },  
    },  
  ],  
}
```

```

        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    }
},
{
    "name": "Authorization",
    "metric": "aws:num-authorization-failures",
    "criteria": {
        "comparisonOperator": "less-than",
        "value": {
            "count": 12
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    }
}
],
"version": 2,
"creationDate": 1560278102.528,
"lastModifiedDate": 1560352711.207
}

```

詳細については、AWS IoT デベロッパガイドの「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSecurityProfile](#)」の「」を参照してください。

update-stream

次の例は、update-stream を使用する方法を説明しています。

AWS CLI

ストリームを更新するには

次のupdate-stream例では、既存のストリームを更新します。ストリームバージョンは 1 ずつ増加します。

```
aws iot update-stream \
  --cli-input-json file://update-stream.json
```

update-stream.json の内容:

```
{
  "streamId": "stream12345",
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
  "files": [
    {
      "fileId": 123,
      "s3Location": {
        "bucket": "codesign-ota-bucket",
        "key": "48c67f3c-63bb-4f92-a98a-4ee0fbc2bef6"
      }
    }
  ]
  "roleArn": "arn:aws:iam:us-west-2:123456789012:role/service-role/my_ota_stream_role"
}
```

出力:

```
{
  "streamId": "stream12345",
  "streamArn": "arn:aws:iot:us-west-2:123456789012:stream/stream12345",
  "description": "This stream is used for Amazon FreeRTOS OTA Update 12345.",
  "streamVersion": 2
}
```

詳細については、AWS IoT APIリファレンスの[UpdateStream](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateStream](#)」の「」を参照してください。

update-thing-group

次の例は、update-thing-group を使用する方法を説明しています。

AWS CLI

モノのグループの定義を更新するには

次のupdate-thing-group例では、指定されたモノグループの定義を更新し、説明と2つの属性を変更します。

```
aws iot update-thing-group \
```

```
--thing-group-name HalogenBulbs \  
--thing-group-properties "thingGroupDescription=\"Halogen bulb group\",  
attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

出力:

```
{  
  "version": 2  
}
```

詳細については、IoT デベロッパーガイドの「[モノのグループ](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateThingGroup](#)」の「」を参照してください。

update-thing-groups-for-thing

次の例は、update-thing-groups-for-thing を使用する方法を説明しています。

AWS CLI

モノが属するグループを変更するには

次のupdate-thing-groups-for-thing例では、 という名前のモノを という名前のグループMyLightBulbから削除DeadBulbsし、replaceableItems同時に という名前のグループに追加します。

```
aws iot update-thing-groups-for-thing \  
--thing-name MyLightBulb \  
--thing-groups-to-add "replaceableItems" \  
--thing-groups-to-remove "DeadBulbs"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[モノのグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateThingGroupsForThing](#)」の「」を参照してください。

update-thing

次の例は、update-thing を使用する方法を説明しています。

AWS CLI

モノをモノタイプに関連付けるには

次のupdate-thing例では、AWS IoT レジストリ内のモノをモノタイプに関連付けます。関連付けを行うときは、モノのタイプで定義された属性の値を指定します。

```
aws iot update-thing \  
  --thing-name "MyOtherLightBulb" \  
  --thing-type-name "LightBulb" \  
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

このコマンドは出力を生成しません。describe-thing コマンドを使用して結果を表示します。

詳細については、IoT デベロッパーガイドの「[モノのタイプ](#)」を参照してください。AWS IoT
• API 詳細については、AWS CLI 「コマンドリファレンス[UpdateThing](#)」の「」を参照してください。

update-topic-rule-destination

次の例は、update-topic-rule-destination を使用する方法を説明しています。

AWS CLI

例 1: トピックルールの送信先を有効にするには

次のupdate-topic-rule-destination例では、トピックルールの送信先へのトラフィックを有効にします。

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status ENABLED
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[トピックルールの送信先を有効にする](#)」を参照してください。

例 2: トピックルールの送信先を無効にするには

次のupdate-topic-rule-destination例では、トピックルールの送信先へのトラフィックを無効にします。

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status DISABLED
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[トピックルールの送信先を無効にする](#)」を参照してください。

例 3: 新しい確認メッセージを送信するには

次のupdate-topic-rule-destination例では、トピックルールの送信先に関する新しい確認メッセージを送信します。

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status IN_PROGRESS
```

このコマンドでは何も出力されません。

詳細については、AWS IoT デベロッパーガイドの「[新しい確認メッセージの送信](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTopicRuleDestination](#)」の「」を参照してください。

validate-security-profile-behaviors

次の例は、validate-security-profile-behaviors を使用する方法を説明しています。

AWS CLI

例 1: セキュリティプロファイルの動作パラメータを検証するには

次のvalidate-security-profile-behaviors例では、AWS IoT Device Defender セキュリティプロファイルに対して、適切に形成された正しい動作セットを検証します。

```
aws iot validate-security-profile-behaviors \  
  --behaviors "[{\\"name\\":\\"CellularBandwidth\\",\\"metric\\":\\"aws:message-byte-size\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":128},\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}},{\\"name\\":\\"Authorization\\",\\"metric\\":\\"aws:num-authorization-failures\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":12},\\"durationSeconds\\":300,\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}}]"
```

出力:

```
{  
  "valid": true,  
  "validationErrors": []  
}
```

例 2: セキュリティプロファイルの不正な動作パラメータを検証するには

次のvalidate-security-profile-behaviors例では、AWS IoT Device Defender セキュリティプロファイルのエラーを含む一連の動作を検証します。

```
aws iot validate-security-profile-behaviors \  
  --behaviors "[{\\"name\\":\\"CellularBandwidth\\",\\"metric\\":\\"aws:message-byte-size\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":128},\\"consecutiveDatapointsToAlarm\\":1,\\"consecutiveDatapointsToClear\\":1}},{\\"name\\":\\"Authorization\\",\\"metric\\":\\"aws:num-authorization-failures\\",\\"criteria\\":{\\"comparisonOperator\\":\\"greater-than\\",\\"value\\":{\\"count\\":12},\\"durationSeconds\\":300,\\"consecutiveDatapointsToAlarm\\":100000,\\"consecutiveDatapointsToClear\\":1}}]"
```

出力:

```
{  
  "valid": false,  
  "validationErrors": [  
    {  
      "errorMessage": "Behavior Authorization is malformed.  
consecutiveDatapointsToAlarm 100000 should be in range[1,10]"  
    }  
  ]  
}
```

詳細については、AWS 「IoT デベロッパーガイド」の「[コマンドの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ValidateSecurityProfileBehaviors](#)」の「」を参照してください。

AWS IoT 1-Click を使用したデバイスの例 AWS CLI

次のコード例は、を AWS IoT 1-Click デバイス AWS Command Line Interface で使用することでアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

claim-devices-by-claim-code

次のコード例は、claim-devices-by-claim-code を使用する方法を示しています。

AWS CLI

クレームコードを使用して 1 つ以上の AWS IoT 1-Click デバイスをクレームするには

次のclaim-devices-by-claim-code例では、(デバイス ID の代わりに) クレームコードを使用して、指定された AWS IoT 1-Click デバイスをクレームします。

```
aws iot1click-devices claim-devices-by-claim-code \  
  --claim-code C-123EXAMPLE
```

出力:

```
{
```

```
"Total": 9
"ClaimCode": "C-123EXAMPLE"
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ClaimDevicesByClaimCode](#)」の「」を参照してください。

describe-device

次のコード例は、describe-device を使用する方法を示しています。

AWS CLI

デバイスを説明するには

次のdescribe-device例では、指定されたデバイスについて説明します。

```
aws iot1click-devices describe-device \
  --device-id G030PM0123456789
```

出力:

```
{
  "DeviceDescription": {
    "Arn": "arn:aws:iot1click:us-west-2:012345678901:devices/G030PM0123456789",
    "Attributes": {
      "projectRegion": "us-west-2",
      "projectName": "AnytownDumpsters",
      "placementName": "customer217",
      "deviceTemplateName": "empty-dumpster-request"
    },
    "DeviceId": "G030PM0123456789",
    "Enabled": false,
    "RemainingLife": 99.9,
    "Type": "button",
    "Tags": {}
  }
}
```

詳細については、[AWS IoT 1-Click デベロッパーガイド](#)の「[での AWS CLI AWS IoT 1-Click の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeDevice](#)」の「」を参照してください。

finalize-device-claim

次のコード例は、finalize-device-claim を使用する方法を示しています。

AWS CLI

デバイス ID を使用して AWS IoT 1-Click デバイスのクレームリクエストを確定するには

次のfinalize-device-claim例では、(クレームコードの代わりに) デバイス ID を使用して、指定された AWS IoT 1-Click デバイスのクレームリクエストを確定します。

```
aws iot1click-devices finalize-device-claim \  
  --device-id G030PM0123456789
```

出力:

```
{  
  "State": "CLAIMED"  
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスFinalizeDeviceClaim](#)」の「」を参照してください。

get-device-methods

次の例は、get-device-methods を使用する方法を説明しています。

AWS CLI

デバイスで使用できるメソッドを一覧表示するには

次のget-device-methods例では、デバイスの使用可能なメソッドを一覧表示します。

```
aws iot1click-devices get-device-methods \  
--device-id G030PM0123456789
```

出力:

```
{  
  "DeviceMethods": [  
    {  
      "MethodName": "getDeviceHealthParameters"  
    },  
    {  
      "MethodName": "setDeviceHealthMonitorCallback"  
    },  
    {  
      "MethodName": "getDeviceHealthMonitorCallback"  
    },  
    {  
      "MethodName": "setOnClickCallback"  
    },  
    {  
      "MethodName": "getOnClickCallback"  
    }  
  ]  
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeviceMethods](#)」の「」を参照してください。

initiate-device-claim

次のコード例は、`initiate-device-claim` を使用する方法を示しています。

AWS CLI

デバイス ID を使用して AWS IoT 1-Click デバイスのクレームリクエストを開始するには

次の `initiate-device-claim` 例では、(クレームコードの代わりに) デバイス ID を使用して、指定された AWS IoT 1-Click デバイスのクレームリクエストを開始します。

```
aws iot1click-devices initiate-device-claim \  
  --device-id G030PM0123456789
```

出力:

```
{  
  "State": "CLAIM_INITIATED"  
}
```

詳細については、[AWS IoT 1-Click デベロッパーガイド](#)の「[での AWS CLI AWS IoT 1-Click の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスInitiateDeviceClaim](#)」の「[」](#)を参照してください。

invoke-device-method

次の例は、`invoke-device-method` を使用する方法を説明しています。

AWS CLI

デバイスでデバイスメソッドを呼び出すには

次の`invoke-device-method`例では、デバイスで指定されたメソッドを呼び出します。

```
aws iot1click-devices invoke-device-method \  
  --cli-input-json file://invoke-device-method.json
```

`invoke-device-method.json` の内容:

```
{  
  "DeviceId": "G030PM0123456789",  
  "DeviceMethod": {  
    "DeviceType": "device",  
    "MethodName": "getDeviceHealthParameters"  
  }  
}
```

出力:

```
{
```



```
"DeviceMethodResponse": "{\"remainingLife\": 99.8}"
}
```

詳細については、[AWS IoT 1-Click デベロッパーガイド](#)の「[での AWS CLI AWS IoT 1-Click の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス InvokeDeviceMethod](#)」の「」を参照してください。

list-device-events

次の例は、list-device-events を使用する方法を説明しています。

AWS CLI

指定した時間範囲のデバイスのイベントを一覧表示するには

次のlist-device-events例では、指定された時間範囲の指定されたデバイスのイベントを一覧表示します。

```
aws iot1click-devices list-device-events \
  --device-id G030PM0123456789 \
  --from-time-stamp 2019-07-17T15:45:12.880Z --to-time-
stamp 2019-07-19T15:45:12.880Z
```

出力:

```
{
  "Events": [
    {
      "Device": {
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
      },
      "StdEvent": "{\"clickType\": \"SINGLE\"",
      "\"reportedTime\": \"2019-07-18T23:47:55.015Z\", \"certificateId\":",
      "\"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",",
      "\"remainingLife\": 99.85000000000001, \"testMode\": false}"
    },
    {
      "Device": {
```

```
        "Attributes": {},
        "DeviceId": "G030PM0123456789",
        "Type": "button"
    },
    "StdEvent": "{\"clickType\": \"DOUBLE\",
\"reportedTime\": \"2019-07-19T00:14:41.353Z\", \"certificateId\":
\"fe8798a6c97c62ef8756b80eeefdcf2280f3352f82faa8080c74cc4f4a4d1811\",
\"remainingLife\": 99.8, \"testMode\": false}"
    }
]
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeviceEvents](#)」の「」を参照してください。

list-devices

次の例は、list-devices を使用する方法を説明しています。

AWS CLI

指定されたタイプのデバイスを一覧表示するには

次のlist-devices例では、指定されたタイプのデバイスを一覧表示します。

```
aws iot1click-devices list-devices \
  --device-type button
```

このコマンドでは何も出力されません。

出力:

```
{
  "Devices": [
    {
      "remainingLife": 99.9,
      "attributes": {
        "arn": "arn:aws:iot1click:us-west-2:123456789012:devices/
G030PM0123456789",
```

```
        "type": "button",
        "deviceId": "G030PM0123456789",
        "enabled": false
      }
    ]
  }
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「で AWS CLI AWS IoT 1-Click を使用する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDevices](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用方法を説明しています。

AWS CLI

デバイスのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたデバイスのタグを一覧表示します。

```
aws iot1click-devices list-tags-for-resource \
  --resource-arn "arn:aws:iot1click:us-west-2:012345678901:devices/
  G030PM0123456789"
```

出力:

```
{
  "Tags": {
    "Driver Phone": "123-555-0199",
    "Driver": "Jorge Souza"
  }
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

デバイス AWS リソースにタグを追加するには

次のtag-resource例では、指定されたリソースに 2 つのタグを追加します。

```
aws iot1click-devices tag-resource \  
  --cli-input-json file://devices-tag-resource.json
```

devices-tag-resource.json の内容:

```
{  
  "ResourceArn": "arn:aws:iot1click:us-west-2:123456789012:devices/  
G030PM0123456789",  
  "Tags": {  
    "Driver": "Jorge Souza",  
    "Driver Phone": "123-555-0199"  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

unclaim-device

次の例は、unclaim-device を使用する方法を説明しています。

AWS CLI

AWS アカウントからデバイスを請求 (登録解除) するには

次のunclaim-device例では、指定されたデバイスを AWS アカウントから要求解除 (登録解除) します。

```
aws iot1click-devices unclaim-device \  
  --device-id G030PM0123456789
```

出力:

```
{  
  "State": "UNCLAIMED"  
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UnclaimDevice](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

デバイス AWS リソースからタグを削除するには

次のuntag-resource例では、指定されたデバイスリソースDriverから Driver Phoneおよびという名前のタグを削除します。

```
aws iot1click-devices untag-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters" \  
  --tag-keys "Driver Phone" "Driver"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-device-state

次の例は、update-device-state を使用する方法を説明しています。

AWS CLI

デバイスの「有効」状態を更新するには

次に、指定されたデバイスの状態を update-device-state に設定します `enabled`。

```
aws iot1click-devices update-device-state \  
  --device-id G030PM0123456789 \  
  --enabled
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス UpdateDeviceState](#)」の「」を参照してください。

AWS IoT 1-Click を使用したプロジェクトの例 AWS CLI

次のコード例は、を AWS IoT 1-Click Projects AWS Command Line Interface で使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-device-with-placement

次の例は、`associate-device-with-placement` を使用する方法を説明しています。

AWS CLI

AWS IoT 1-Click デバイスを既存の配置に関連付けるには

次の`associate-device-with-placement`例では、指定された AWS IoT 1-Click デバイスを既存の配置に関連付けます。

```
aws iot1click-projects associate-device-with-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --device-template-name empty-dumpster-request \  
  --device-id G030PM0123456789
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateDeviceWithPlacement](#)」の「」を参照してください。

create-placement

次のコード例は、`create-placement` を使用する方法を示しています。

AWS CLI

プロジェクトの AWS IoT 1-Click 配置を作成するには

次の`create-placement`例では、指定されたプロジェクトの AWS IoT 1-Click 配置を作成します。

```
aws iot1click-projects create-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --device-template-name empty-dumpster-request
```

```
--attributes '{"location": "123 Any Street Anytown, USA 10001", "phone": "123-456-7890"}'
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT 1-Click デベロッパーガイド](#)の「[での AWS CLI AWS IoT 1-Click の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreatePlacement](#)」の「」を参照してください。

create-project

次のコード例は、create-project を使用する方法を示しています。

AWS CLI

ゼロ以上の配置で AWS IoT 1-Click プロジェクトを作成するには

次のcreate-project例では、配置用の AWS IoT 1-Click プロジェクトを作成します。

```
aws iot1click-projects create-project --cli-input-json file://create-project.json
```

create-project.json の内容:

```
{
  "projectName": "AnytownDumpsters",
  "description": "All dumpsters in the Anytown region.",
  "placementTemplate": {
    "defaultAttributes": {
      "City" : "Anytown"
    },
    "deviceTemplates": {
      "empty-dumpster-request" : {
        "deviceType": "button"
      }
    }
  }
}
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProject](#)」の「」を参照してください。

delete-placement

次のコード例は、delete-placement を使用する方法を示しています。

AWS CLI

プロジェクトからプレイスメントを削除するには

次のdelete-placement例では、指定された配置をプロジェクトから削除します。

```
aws iot1click-projects delete-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePlacement](#)」の「」を参照してください。

delete-project

次の例は、delete-project を使用する方法を説明しています。

AWS CLI

AWS アカウントからプロジェクトを削除するには

次のdelete-project例では、指定されたプロジェクトを AWS アカウントから削除します。

```
aws iot1click-projects delete-project \  
  --project-name AnytownDumpsters
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT 1-Click デベロッパーガイド](#)の「[での AWS CLI AWS IoT 1-Click の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteProject](#)」の「」を参照してください。

describe-placement

次のコード例は、describe-placement を使用する方法を示しています。

AWS CLI

プロジェクトの配置を記述するには

次のdescribe-placement例では、指定されたプロジェクトの配置について説明します。

```
aws iot1click-projects describe-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

出力:

```
{  
  "placement": {  
    "projectName": "AnytownDumpsters",  
    "placementName": "customer217",  
    "attributes": {  
      "phone": "123-555-0110",  
      "location": "123 Any Street Anytown, USA 10001"  
    },  
    "createdDate": 1563488454,  
    "updatedAt": 1563488454  
  }  
}
```

詳細については、[AWS IoT 1-Click デベロッパーガイド](#)の「[での AWS CLI AWS IoT 1-Click の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribePlacement](#)」の「」を参照してください。

describe-project

次の例は、describe-project を使用する方法を説明しています。

AWS CLI

AWS IoT 1-Click プロジェクトを記述するには

次のdescribe-project例では、指定された AWS IoT 1-Click プロジェクトについて説明します。

```
aws iot1click-projects describe-project \  
  --project-name AnytownDumpsters
```

出力:

```
{  
  "project": {  
    "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/AnytownDumpsters",  
    "projectName": "AnytownDumpsters",  
    "description": "All dumpsters in the Anytown region.",  
    "createdDate": 1563483100,  
    "updatedAt": 1563483100,  
    "placementTemplate": {  
      "defaultAttributes": {  
        "City": "Anytown"  
      },  
      "deviceTemplates": {  
        "empty-dumpster-request": {  
          "deviceType": "button",  
          "callbackOverrides": {}  
        }  
      }  
    },  
    "tags": {}  
  }  
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeProject](#)」の「」を参照してください。

disassociate-device-from-placement

次の例は、disassociate-device-from-placement を使用する方法を説明しています。

AWS CLI

配置からデバイスの関連付けを解除するには

次のdisassociate-device-from-placement例では、指定されたデバイスをプレイスメントから関連付け解除します。

```
aws iot1click-projects disassociate-device-from-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --device-template-name empty-dumpster-request
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「で AWS CLI AWS IoT 1-Click を使用する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateDeviceFromPlacement](#)」の「」を参照してください。

get-devices-in-placement

次のコード例は、get-devices-in-placement を使用する方法を示しています。

AWS CLI

プロジェクトに含まれる配置内のすべてのデバイスを一覧表示するには

次のget-devices-in-placement例では、指定されたプロジェクトに含まれる指定された配置内のすべてのデバイスを一覧表示します。

```
aws iot1click-projects get-devices-in-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

出力:

```
{  
  "devices": {
```

```
    "empty-dumpster-request": "G030PM0123456789"
  }
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDevicesInPlacement](#)」の「」を参照してください。

list-placements

次の例は、list-placements を使用する方法を説明しています。

AWS CLI

プロジェクトのすべての AWS IoT 1-Click 配置を一覧表示するには

次のlist-placements例では、指定されたプロジェクトのすべての AWS IoT 1-Click 配置を一覧表示します。

```
aws iot1click-projects list-placements \
  --project-name AnytownDumpsters
```

出力:

```
{
  "placements": [
    {
      "projectName": "AnytownDumpsters",
      "placementName": "customer217",
      "createdDate": 1563488454,
      "updatedAt": 1563488454
    }
  ]
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPlacements](#)」の「」を参照してください。

list-projects

次の例は、list-projects を使用する方法を説明しています。

AWS CLI

すべての AWS IoT 1-Click プロジェクトを一覧表示するには

次の list-projects 例では、アカウント内のすべての AWS IoT 1-Click プロジェクトを一覧表示します。

```
aws iot1click-projects list-projects
```

出力:

```
{
  "projects": [
    {
      "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/AnytownDumpsters",
      "projectName": "AnytownDumpsters",
      "createdDate": 1563483100,
      "updatedAt": 1563483100,
      "tags": {}
    }
  ]
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListProjects](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

プロジェクトリソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたプロジェクトリソースのタグを一覧表示します。

```
aws iot1click-projects list-tags-for-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
  AnytownDumpsters"
```

出力:

```
{  
  "tags": {  
    "Manager": "Li Juan",  
    "Account": "45215"  
  }  
}
```

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

プロジェクトリソースにタグを追加するには

次のtag-resource例では、指定されたプロジェクトリソースに 2 つのタグを追加します。

```
aws iot1click-projects tag-resource \  
  --cli-input-json file://devices-tag-resource.json
```

devices-tag-resource.json の内容:

```
{  
  "resourceArn": "arn:aws:iot1click:us-west-2:123456789012:projects/  
  AnytownDumpsters",
```

```
"tags": {
  "Account": "45215",
  "Manager": "Li Juan"
}
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次の例は、`untag-resource` を使用する方法を説明しています。

AWS CLI

プロジェクトリソースからタグを削除するには

次の `untag-resource` 例では、指定されたプロジェクト Manager からキー名のタグを削除します。

```
aws iot1click-projects untag-resource \
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters" \
  --tag-keys "Manager"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-placement

次の例は、`update-placement` を使用する方法を説明しています。

AWS CLI

プレースメントの「属性」キーと値のペアを更新するには

次のupdate-placement例では、プレースメントの「属性」キーと値のペアを更新します。

```
aws iot1click-projects update-placement \  
  --cli-input-json file://update-placement.json
```

update-placement.json の内容:

```
{  
  "projectName": "AnytownDumpsters",  
  "placementName": "customer217",  
  "attributes": {  
    "phone": "123-456-7890",  
    "location": "123 Any Street Anytown, USA 10001"  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT 1-Click デベロッパーガイド](#)の「[での AWS CLI](#) AWS IoT 1-Click の使用」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdatePlacement](#)」の「」を参照してください。

update-project

次のコード例は、update-project を使用する方法を示しています。

AWS CLI

プロジェクトの設定を更新するには

次のupdate-project例では、プロジェクトの説明を更新します。

```
aws iot1click-projects update-project \  
  --project-name AnytownDumpsters \  
  --description "All dumpsters (yard waste, recycling, garbage) in the Anytown region."
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT 1-Click デベロッパーガイド」の「での AWS CLI AWS IoT 1-Click の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateProject](#)」の「」を参照してください。

AWS IoT Analytics を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Analytics。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-put-message

次の例は、batch-put-message を使用する方法を説明しています。

AWS CLI

チャンネルにメッセージを送信するには

次のbatch-put-message例は、指定されたチャンネルにメッセージを送信します。

```
aws iotanalytics batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://batch-put-message.json
```

batch-put-message.json の内容:

```
{
  "channelName": "mychannel",
  "messages": [
    {
      "messageId": "0001",
      "payload": "eyJhdGVtcGVyYXR1cmUiOiAyMCB9"
    }
  ]
}
```

出力:

```
{
  "batchPutMessageErrorEntries": []
}
```

詳細については、AWS IoT Analytics APIリファレンスの[BatchPutMessage](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[BatchPutMessage](#)」の「」を参照してください。

cancel-pipeline-reprocessing

次のコード例は、cancel-pipeline-reprocessing を使用する方法を示しています。

AWS CLI

パイプラインを介したデータの再処理をキャンセルするには

次のcancel-pipeline-reprocessing例では、指定されたパイプラインを介したデータの再処理をキャンセルします。

```
aws iotanalytics cancel-pipeline-reprocessing \
  --pipeline-name mypipeline \
  --reprocessing-id "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[CancelPipelineReprocessing](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CancelPipelineReprocessing](#)」の「」を参照してください。

create-channel

次の例は、create-channel を使用する方法を説明しています。

AWS CLI

チャンネルを作成するには

次のcreate-channel例では、指定された設定でチャンネルを作成します。チャンネルは、パイプラインにデータを公開する前に、MQTTトピックからデータを収集し、未処理の未処理のメッセージをアーカイブします。

```
aws iotanalytics create-channel \  
  --cli-input-json file://create-channel.json
```

create-channel.json の内容:

```
{  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "unlimited": true  
  },  
  "tags": [  
    {  
      "key": "Environment",  
      "value": "Production"  
    }  
  ]  
}
```

出力:

```
{  
  "channelArn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel",  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "unlimited": true  
  }  
}
```

```
}
```

詳細については、AWS 「IoT Analytics APIリファレンス[CreateChannel](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateChannel](#)」の「」を参照してください。

create-dataset-content

次の例は、create-dataset-content を使用する方法を説明しています。

AWS CLI

データセットのコンテンツを作成するには

次のcreate-dataset-content例では、queryAction (SQLクエリ) または containerAction (コンテナ化されたアプリケーションを実行) を適用して、指定されたデータセットの内容を作成します。

```
aws iotanalytics create-dataset-content \  
  --dataset-name mydataset
```

出力:

```
{  
  "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"  
}
```

詳細については、AWS IoT Analytics APIリファレンスの[CreateDatasetContent](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDatasetContent](#)」の「」を参照してください。

create-dataset

次の例は、create-dataset を使用する方法を説明しています。

AWS CLI

データセットを作成するには

次のcreate-dataset例では、データセットを作成します。データセットは、queryAction (SQLクエリ) または containerAction (コンテナ化されたアプリケーションの実行) を適用してデータストアから取得したデータを保存します。このオペレーションによりデータセットのスケルトンが作成されます。データセットを手動で入力するには、 を呼び出しCreateDatasetContentたり、指定した に従って自動的に入力triggerしたりできます。

```
aws iotanalytics create-dataset \  
  --cli-input-json file://create-dataset.json
```

create-dataset.json の内容:

```
{  
  "datasetName": "mydataset",  
  "actions": [  
    {  
      "actionName": "myDatasetAction",  
      "queryAction": {  
        "sqlQuery": "SELECT * FROM mydatastore"  
      }  
    }  
  ],  
  "retentionPeriod": {  
    "unlimited": true  
  },  
  "tags": [  
    {  
      "key": "Environment",  
      "value": "Production"  
    }  
  ]  
}
```

出力:

```
{  
  "datasetName": "mydataset",  
  "retentionPeriod": {  
    "unlimited": true  
  },  
  "datasetArn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"  
}
```

詳細については、AWS IoT Analytics APIリファレンスの[CreateDataset](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateDataset](#)」の「」を参照してください。

create-datastore

次の例は、create-datastore を使用する方法を説明しています。

AWS CLI

データストアを作成するには

次のcreate-datastore例では、メッセージのリポジトリであるデータストアを作成します。

```
aws iotanalytics create-datastore \  
  --cli-input-json file://create-datastore.json
```

create-datastore.json の内容:

```
{  
  "datastoreName": "mydatastore",  
  "retentionPeriod": {  
    "numberOfDays": 90  
  },  
  "tags": [  
    {  
      "key": "Environment",  
      "value": "Production"  
    }  
  ]  
}
```

出力:

```
{  
  "datastoreName": "mydatastore",  
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/  
mydatastore",  
  "retentionPeriod": {  
    "numberOfDays": 90,  
  }  
}
```

```
    "unlimited": false
  }
}
```

詳細については、AWS IoT Analytics APIリファレンスの[CreateDatastore](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateDatastore](#)」の「」を参照してください。

create-pipeline

次のコード例は、create-pipeline を使用する方法を示しています。

AWS CLI

IoT Analytics Pipeline を作成する

次のcreate-pipeline例では、パイプラインを作成します。パイプラインではチャンネルからのメッセージが消費されるため、メッセージをデータストアに保存する前に処理できます。チャンネルとデータストアアクティビティの両方を指定し、オプションで、pipelineActivities配列内の最大 23 個の追加アクティビティを指定する必要があります。

```
aws iotanalytics create-pipeline \  
  --cli-input-json file://create-pipeline.json
```

create-pipeline.json の内容:

```
{  
  "pipelineName": "mypipeline",  
  "pipelineActivities": [  
    {  
      "channel": {  
        "name": "myChannelActivity",  
        "channelName": "mychannel",  
        "next": "myMathActivity"  
      }  
    },  
    {  
      "datastore": {  
        "name": "myDatastoreActivity",  
        "datastoreName": "mydatastore"  
      }  
    }  
  ]  
}
```



```
    }
  },
  {
    "math": {
      "name": "myMathActivity",
      "math": "((temp - 32) * 5.0) / 9.0",
      "attribute": "tempC",
      "next": "myDatastoreActivity"
    }
  }
],
"tags": [
  {
    "key": "Environment",
    "value": "Beta"
  }
]
}
```

出力:

```
{
  "pipelineArn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/
mypipeline",
  "pipelineName": "mypipeline"
}
```

詳細については、AWS IoT Analytics APIリファレンスの[CreatePipeline](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreatePipeline](#)」の「」を参照してください。

delete-channel

次の例は、delete-channel を使用する方法を説明しています。

AWS CLI

IoT Analytics Channel を削除する

次のdelete-channel例では、指定されたチャンネルを削除します。

```
aws iotanalytics delete-channel \  
  --channel-name mychannel
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[DeleteChannel](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteChannel](#)」の「」を参照してください。

delete-dataset-content

次のコード例は、delete-dataset-content を使用する方法を示しています。

AWS CLI

データセットコンテンツを削除するには

次のdelete-dataset-content例では、指定されたデータセットの内容を削除します。

```
aws iotanalytics delete-dataset-content \  
  --dataset-name mydataset
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[DeleteDatasetContent](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteDatasetContent](#)」の「」を参照してください。

delete-dataset

次のコード例は、delete-dataset を使用する方法を示しています。

AWS CLI

データセットを削除するには

次のdelete-dataset例では、指定されたデータセットを削除します。このオペレーションの実行前にデータセットのコンテンツを消去する必要はありません。

```
aws iotanalytics delete-dataset \  
  --dataset-name mydataset
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[DeleteDataset](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteDataset](#)」の「」を参照してください。

delete-datastore

次の例は、delete-datastore を使用する方法を説明しています。

AWS CLI

データストアを削除するには

次のdelete-datastore例では、指定されたデータストアを削除します。

```
aws iotanalytics delete-datastore \  
  --datastore-name mydatastore
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[DeleteDatastore](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteDatastore](#)」の「」を参照してください。

delete-pipeline

次の例は、delete-pipeline を使用する方法を説明しています。

AWS CLI

パイプラインを削除するには

次のdelete-pipeline例では、指定されたパイプラインを削除します。

```
aws iotanalytics delete-pipeline \  
  --pipeline-name mypipeline
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[DeletePipeline](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeletePipeline](#)」の「」を参照してください。

describe-channel

次のコード例は、describe-channel を使用する方法を示しています。

AWS CLI

チャンネルに関する情報を取得するには

次のdescribe-channel例では、指定したチャンネルの統計を含む詳細を表示します。

```
aws iotanalytics describe-channel \  
  --channel-name mychannel \  
  --include-statistics
```

出力:

```
{  
  "statistics": {  
    "size": {  
      "estimatedSizeInBytes": 402.0,  
      "estimatedOn": 1561504380.0  
    }  
  },  
  "channel": {  
    "status": "ACTIVE",  
    "name": "mychannel",  
    "lastUpdateTime": 1557860351.001,  
    "creationTime": 1557860351.001,  
    "retentionPeriod": {  
      "unlimited": true  
    }  
  },  
}
```

```
    "arn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
  }
}
```

詳細については、AWS 「IoT Analytics APIリファレンス[DescribeChannel](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeChannel](#)」の「」を参照してください。

describe-dataset

次のコード例は、describe-dataset を使用する方法を示しています。

AWS CLI

データセットに関する情報を取得するには

次のdescribe-dataset例では、指定されたデータセットの詳細を表示します。

```
aws iotanalytics describe-dataset \  
  --dataset-name mydataset
```

出力:

```
{  
  "dataset": {  
    "status": "ACTIVE",  
    "contentDeliveryRules": [],  
    "name": "mydataset",  
    "lastUpdateTime": 1557859240.658,  
    "triggers": [],  
    "creationTime": 1557859240.658,  
    "actions": [  
      {  
        "actionName": "query_32",  
        "queryAction": {  
          "sqlQuery": "SELECT * FROM mydatastore",  
          "filters": []  
        }  
      }  
    ],  
  },  
}
```

```
    "retentionPeriod": {
      "numberOfDays": 90,
      "unlimited": false
    },
    "arn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
  }
}
```

詳細については、AWS IoT Analytics APIリファレンスの[DescribeDataset](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeDataset](#)」の「」を参照してください。

describe-datastore

次の例は、describe-datastore を使用する方法を説明しています。

AWS CLI

データストアに関する情報を取得するには

次のdescribe-datastore例では、指定されたデータストアの統計を含む詳細を表示します。

```
aws iotanalytics describe-datastore \
  --datastore-name mydatastore \
  --include-statistics
```

出力:

```
{
  "datastore": {
    "status": "ACTIVE",
    "name": "mydatastore",
    "lastUpdateTime": 1557858971.02,
    "creationTime": 1557858971.02,
    "retentionPeriod": {
      "unlimited": true
    },
    "arn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/mydatastore"
  },
  "statistics": {
```

```
    "size": {
      "estimatedSizeInBytes": 397.0,
      "estimatedOn": 1561592040.0
    }
  }
}
```

詳細については、AWS IoT Analytics APIリファレンスの[DescribeDatastore](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeDatastore](#)」の「」を参照してください。

describe-logging-options

次の例は、describe-logging-options を使用する方法を説明しています。

AWS CLI

現在のログ記録オプションを取得するには

次のdescribe-logging-options例は、現在の AWS IoT Analytics ログ記録オプションを示しています。

```
aws iotanalytics describe-logging-options
```

このコマンドでは何も出力されません。出力:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
    "enabled": true,
    "level": "ERROR"
  }
}
```

詳細については、AWS IoT Analytics APIリファレンスの[DescribeLoggingOptions](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeLoggingOptions](#)」の「」を参照してください。

describe-pipeline

次の例は、describe-pipeline を使用する方法を説明しています。

AWS CLI

パイプラインに関する情報を取得するには

次のdescribe-pipeline例では、指定されたパイプラインの詳細を表示します。

```
aws iotanalytics describe-pipeline \  
  --pipeline-name mypipeline
```

出力:

```
{  
  "pipeline": {  
    "activities": [  
      {  
        "channel": {  
          "channelName": "mychannel",  
          "name": "mychannel_28",  
          "next": "mydatastore_29"  
        }  
      },  
      {  
        "datastore": {  
          "datastoreName": "mydatastore",  
          "name": "mydatastore_29"  
        }  
      }  
    ],  
    "name": "mypipeline",  
    "lastUpdateTime": 1561676362.515,  
    "creationTime": 1557859124.432,  
    "reprocessingSummaries": [  
      {  
        "status": "SUCCEEDED",  
        "creationTime": 1561676362.189,  
        "id": "6ad2764f-fb13-4de3-b101-4e74af03b043"  
      }  
    ],  
    "arn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/mypipeline"  
  }  
}
```



```
}  
}
```

詳細については、AWS IoT Analytics APIリファレンスの[DescribePipeline](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribePipeline](#)」の「」を参照してください。

get-dataset-content

次の例は、get-dataset-content を使用する方法を説明しています。

AWS CLI

データセットの内容を取得するには

次のget-dataset-content例では、データセットの内容を署名付きとして取得しますURIs。

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

出力:

```
{  
  "status": {  
    "state": "SUCCEEDED"  
  },  
  "timestamp": 1557863215.995,  
  "entries": [  
    {  
      "dataURI": "https://aws-radiant-  
dataset-12345678-1234-1234-1234-123456789012.s3.us-west-2.amazonaws.com/  
results/12345678-e8b3-46ba-b2dd-efe8d86cf385.csv?X-Amz-Security-Token=...-Amz-  
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190628T173437Z&X-Amz-SignedHeaders=host&X-  
Amz-Expires=7200&X-Amz-Credential=...F20190628%2Fus-west-2%2Fs3%2Faws4_request&X-  
Amz-Signature=..."  
    }  
  ]  
}
```

詳細については、ガイドの[GetDatasetContent](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDatasetContent](#)」の「」を参照してください。

list-channels

次の例は、list-channels を使用する方法を説明しています。

AWS CLI

チャンネルのリストを取得するには

次のlist-channels例では、使用可能なチャンネルの概要情報を表示します。

```
aws iotanalytics list-channels
```

出力:

```
{
  "channelSummaries": [
    {
      "status": "ACTIVE",
      "channelName": "mychannel",
      "creationTime": 1557860351.001,
      "lastUpdateTime": 1557860351.001
    }
  ]
}
```

詳細については、AWS IoT Analytics APIリファレンスの[ListChannels](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListChannels](#)」の「」を参照してください。

list-dataset-contents

次の例は、list-dataset-contents を使用する方法を説明しています。

AWS CLI

データセットの内容に関する情報を一覧表示するには

次のlist-dataset-contents例では、作成されたデータセットの内容に関する情報を示しています。

```
aws iotanalytics list-dataset-contents \  
  --dataset-name mydataset
```

出力:

```
{  
  "datasetContentSummaries": [  
    {  
      "status": {  
        "state": "SUCCEEDED"  
      },  
      "scheduleTime": 1557863215.995,  
      "version": "b10ea2a9-66c1-4d99-8d1f-518113b738d0",  
      "creationTime": 1557863215.995  
    }  
  ]  
}
```

詳細については、AWS IoT Analytics APIリファレンスの[ListDatasetContents](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListDatasetContents](#)」の「」を参照してください。

list-datasets

次の例は、list-datasets を使用する方法を説明しています。

AWS CLI

データセットに関する情報を取得するには

次のlist-datasets例では、使用可能なデータセットの概要情報を一覧表示します。

```
aws iotanalytics list-datasets
```

出力:

```
{
  "datasetSummaries": [
    {
      "status": "ACTIVE",
      "datasetName": "mydataset",
      "lastUpdateTime": 1557859240.658,
      "triggers": [],
      "creationTime": 1557859240.658,
      "actions": [
        {
          "actionName": "query_32",
          "actionType": "QUERY"
        }
      ]
    }
  ]
}
```

詳細については、AWS IoT Analytics APIリファレンスの[ListDatasets](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListDatasets](#)」の「」を参照してください。

list-datastores

次のコード例は、list-datastores を使用する方法を示しています。

AWS CLI

データストアのリストを取得するには

次のlist-datastores例では、使用可能なデータストアの概要情報を表示します。

```
aws iotanalytics list-datastores
```

出力:

```
{
  "datastoreSummaries": [
    {
      "status": "ACTIVE",
      "datastoreName": "mydatastore",

```

```
        "creationTime": 1557858971.02,  
        "lastUpdateTime": 1557858971.02  
      }  
    ]  
  }
```

詳細については、AWS IoT Analytics APIリファレンスの[ListDatastores](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListDatastores](#)」の「」を参照してください。

list-pipelines

次のコード例は、list-pipelines を使用する方法を示しています。

AWS CLI

パイプラインのリストを取得するには

次のlist-pipelines例では、使用可能なパイプラインのリストを表示します。

```
aws iotanalytics list-pipelines
```

出力:

```
{  
  "pipelineSummaries": [  
    {  
      "pipelineName": "mypipeline",  
      "creationTime": 1557859124.432,  
      "lastUpdateTime": 1557859124.432,  
      "reprocessingSummaries": []  
    }  
  ]  
}
```

詳細については、AWS IoT Analytics APIリファレンスの[ListPipelines](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListPipelines](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定したリソースにアタッチしたタグを一覧表示します。

```
aws iotanalytics list-tags-for-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
```

出力:

```
{  
  "tags": [  
    {  
      "value": "bar",  
      "key": "foo"  
    }  
  ]  
}
```

詳細については、AWS IoT Analytics APIリファレンスの[ListTagsForResource](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-logging-options

次のコード例は、put-logging-options を使用する方法を示しています。

AWS CLI

ログ記録オプションを設定または更新するには

次のput-logging-options例では、AWS IoT Analytics のログ記録オプションを設定または更新します。loggingOptions フィールドの値を更新すると、変更が有効になるまでに最大 1 分かかることがあります。また、roleArn「」フィールドで指定したロールにアタッチされたポリ

シーを変更する場合 (無効なポリシーを修正する場合など)、その変更が有効になるまでに最大 5 分かかることがあります。

```
aws iotanalytics put-logging-options \  
  --cli-input-json file://put-logging-options.json
```

put-logging-options.json の内容:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",  
    "level": "ERROR",  
    "enabled": true  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[PutLoggingOptions](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[PutLoggingOptions](#)」の「」を参照してください。

run-pipeline-activity

次の例は、run-pipeline-activity を使用する方法を説明しています。

AWS CLI

パイプラインアクティビティをシミュレートするには

次のrun-pipeline-activity例では、メッセージペイロードでパイプラインアクティビティを実行した結果をシミュレートします。

```
aws iotanalytics run-pipeline-activity \  
  --pipeline-activity file://maths.json \  
  --payloads file://payloads.json
```

maths.json の内容:

```
{
```

```
"math": {
  "name": "MyMathActivity",
  "math": "((temp - 32) * 5.0) / 9.0",
  "attribute": "tempC"
}
```

payloads.json の内容:

```
[
  "{\"humidity\": 52, \"temp\": 68 }",
  "{\"humidity\": 52, \"temp\": 32 }"
]
```

出力:

```
{
  "logResult": "",
  "payloads": [
    "eyJodW1pZG10eSI6NTIsInRlbXAiOiJY4LCJ0ZW1wQyI6MjB9",
    "eyJodW1pZG10eSI6NTIsInRlbXAiOiJyMyLCJ0ZW1wQyI6MH0="
  ]
}
```

詳細については、AWS IoT Analytics APIリファレンスの[RunPipelineActivity](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[RunPipelineActivity](#)」の「」を参照してください。

sample-channel-data

次のコード例は、sample-channel-data を使用する方法を示しています。

AWS CLI

チャンネルからサンプルメッセージを取得するには

次のsample-channel-data例では、指定された時間枠に取り込まれた指定されたチャンネルからメッセージのサンプルを取得します。最大 10 個のメッセージを取得できます。

```
aws iotanalytics sample-channel-data \
```



```
--channel-name mychannel
```

出力:

```
{
  "payloads": [
    "eyJhdGVtcGVyYXR1cmUiOiAyMCM9",
    "eyJhZm9vIjogImJhcnVzIj0="
  ]
}
```

詳細については、AWS IoT Analytics APIリファレンスの[SampleChannelData](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[SampleChannelData](#)」の「」を参照してください。

start-pipeline-reprocessing

次のコード例は、start-pipeline-reprocessing を使用する方法を示しています。

AWS CLI

パイプラインの再処理を開始するには

次のstart-pipeline-reprocessing例では、指定されたパイプラインを介して未加工メッセージデータの再処理を開始します。

```
aws iotanalytics start-pipeline-reprocessing \
  --pipeline-name mypipeline
```

出力:

```
{
  "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"
}
```

詳細については、AWS IoT Analytics APIリファレンスの[StartPipelineReprocessing](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[StartPipelineReprocessing](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを追加または変更するには

次のtag-resource例では、指定されたリソースにアタッチされたタグを に追加または変更します。

```
aws iotanalytics tag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tags "[{"key": "Environment", "value": "Production"}]"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[TagResource](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースからタグを削除する

次のuntag-resource例では、指定されたリソースから指定されたキー名のタグを削除します。

```
aws iotanalytics untag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tag-keys ["Environment"]"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの [UntagResource](https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UntagResource.html)「 < https:// docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UntagResource.html >」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-channel

次のコード例は、update-channel を使用する方法を示しています。

AWS CLI

チャンネルを変更するには

次のupdate-channel例では、指定されたチャンネルの設定を変更します。

```
aws iotanalytics update-channel \  
  --cli-input-json file://update-channel.json
```

update-channel.json の内容:

```
{  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "numberOfDays": 92  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[UpdateChannel](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateChannel](#)」の「」を参照してください。

update-dataset

次のコード例は、update-dataset を使用する方法を示しています。

AWS CLI

データセットを更新するには

次のupdate-dataset例では、指定されたデータセットの設定を変更します。

```
aws iotanalytics update-dataset \  
  --cli-input-json file://update-dataset.json
```

update-dataset.json の内容:

```
{  
  "datasetName": "mydataset",  
  "actions": [  
    {  
      "actionName": "myDatasetUpdateAction",  
      "queryAction": {  
        "sqlQuery": "SELECT * FROM mydatastore"  
      }  
    }  
  ],  
  "retentionPeriod": {  
    "numberOfDays": 92  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの UpdateDataset 「< https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UpdateDataset.html >」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateDataset](#)」の「」を参照してください。

update-datastore

次のコード例は、update-datastore を使用する方法を示しています。

AWS CLI

データストアを更新するには

次のupdate-datastore例では、指定されたデータストアの設定を変更します。

```
aws iotanalytics update-datastore \  
  --cli-input-json file://update-datastore.json
```

```
--cli-input-json file://update-datastore.json
```

update-datastore.json の内容 :

```
{
  "datastoreName": "mydatastore",
  "retentionPeriod": {
    "numberOfDays": 93
  }
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[UpdateDatastore](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDatastore](#)」の「」を参照してください。

update-pipeline

次の例は、update-pipeline を使用する方法を説明しています。

AWS CLI

パイプラインを更新するには

次のupdate-pipeline例では、指定されたパイプラインの設定を変更します。チャンネルとデータストアアクティビティの両方を指定し、オプションで 23 個の追加アクティビティをpipelineActivities配列に指定する必要があります。

```
aws iotanalytics update-pipeline \  
  --cli-input-json file://update-pipeline.json
```

update-pipeline.json の内容 :

```
{
  "pipelineName": "mypipeline",
  "pipelineActivities": [
    {
```

```
    "channel": {
      "name": "myChannelActivity",
      "channelName": "mychannel",
      "next": "myMathActivity"
    }
  },
  {
    "datastore": {
      "name": "myDatastoreActivity",
      "datastoreName": "mydatastore"
    }
  },
  {
    "math": {
      "name": "myMathActivity",
      "math": "(((temp - 32) * 5.0) / 9.0) + 273.15",
      "attribute": "tempK",
      "next": "myDatastoreActivity"
    }
  }
]
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Analytics APIリファレンスの[UpdatePipeline](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdatePipeline](#)」の「」を参照してください。

を使用した Device Advisor の例 AWS CLI

次のコード例は、Device Advisor AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-suite-definition

次の例は、create-suite-definition を使用する方法を説明しています。

AWS CLI

例 1: IoT Device Advisor テストスイートを作成するには

次のcreate-suite-definition例では、指定されたスイート定義設定を使用して AWS IoT にデバイスアドバイザーテストスイートを作成します。

```
aws iotdeviceadvisor create-suite-definition \
  --suite-definition-configuration '{ \
    "suiteDefinitionName": "TestSuiteName", \
    "devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing"}], \
    "intendedForQualification": false, \
    "rootGroup": "{\"configuration\": {}, \"tests\": [{\"name\": \"MQTT Connect\", \
  \"configuration\": {\"EXECUTION_TIMEOUT\": 120}, \"tests\": [{\"name\": \"MQTT_Connect\", \
  \"configuration\": {}, \"test\": {\"id\": \"MQTT_Connect\", \"testCase\": null, \"version \
  \": \"0.0.0\"}}]}]}", \
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}
```

出力:

```
{
  "suiteDefinitionId": "0jtsgio7yenu",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/0jtsgio7yenu",
  "suiteDefinitionName": "TestSuiteName",
  "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

詳細については、[「IoT Core デベロッパーガイド」の「テストスイート定義AWS の作成 IoT」](#)を参照してください。

例 2: IoT Device Advisor の最新の認定テストスイートを作成するには

次のcreate-suite-definition例では、指定されたスイート定義設定で AWS IoT の最新バージョンを持つデバイスアドバイザー認定テストスイートを作成します。

```
aws iotdeviceadvisor create-suite-definition \
  --suite-definition-configuration '{ \
    "suiteDefinitionName": "TestSuiteName", \
    "devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing"}], \
    "intendedForQualification": true, \
    "rootGroup": "", \
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}
```

出力:

```
{
  "suiteDefinitionId": "txgsuolk2myj",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/txgsuolk2myj",
  "suiteDefinitionName": "TestSuiteName",
  "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

詳細については、[「IoT Core デベロッパーガイド」の「テストスイート定義AWS の作成 IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSuiteDefinition](#)」の「」を参照してください。

delete-suite-definition

次のコード例は、delete-suite-definition を使用する方法を示しています。

AWS CLI

IoT Device Advisor テストスイートを削除するには

次のdelete-suite-definition例では、指定されたスイート定義 ID を持つデバイスアドバイザーテストスイートを削除します。

```
aws iotdeviceadvisor delete-suite-definition \
```



```
--suite-definition-id 0jtsgio7yenu
```

このコマンドでは何も出力されません。

詳細については、AWS IoT APIリファレンスの[DeleteSuiteDefinition](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteSuiteDefinition](#)」の「」を参照してください。

get-endpoint

次のコード例は、get-endpoint を使用する方法を示しています。

AWS CLI

例 1: IoT Device Advisor アカウントレベルのエンドポイントに関する情報を取得するには

次のget-endpoint例では、デバイスアドバイザーのアカウントレベルのテストエンドポイントに関する情報を取得します。

```
aws iotdeviceadvisor get-endpoint
```

出力:

```
{
  "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

例 2: IoT Device Advisor デバイスレベルのエンドポイントに関する情報を取得するには

次のget-endpoint例では、指定された thing-arn または certificate-arn を持つデバイスアドバイザーのデバイスレベルのテストエンドポイントに関する情報を取得します。

```
aws iotdeviceadvisor get-endpoint \
  --thing-arn arn:aws:iot:us-east-1:123456789012:thing/MyIotThing
```

出力:

```
{
  "endpoint": "tdb7719be5t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

詳細については、AWS IoT Core デベロッパーガイド」の「[テストエンドポイントの取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEndpoint](#)」の「」を参照してください。

get-suite-definition

次の例は、get-suite-definition を使用する方法を説明しています。

AWS CLI

IoT Device Advisor テストスイートに関する情報を取得するには

次のget-suite-definition例では、指定されたスイート定義 ID を持つ Aevice アドバイザー テストスイートに関する情報を取得します。

```
aws iotdeviceadvisor get-suite-definition \  
  --suite-definition-id qqcsmtyyjabl
```

出力:

```
{  
  "suiteDefinitionId": "qqcsmtyyjabl",  
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-  
east-1:123456789012:suitedefinition/qqcsmtyyjabl",  
  "suiteDefinitionVersion": "v1",  
  "latestVersion": "v1",  
  "suiteDefinitionConfiguration": {  
    "suiteDefinitionName": "MQTT connection",  
    "devices": [],  
    "intendedForQualification": false,  
    "isLongDurationTest": false,  
    "rootGroup": "{\"configuration\":{},\"tests\": [{\"id\": \"uta5d9j1kvwc\",  
  \"name\": \"Test group 1\", \"configuration\": {}, \"tests\": [{\"id\": \"awr8pq5vc9yp\",  
  \"name\": \"MQTT Connect\", \"configuration\": {}, \"test\": {\"id\": \"MQTT_Connect\",  
  \"testCase\": null, \"version\": \"0.0.0\"}]}]}\"",  
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole",  
    "protocol": "MqttV3_1_1"  
  },  
  "createdAt": "2022-11-11T22:28:52.389000-05:00",  
  "lastModifiedAt": "2022-11-11T22:28:52.389000-05:00",  
}
```

```
"tags": {}  
}
```

詳細については、[「IoT Core デベロッパーガイド」の「テストスイート定義の取得」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSuiteDefinition](#)」の「」を参照してください。

get-suite-run-report

次の例は、get-suite-run-report を使用する方法を説明しています。

AWS CLI

IoT Device Advisor 認定テストスイート実行レポートに関する情報を取得するには

次のget-suite-run-report例では、指定されたスイート定義 ID とスイート実行 ID で正常に実行されたデバイスアドバイザー認定テストスイートに関するレポートダウンロードリンクを取得します。

```
aws iotdeviceadvisor get-suite-run-report \  
  --suite-definition-id ztvb5aek4w4x \  
  --suite-run-id p6awv83nre6v
```

出力:

```
{  
  "qualificationReportDownloadUrl": "https://senate-apn-reports-us-east-1-  
prod.s3.amazonaws.com/report.downloadlink"  
}
```

詳細については、AWS 「IoT Core デベロッパーガイド」の[「認定テストスイートが正常に実行されるための認定レポートの取得」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSuiteRunReport](#)」の「」を参照してください。

get-suite-run

次のコード例は、get-suite-run を使用する方法を示しています。

AWS CLI

IoT Device Advisor テストスイートの実行ステータスに関する情報を取得するには

次の`get-suite-run`例では、指定されたスイート定義 ID とスイート実行 ID を持つデバイスアドバイザーテストスイート実行ステータスに関する情報を取得します。

```
aws iotdeviceadvisor get-suite-run \  
  --suite-definition-id qqcsmtyyjabl \  
  --suite-run-id nzlfyhaa18oa
```

出力:

```
{  
  "suiteDefinitionId": "qqcsmtyyjabl",  
  "suiteDefinitionVersion": "v1",  
  "suiteRunId": "nzlfyhaa18oa",  
  "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/  
qqcsmtyyjabl/nzlfyhaa18oa",  
  "suiteRunConfiguration": {  
    "primaryDevice": {  
      "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing",  
      "certificateArn": "arn:aws:iot:us-east-1:123456789012:cert/certFile"  
    },  
    "parallelRun": false  
  },  
  "testResult": {  
    "groups": [  
      {  
        "groupId": "uta5d9j1kvwc",  
        "groupName": "Test group 1",  
        "tests": [  
          {  
            "testCaseRunId": "2ve2twrqyr0s",  
            "testCaseDefinitionId": "awr8pq5vc9yp",  
            "testCaseDefinitionName": "MQTT Connect",  
            "status": "PASS",  
            "startTime": "2022-11-12T00:01:53.693000-05:00",  
            "endTime": "2022-11-12T00:02:15.443000-05:00",  
            "logUrl": "https://console.aws.amazon.com/  
cloudwatch/home?region=us-east-1#logEventViewer:group=/aws/iot/deviceadvisor/  
qqcsmtyyjabl;stream=nzlfyhaa18oa_2ve2twrqyr0s",  
            "warnings": "null",  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
        "failure": "null"
      }
    ]
  },
  "startTime": "2022-11-12T00:01:52.673000-05:00",
  "endTime": "2022-11-12T00:02:16.496000-05:00",
  "status": "PASS",
  "tags": {}
}
```

詳細については、「IoT Core デベロッパーガイド」の「[テストスイートの実行を取得する](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSuiteRun](#)」の「」を参照してください。

list-suite-definitions

次のコード例は、list-suite-definitions を使用する方法を示しています。

AWS CLI

例 1: 作成した IoT Device Advisor テストスイートを一覧表示するには

次のlist-suite-definitions例では、AWS IoT で作成した最大 25 のデバイスアドバイザーテストスイートを一覧表示します。テストスイートが 25 個を超える場合は、出力に nextToken 「」が表示されます。このnextToken 「」を使用して、作成した残りのテストスイートを表示できます。

```
aws iotdeviceadvisor list-suite-definitions
```

出力:

```
{
  "suiteDefinitionInformationList": [
    {
      "suiteDefinitionId": "3hsn88h4p2g5",
      "suiteDefinitionName": "TestSuite1",
      "defaultDevices": [
        {
```

```

        "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/
MyIotThing"
        }
    ],
    "intendedForQualification": false,
    "isLongDurationTest": false,
    "protocol": "MqttV3_1_1",
    "createdAt": "2022-11-17T14:15:56.830000-05:00"
},
{
    .....
}
],
"nextToken": "nextTokenValue"
}

```

例 2: 指定した設定で作成した IoT Device Advisor テストスイートを一覧表示するには

次の `list-suite-definitions` 例では、IoT で作成したデバイスアドバイザーテストスイートを、指定された最大結果数 AWS IoT で一覧表示します。最大数よりも多くのテストスイートがある場合、出力に `nextToken` 「」が表示されます。`nextToken` 「」がある場合は、`nextToken` 「」を使用して、以前に表示されなかったテストスイートを表示できます。

```

aws iotdeviceadvisor list-suite-definitions \
  --max-result 1 \
  --next-token "nextTokenValue"

```

出力:

```

{
  "suiteDefinitionInformationList": [
    {
      "suiteDefinitionId": "ztlv5aew4w4x",
      "suiteDefinitionName": "TestSuite2",
      "defaultDevices": [],
      "intendedForQualification": true,
      "isLongDurationTest": false,
      "protocol": "MqttV3_1_1",
      "createdAt": "2022-11-17T14:15:56.830000-05:00"
    }
  ],
  "nextToken": "nextTokenValue"
}

```

```
}
```

詳細については、AWS IoT APIリファレンスの[ListSuiteDefinitions](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListSuiteDefinitions](#)」の「」を参照してください。

list-suite-runs

次の例は、list-suite-runs を使用する方法を説明しています。

AWS CLI

例 1: 指定された IoT Device Advisor テストスイートの実行ステータスに関するすべての情報を一覧表示するには

次のlist-suite-runs例では、指定されたスイート定義 ID を持つデバイスアドバイザーテストスイートの実行ステータスに関するすべての情報を一覧表示します。25 回以上のテストスイートの実行がある場合、出力にnextToken「」が表示されます。このnextToken「」を使用して、残りのテストスイートの実行を表示できます。

```
aws iotdeviceadvisor list-suite-runs \  
  --suite-definition-id ztvb5aew4w4x
```

出力:

```
{  
  "suiteRunsList": [  
    {  
      "suiteDefinitionId": "ztvb5aew4w4x",  
      "suiteDefinitionVersion": "v1",  
      "suiteDefinitionName": "TestSuite",  
      "suiteRunId": "p6awv89nre6v",  
      "createdAt": "2022-12-01T16:33:14.212000-05:00",  
      "startedAt": "2022-12-01T16:33:15.710000-05:00",  
      "endAt": "2022-12-01T16:42:03.323000-05:00",  
      "status": "PASS",  
      "passed": 6,  
      "failed": 0  
    }  
  ]  
}
```

例 2: 指定された IoT Device Advisor テストスイートの実行ステータスに関する情報を指定された設定で一覧表示するには

次の `list-suite-runs` 例は、指定されたスイート定義 ID と指定された最大結果数を持つデバイスアドバイザーテストスイートの実行ステータスに関する情報を示しています。最大数よりも多くのテストスイート実行がある場合、出力に `nextToken` 「」が表示されます。`nextToken` 「」がある場合は、`nextToken` 「」を使用して、以前に表示されなかったテストスイートの実行を表示できます。

```
aws iotdeviceadvisor list-suite-runs \  
  --suite-definition-id qqcsmtyyjaml \  
  --max-result 1 \  
  --next-token "nextTokenValue"
```

出力:

```
{  
  "suiteRunsList": [  
    {  
      "suiteDefinitionId": "qqcsmtyyjaml",  
      "suiteDefinitionVersion": "v1",  
      "suiteDefinitionName": "MQTT connection",  
      "suiteRunId": "gz9vm2s6d2jy",  
      "createdAt": "2022-12-01T20:10:27.079000-05:00",  
      "startedAt": "2022-12-01T20:10:28.003000-05:00",  
      "endAt": "2022-12-01T20:10:45.084000-05:00",  
      "status": "STOPPED",  
      "passed": 0,  
      "failed": 0  
    }  
  ],  
  "nextToken": "nextTokenValue"  
}
```

詳細については、AWS IoT API リファレンスの [ListSuiteRuns](#) 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListSuiteRuns](#)」の「」を参照してください。

list-tags-for-resource

次の例は、`list-tags-for-resource` を使用する方法を説明しています。

AWS CLI

IoT Device Advisor リソースにアタッチされたタグを一覧表示するには

次のlist-tags-for-resource例では、デバイスアドバイザーリソースにアタッチされたタグを一覧表示します。デバイスアドバイザーリソースは、Suitedefinition-Arn または Suiterun-Arn にすることができます。

```
aws iotdeviceadvisor list-tags-for-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny
```

出力:

```
{  
  "tags": {  
    "TestTagKey": "TestTagValue"  
  }  
}
```

詳細については、「サービス認証リファレンス[ListTagsForResource](#)」の「IoT [Core Device Advisor](#) で定義される [AWS IoT リファレンスとリソースタイプ](#)」の「」を参照してください。

AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

start-suite-run

次のコード例は、start-suite-run を使用する方法を示しています。

AWS CLI

IoT Device Advisor テストスイートの実行を開始するには

次のstart-suite-run例では、AWS アカウントで使用可能なウィジェットを一覧表示します。

```
aws iotdeviceadvisor start-suite-run \  
  --suite-definition-id qqcsmtyyjabl \  
  --suite-definition-version v1 \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny
```

```
--suite-run-configuration '{"primaryDevice":{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing"}, "certificateArn": "arn:aws:iot:us-east-1:123456789012:cert/certFile"}'
```

出力:

```
{
  "suiteRunId": "pwmucgw7lt9s",
  "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/qqcsmtyyjabl/pwmucgw7lk9s",
  "createdAt": "2022-12-02T15:43:05.581000-05:00"
}
```

詳細については、[「IoT Core デベロッパーガイド」の「テストスイートの実行を開始するAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartSuiteRun](#)」の「」を参照してください。

stop-suite-run

次のコード例は、stop-suite-run を使用する方法を示しています。

AWS CLI

現在実行中の IoT Device Advisor テストスイートを停止するには

次のstop-suite-run例では、指定されたスイート定義 ID とスイート実行 ID で現在実行されているデバイスアドバイザーテストスイートを停止します。

```
aws iotdeviceadvisor stop-suite-run \
  --suite-definition-id qqcsmtyyjabl \
  --suite-run-id nzlfyhaa18oa
```

このコマンドでは何も出力されません。

詳細については、[「IoT Core デベロッパーガイド」の「テストスイートの実行を停止するAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopSuiteRun](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

IoT Device Advisor リソースの既存のタグに追加および変更するには

次のtag-resource例では、指定されたリソース ARN とタグを使用して、デバイスアドバイザーリソースの既存のタグを に追加して変更します。デバイスアドバイザーリソースは、Suitedefinition-Arn または Suiterun-Arn にすることができます。

```
aws iotdeviceadvisor tag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tags '{"TagKey": "TagValue"}
```

このコマンドでは何も出力されません。

詳細については、「サービス認証リファレンス [TagResource](#)」の「IoT [Core Device Advisor](#) で定義される [AWS IoT リファレンスとリソースタイプ](#)」の「」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

IoT Device Advisor リソースから既存のタグを削除するには

次のuntag-resource例では、指定されたリソース ARN とタグキーを持つデバイスアドバイザーリソースから既存のタグを削除します。デバイスアドバイザーリソースは、Suitedefinition-Arn または Suiterun-Arn にすることができます。

```
aws iotdeviceadvisor untag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tag-keys "TagKey"
```

このコマンドでは何も出力されません。

詳細については、「サービス認証リファレンス [UntagResource](#)」の「IoT [Core Device Advisor](#) で定義される [AWS IoT リファレンスとリソースタイプ](#)」の「」を参照してください。AWS IoT API

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-suite-definition

次の例は、update-suite-definition を使用する方法を説明しています。

AWS CLI

例 1: IoT Device Advisor テストスイートを更新するには

次のupdate-suite-definition例では、指定されたスイート定義 ID とスイート定義設定を使用して AWS IoT のデバイスアドバイザーテストスイートを更新します。

```
aws iotdeviceadvisor update-suite-definition \
  --suite-definition-id 3hsn88h4p2g5 \
  --suite-definition-configuration '{ \
    "suiteDefinitionName": "TestSuiteName", \
    "devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyIotThing"}], \
    "intendedForQualification": false, \
    "rootGroup": "{\"configuration\": {}, \"tests\": [{\"name\": \"MQTT Connect\", \
  \"configuration\": {\"EXECUTION_TIMEOUT\": 120}, \"tests\": [{\"name\": \"MQTT_Connect\", \
  \"configuration\": {}, \"test\": {\"id\": \"MQTT_Connect\", \"testCase\": null, \"version \
  \": \"0.0.0\"}]}]}\", \
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}
```

出力:

```
{
  "suiteDefinitionId": "3hsn88h4p2g5",
  "suiteDefinitionName": "TestSuiteName",
  "suiteDefinitionVersion": "v3",
  "createdAt": "2022-11-17T14:15:56.830000-05:00",
  "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

例 2: IoT Device Advisor 認定テストスイートを更新するには

次のupdate-suite-definition例では、指定されたスイート定義 ID とスイート定義設定を使用して AWS IoT のデバイスアドバイザー認定テストスイートを更新します。

```
aws iotdeviceadvisor update-suite-definition \  
  --suite-definition-id txgsuolk2myj \  
  --suite-definition-configuration '{  
    "suiteDefinitionName": "TestSuiteName", \  
    "devices": [{"thingArn": "arn:aws:iot:us-east-1:123456789012:thing/  
MyIotThing"}], \  
    "intendedForQualification": true, \  
    "rootGroup": "", \  
    "devicePermissionRoleArn": "arn:aws:iam::123456789012:role/Myrole"}'
```

出力:

```
{  
  "suiteDefinitionId": "txgsuolk2myj",  
  "suiteDefinitionName": "TestSuiteName",  
  "suiteDefinitionVersion": "v3",  
  "createdAt": "2022-11-17T14:15:56.830000-05:00",  
  "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"  
}
```

詳細については、AWS IoT APIリファレンスの[UpdateSuiteDefinition](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSuiteDefinition](#)」の「」を参照してください。

AWS IoT data を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT data。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-thing-shadow

次の例は、delete-thing-shadow を使用する方法を説明しています。

AWS CLI

デバイスのシャドウドキュメントを削除するには

次のdelete-thing-shadow例では、 という名前のデバイスのシャドウドキュメント全体を削除しますMyRPI。

```
aws iot-data delete-thing-shadow \  
  --thing-name MyRPI \  
  "output.txt"
```

コマンドはディスプレイに出力を生成しませんが、削除したシャドウドキュメントのバージョンとタイムスタンプを確認する情報output.txtが含まれています。

```
{"version":2,"timestamp":1560270384}
```

詳細については、IoT デベロッパーガイドの「[シャドウの使用](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteThingShadow](#)」の「」を参照してください。

get-thing-shadow

次の例は、get-thing-shadow を使用する方法を説明しています。

AWS CLI

モノのシャドウドキュメントを取得するには

次のget-thing-shadow例では、指定された IoT モノのモノシャドウドキュメントを取得します。

```
aws iot-data get-thing-shadow \  
  --thing-name MyRPi \  
  output.txt
```

コマンドはディスプレイに出力を生成しませんが、次は の内容を示していますoutput.txt。

```
{  
  "state":{  
    "reported":{  
      "moisture":"low"  
    }  
  },  
  "metadata":{  
    "reported":{  
      "moisture":{  
        "timestamp":1560269319  
      }  
    }  
  },  
  "version":1,"timestamp":1560269405  
}
```

詳細については、IoT デベロッパーガイドの「[Device Shadow Service Data Flow](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[GetThingShadow](#)」の「」を参照してください。

update-thing-shadow

次の例は、update-thing-shadow を使用する方法を説明しています。

AWS CLI

モノのシャドウを更新するには

次のupdate-thing-shadow例では、指定されたモノのデバイスシャドウの現在の状態を変更し、それを ファイル に保存しますoutput.txt。

```
aws iot-data update-thing-shadow \  
  --thing-name MyRPi \  
  --payload '{"state":{"reported":{"moisture":"okay"}}}' \  
  output.txt
```

"output.txt"

コマンドはディスプレイに出力を生成しませんが、次は の内容を示していますoutput.txt。

```
{
  "state": {
    "reported": {
      "moisture": "okay"
    }
  },
  "metadata": {
    "reported": {
      "moisture": {
        "timestamp": 1560270036
      }
    }
  },
  "version": 2,
  "timestamp": 1560270036
}
```

詳細については、IoT デベロッパーガイドの「[Device Shadow Service Data Flow](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateThingShadow](#)」の「」を参照してください。

AWS IoT Events を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Events。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-put-message

次のコード例は、batch-put-message を使用する方法を示しています。

AWS CLI

AWS IoT Events にメッセージ (入力) を送信するには

次のbatch-put-message例では、一連のメッセージを AWS IoT Events システムに送信します。各メッセージペイロードは、指定した入力 () inputName に変換され、その入力を監視するディテクターに取り込まれます。複数のメッセージが送信された場合、メッセージが処理される順序は保証されません。注文を保証するには、メッセージを一度に 1 つずつ送信し、正常なレスポンスを待つ必要があります。

```
aws iotevents-data batch-put-message \  
  --cli-input-json file://highPressureMessage.json
```

highPressureMessage.json の内容:

```
{  
  "messages": [  
    {  
      "messageId": "00001",  
      "inputName": "PressureInput",  
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":  
80, \"temperature\": 39} }"  
    }  
  ]  
}
```

出力:

```
{  
  "BatchPutMessageErrorEntries": []  
}
```

詳細については、AWS IoT Events APIリファレンスの[BatchPutMessage](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchPutMessage](#)」の「」を参照してください。

batch-update-detector

次のコード例は、batch-update-detector を使用する方法を示しています。

AWS CLI

ディテクターを更新するには (インスタンス)

次のbatch-update-detector例では、指定されたディテクターモデル内の 1 つ以上のディテクター (インスタンス) の状態、変数値、タイマー設定を更新します。

```
aws iotevents-data batch-update-detector \  
  --cli-input-json file://budFulton-A32.json
```

budFulton-A32.json の内容:

```
{  
  "detectors": [  
    {  
      "messageId": "00001",  
      "detectorModelName": "motorDetectorModel",  
      "keyValue": "Fulton-A32",  
      "state": {  
        "stateName": "Normal",  
        "variables": [  
          {  
            "name": "pressureThresholdBreach",  
            "value": "0"  
          }  
        ],  
        "timers": [  
        ]  
      }  
    }  
  ]  
}
```

出力:

```
{
  "batchUpdateDetectorErrorEntries": []
}
```

詳細については、AWS 「IoT Events API Reference [BatchUpdateDetector](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchUpdateDetector](#)」の「」を参照してください。

create-detector-model

次のコード例は、create-detector-model を使用する方法を示しています。

AWS CLI

ディテクターモデルを作成するには

次のcreate-detector-model例では、パラメータファイルで指定された設定でディテクターモデルを作成します。

```
aws iotevents create-detector-model \
  --cli-input-json file://motorDetectorModel.json
```

motorDetectorModel.json の内容:

```
{
  "detectorModelName": "motorDetectorModel",
  "detectorModelDefinition": {
    "states": [
      {
        "stateName": "Normal",
        "onEnter": {
          "events": [
            {
              "eventName": "init",
              "condition": "true",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreach",
                    "value": "0"
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

```

    }
  }
]
},
"onInput": {
  "transitionEvents": [
    {
      "eventName": "Overpressurized",
      "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
      "actions": [
        {
          "setVariable": {
            "variableName": "pressureThresholdBreach",
            "value":
"$variable.pressureThresholdBreach + 3"
          }
        }
      ],
      "nextState": "Dangerous"
    }
  ]
},
{
  "stateName": "Dangerous",
  "onEnter": {
    "events": [
      {
        "eventName": "Pressure Threshold Breached",
        "condition": "$variable.pressureThresholdBreach &gt;
1",
        "actions": [
          {
            "sns": {
              "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
            }
          }
        ]
      }
    ]
  }
}
]

```

```

    },
    "onInput": {
      "events": [
        {
          "eventName": "Overpressurized",
          "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
          "actions": [
            {
              "setVariable": {
                "variableName": "pressureThresholdBreached",
                "value": "3"
              }
            }
          ]
        },
        {
          "eventName": "Pressure Okay",
          "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70",
          "actions": [
            {
              "setVariable": {
                "variableName": "pressureThresholdBreached",
                "value":
"$variable.pressureThresholdBreached - 1"
              }
            }
          ]
        }
      ],
      "transitionEvents": [
        {
          "eventName": "BackToNormal",
          "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70 &amp; $variable.pressureThresholdBreached &lt;= 1",
          "nextState": "Normal"
        }
      ]
    },
    "onExit": {
      "events": [
        {
          "eventName": "Normal Pressure Restored",

```

```

        "condition": "true",
        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                }
            }
        ]
    }
]
},
"initialStateName": "Normal"
},
"key": "motorid",
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

出力:

```

{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560796816.077,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560796816.077,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
}

```

詳細については、AWS 「IoT Events API Reference [CreateDetectorModel](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDetectorModel](#)」の「」を参照してください。

create-input

次の例は、create-input を使用する方法を説明しています。

AWS CLI

入力を作成するには

次のcreate-input例では、入力を作成します。

```
aws iotevents create-input \  
--cli-input-json file://pressureInput.json
```

pressureInput.json の内容:

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

出力:

```
{  
  "inputConfiguration": {  
    "status": "ACTIVE",  
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
    "lastUpdateTime": 1560795312.542,  
    "creationTime": 1560795312.542,  
    "inputName": "PressureInput",  
    "inputDescription": "Pressure readings from a motor"  
  }  
}
```

詳細については、AWS 「IoT イベントAPIリファレンス[CreateInput](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInput](#)」の「」を参照してください。

delete-detector-model

次のコード例は、delete-detector-model を使用する方法を示しています。

AWS CLI

ディテクターモデルを削除するには

次のdelete-detector-model例では、指定されたディテクターモデルを削除します。ディテクターモデルのアクティブなインスタンスも削除されます。

```
aws iotevents delete-detector-model \  
  --detector-model-name motorDetectorModel
```

このコマンドでは何も出力されません。

詳細については、AWS 「IoT Events API Reference [DeleteDetectorModel](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDetectorModel](#)」の「」を参照してください。

delete-input

次の例は、delete-input を使用する方法を説明しています。

AWS CLI

入力を削除するには

次のdelete-input例では、指定された入力を削除します。

```
aws iotevents delete-input \  
  --input-name PressureInput
```

このコマンドでは何も出力されません。

詳細については、AWS 「IoT イベントAPIリファレンス[DeleteInput](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteInput](#)」の「」を参照してください。

describe-detector-model

次の例は、describe-detector-model を使用する方法を説明しています。

AWS CLI

ディテクターモデルに関する情報を取得するには

次のdescribe-detector-model例では、指定されたディテクターモデルの詳細を表示します。version パラメータが指定されていないため、最新バージョンに関する情報が返されます。

```
aws iotevents describe-detector-model \  
  --detector-model-name motorDetectorModel
```

出力:

```
{  
  "detectorModel": {  
    "detectorModelConfiguration": {  
      "status": "ACTIVE",  
      "lastUpdateTime": 1560796816.077,  
      "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
      "creationTime": 1560796816.077,  
      "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/motorDetectorModel",  
      "key": "motorid",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    },  
    "detectorModelDefinition": {  
      "states": [  
        {  
          "onInput": {  
            "transitionEvents": [  
              {  
                "eventName": "Overpressurized",  
                "actions": [  
                  {  
                    "setVariable": {
```

```

        "variableName":
"pressureThresholdBreached",
        "value":
"$variable.pressureThresholdBreached + 3"
    }
    ],
    "condition":
"$input.PressureInput.sensorData.pressure > 70",
    "nextState": "Dangerous"
    }
    ],
    "events": []
},
"stateName": "Normal",
"onEnter": {
    "events": [
        {
            "eventName": "init",
            "actions": [
                {
                    "setVariable": {
                        "variableName":
"pressureThresholdBreached",
                        "value": "0"
                    }
                }
            ],
            "condition": "true"
        }
    ]
},
"onExit": {
    "events": []
}
},
{
    "onInput": {
        "transitionEvents": [
            {
                "eventName": "BackToNormal",
                "actions": [],

```

```

        "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
        "nextState": "Normal"
    }
],
"events": [
    {
        "eventName": "Overpressurized",
        "actions": [
            {
                "setVariable": {
                    "variableName":
"pressureThresholdBreached",
                    "value": "3"
                }
            }
        ],
        "condition":
"$input.PressureInput.sensorData.pressure > 70"
    },
    {
        "eventName": "Pressure Okay",
        "actions": [
            {
                "setVariable": {
                    "variableName":
"pressureThresholdBreached",
                    "value":
"$variable.pressureThresholdBreached - 1"
                }
            }
        ],
        "condition":
"$input.PressureInput.sensorData.pressure <= 70"
    }
],
"stateName": "Dangerous",
"onEnter": {
    "events": [
        {
            "eventName": "Pressure Threshold Breached",
            "actions": [

```

```

        {
            "sns": {
                "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
            }
        },
        "condition": "$variable.pressureThresholdBreached >
1"
    }
],
"onExit": {
    "events": [
        {
            "eventName": "Normal Pressure Restored",
            "actions": [
                {
                    "sns": {
                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                    }
                }
            ],
            "condition": "true"
        }
    ]
}
],
"initialStateName": "Normal"
}
}
}

```

詳細については、AWS 「IoT イベントAPIリファレンス[DescribeDetectorModel](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDetectorModel](#)」の「」を参照してください。

describe-detector

次のコード例は、describe-detector を使用する方法を示しています。

AWS CLI

ディテクター (インスタンス) に関する情報を取得するには。

次のdescribe-detector例では、指定されたディテクター (インスタンス) の詳細を表示します。

```
aws iotevents-data describe-detector \  
  --detector-model-name motorDetectorModel \  
  --key-value "Fulton-A32"
```

出力:

```
{  
  "detector": {  
    "lastUpdateTime": 1560797852.776,  
    "creationTime": 1560797852.775,  
    "state": {  
      "variables": [  
        {  
          "name": "pressureThresholdBreached",  
          "value": "3"  
        }  
      ],  
      "stateName": "Dangerous",  
      "timers": []  
    },  
    "keyValue": "Fulton-A32",  
    "detectorModelName": "motorDetectorModel",  
    "detectorModelVersion": "1"  
  }  
}
```

詳細については、AWS 「IoT イベントAPIリファレンス[DescribeDetector](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDetector](#)」の「」を参照してください。

describe-input

次の例は、describe-input を使用する方法を説明しています。

AWS CLI

入力に関する情報を取得するには

次のdescribe-input例では、指定された入力の詳細を表示します。

```
aws iotevents describe-input \  
  --input-name PressureInput
```

出力:

```
{  
  "input": {  
    "inputConfiguration": {  
      "status": "ACTIVE",  
      "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/  
PressureInput",  
      "lastUpdateTime": 1560795312.542,  
      "creationTime": 1560795312.542,  
      "inputName": "PressureInput",  
      "inputDescription": "Pressure readings from a motor"  
    },  
    "inputDefinition": {  
      "attributes": [  
        {  
          "jsonPath": "sensorData.pressure"  
        },  
        {  
          "jsonPath": "motorid"  
        }  
      ]  
    }  
  }  
}
```

詳細については、AWS IoT Events APIリファレンスの[DescribeInput](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeInput](#)」の「」を参照してください。

describe-logging-options

次の例は、describe-logging-options を使用する方法を説明しています。

AWS CLI

ログ記録設定に関する情報を取得するには

次のdescribe-logging-options例では、AWS IoT Events ログ記録オプションの現在の設定を取得します。

```
aws iotevents describe-logging-options
```

出力:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "enabled": false,
    "level": "ERROR"
  }
}
```

詳細については、AWS 「IoT Events API Reference [DescribeLoggingOptions](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoggingOptions](#)」の「」を参照してください。

list-detector-model-versions

次の例は、list-detector-model-versions を使用する方法を説明しています。

AWS CLI

ディテクターモデルのバージョンに関する情報を取得するには

次のlist-detector-model-versions例では、ディテクターモデルのすべてのバージョンを一覧表示します。各探知器モデルのバージョンに関連付けられているメタデータのみが返されます。

```
aws iotevents list-detector-model-versions \
```

```
--detector-model-name motorDetectorModel
```

出力:

```
{
  "detectorModelVersionSummaries": [
    {
      "status": "ACTIVE",
      "lastUpdateTime": 1560796816.077,
      "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
      "creationTime": 1560796816.077,
      "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
      "detectorModelName": "motorDetectorModel",
      "detectorModelVersion": "1"
    }
  ]
}
```

詳細については、AWS IoT Events APIリファレンスの[ListDetectorModelVersions](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListDetectorModelVersions](#)」の「」を参照してください。

list-detector-models

次の例は、list-detector-models を使用する方法を説明しています。

AWS CLI

ディテクターモデルのリストを取得するには

次のlist-detector-models例では、作成したディテクターモデルを一覧表示します。各探知器モデルに関連付けられているメタデータのみが返されます。

```
aws iotevents list-detector-models
```

出力:

```
{
  "detectorModelSummaries": [
```



```
{
  "detectorModelName": "motorDetectorModel",
  "creationTime": 1552072424.212
  "detectorModelDescription": "Detect overpressure in a motor."
}
]
```

詳細については、AWS「IoT イベントAPIリファレンス[ListDetectorModels](#)」の「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListDetectorModels](#)」の「」を参照してください。

list-detectors

次のコード例は、list-detectors を使用する方法を示しています。

AWS CLI

ディテクターモデルのディテクターのリストを取得するには

次のlist-detectors例では、アカウントのディテクター (ディテクターモデルのインスタンス) を一覧表示します。

```
aws iotevents-data list-detectors \
  --detector-model-name motorDetectorModel
```

出力:

```
{
  "detectorSummaries": [
    {
      "lastUpdateTime": 1558129925.2,
      "creationTime": 1552073155.527,
      "state": {
        "stateName": "Normal"
      },
      "keyValue": "Fulton-A32",
      "detectorModelName": "motorDetectorModel",
      "detectorModelVersion": "1"
    }
  ]
}
```

```
]
}
```

詳細については、AWS 「IoT イベントAPIリファレンス[ListDetectors](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDetectors](#)」の「」を参照してください。

list-inputs

次の例は、list-inputs を使用する方法を説明しています。

AWS CLI

入力を一覧表示するには

次のlist-inputs例では、アカウントで作成した入力を一覧表示します。

```
aws iotevents list-inputs
```

このコマンドでは何も出力されません。出力:

```
{
  {
    "status": "ACTIVE",
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "lastUpdateTime": 1551742986.768,
    "creationTime": 1551742986.768,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
  }
}
```

詳細については、AWS IoT Events APIリファレンスの[ListInputs](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInputs](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースに割り当てられたタグを一覧表示します。

次のlist-tags-for-resource例では、リソースに割り当てたタグキーの名前と値を一覧表示します。

```
aws iotevents list-tags-for-resource \  
  --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

出力:

```
{  
  "tags": [  
    {  
      "value": "motor",  
      "key": "deviceType"  
    }  
  ]  
}
```

詳細については、AWS IoT Events APIリファレンスの[ListTagsForResource](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-logging-options

次のコード例は、put-logging-options を使用する方法を示しています。

AWS CLI

ログ記録オプションを設定するには

次のput-logging-options例では、AWS IoT Events のログ記録オプションを設定または更新します。loggingOptions` field, it can take up to one minute for the change to take effect. Also, if you change the policy attached to the role you specified in the ``roleArn フィールドの値を更新すると (無効なポリシーを修正するなど)、その変更が有効になるまでに最大 5 分かかることがあります。

```
aws iotevents put-logging-options \  
  --cli-input-json file://logging-options.json
```

logging-options.json の内容:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "level": "DEBUG",  
    "enabled": true,  
    "detectorDebugOptions": [  
      {  
        "detectorModelName": "motorDetectorModel",  
        "keyValue": "Fulton-A32"  
      }  
    ]  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Events API リファレンスの [PutLoggingOptions](#) 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutLoggingOptions](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを追加するには

次のtag-resource例では、指定されたリソースがアタッチされたタグを追加または変更します (キーdeviceTypeがすでに存在する場合)。

```
aws iotevents tag-resource \  
  --cli-input-json file://pressureInput.tag.json
```

`pressureInput.tag.json` の内容:

```
{
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
  "tags": [
    {
      "key": "deviceType",
      "value": "motor"
    }
  ]
}
```

このコマンドでは何も出力されません。

詳細については、AWS 「IoT イベントAPIリファレンス[TagResource](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、`untag-resource` を使用する方法を示しています。

AWS CLI

リソースからタグを削除する

次の`untag-resource`例では、指定されたリソースから指定されたキー名のタグを削除します。

```
aws iotevents untag-resource \
  --resource-arn arn:aws:iotevents:us-west-2:123456789012:input/PressureInput \
  --tagkeys deviceType
```

このコマンドでは何も出力されません。

詳細については、AWS 「IoT イベントAPIリファレンス[UntagResource](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-detector-model

次のコード例は、update-detector-model を使用する方法を示しています。

AWS CLI

ディテクターモデルを更新するには

次のupdate-detector-model例では、指定されたディテクターモデルを更新します。以前のバージョンによって生成されたディテクター (インスタンス) が削除され、新しい入力が入力されると再作成されます。

```
aws iotevents update-detector-model \  
  --cli-input-json file://motorDetectorModel.update.json
```

motorDetectorModel.update.json の内容:

```
{  
  "detectorModelName": "motorDetectorModel",  
  "detectorModelDefinition": {  
    "states": [  
      {  
        "stateName": "Normal",  
        "onEnter": {  
          "events": [  
            {  
              "eventName": "init",  
              "condition": "true",  
              "actions": [  
                {  
                  "setVariable": {  
                    "variableName": "pressureThresholdBreach",  
                    "value": "0"  
                  }  
                }  
              ]  
            }  
          ]  
        },  
        "onInput": {  
          "transitionEvents": [  
            {  
              "eventName": "Overpressurized",
```

```

    "condition": "$input.PressureInput.sensorData.pressure >
70",
    "actions": [
      {
        "setVariable": {
          "variableName": "pressureThresholdBreach",
          "value":
"$variable.pressureThresholdBreach + 3"
        }
      },
      {
        "nextState": "Dangerous"
      }
    ]
  },
  {
    "stateName": "Dangerous",
    "onEnter": {
      "events": [
        {
          "eventName": "Pressure Threshold Breached",
          "condition": "$variable.pressureThresholdBreach > 1",
          "actions": [
            {
              "sns": {
                "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
              }
            }
          ]
        }
      ]
    },
    "onInput": {
      "events": [
        {
          "eventName": "Overpressurized",
          "condition": "$input.PressureInput.sensorData.pressure >
70",
          "actions": [
            {
              "setVariable": {
                "variableName": "pressureThresholdBreach",

```

```

        "value": "3"
      }
    }
  ],
},
{
  "eventName": "Pressure Okay",
  "condition": "$input.PressureInput.sensorData.pressure
<= 70",
  "actions": [
    {
      "setVariable": {
        "variableName": "pressureThresholdBreach",
        "value":
"$variable.pressureThresholdBreach - 1"
      }
    }
  ]
},
],
"transitionEvents": [
  {
    "eventName": "BackToNormal",
    "condition": "$input.PressureInput.sensorData.pressure
<= 70 && $variable.pressureThresholdBreach <= 1",
    "nextState": "Normal"
  }
]
},
"onExit": {
  "events": [
    {
      "eventName": "Normal Pressure Restored",
      "condition": "true",
      "actions": [
        {
          "sns": {
            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
          }
        }
      ]
    }
  ]
}
]

```



```

    }
  },
  ],
  "initialStateName": "Normal"
},
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

出力:

```

{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560799387.719,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560799387.719,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "2"
  }
}

```

詳細については、AWS 「IoT Events API Reference [UpdateDetectorModel](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateDetectorModel](#)」の「」を参照してください。

update-input

次のコード例は、update-input を使用する方法を示しています。

AWS CLI

入力を更新するには

次のupdate-input例では、指定された入力を新しい説明と定義で更新します。

```

aws iotevents update-input \
  --cli-input-json file://pressureInput.json

```

pressureInput.json の内容:

```
{
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor",
  "inputDefinition": {
    "attributes": [
      { "jsonPath": "sensorData.pressure" },
      { "jsonPath": "motorid" }
    ]
  }
}
```

出力:

```
{
  "inputConfiguration": {
    "status": "ACTIVE",
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "lastUpdateTime": 1560795976.458,
    "creationTime": 1560795312.542,
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor"
  }
}
```

詳細については、AWS IoT Events API リファレンスの [UpdateInput](#) 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateInput](#)」の「」を参照してください。

AWS IoT Events-Data を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Events-Data。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-put-message

次の例は、batch-put-message を使用する方法を説明しています。

AWS CLI

AWS IoT Events にメッセージ (入力) を送信するには

次のbatch-put-message例では、一連のメッセージを AWS IoT Events システムに送信します。各メッセージペイロードは、指定した入力 () inputName に変換され、その入力を監視するディテクターに取り込まれます。複数のメッセージが送信された場合、メッセージが処理される順序は保証されません。注文を保証するには、メッセージを一度に1つずつ送信し、正常なレスポンスを待つ必要があります。

```
aws iotevents-data batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://highPressureMessage.json
```

highPressureMessage.json の内容:

```
{  
  "messages": [  
    {  
      "messageId": "00001",  
      "inputName": "PressureInput",  
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":  
80, \"temperature\": 39} }"  
    }  
  ]  
}
```

出力:

```
{  
  "BatchPutMessageErrorEntries": []
```

```
}
```

詳細については、AWS IoT Events デベロッパーガイド* [BatchPutMessage](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchPutMessage](#)」の「」を参照してください。

batch-update-detector

次のコード例は、batch-update-detector を使用する方法を示しています。

AWS CLI

ディテクターを更新するには (インスタンス)

次のbatch-update-detector例では、指定されたディテクターモデル内の 1 つ以上のディテクター (インスタンス) の状態、変数値、タイマー設定を更新します。

```
aws iotevents-data batch-update-detector \  
  --cli-input-json file://budFulton-A32.json
```

budFulton-A32.json の内容:

```
{  
  "detectors": [  
    {  
      "messageId": "00001",  
      "detectorModelName": "motorDetectorModel",  
      "keyValue": "Fulton-A32",  
      "state": {  
        "stateName": "Normal",  
        "variables": [  
          {  
            "name": "pressureThresholdBreached",  
            "value": "0"  
          }  
        ],  
        "timers": [  
        ]  
      }  
    }  
  ]  
}
```

```
}
```

出力:

```
{
  "batchUpdateDetectorErrorEntries": []
}
```

詳細については、AWS IoT Events デベロッパーガイド [BatchUpdateDetector](#) の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchUpdateDetector](#)」の「」を参照してください。

create-detector-model

次の例は、create-detector-model を使用する方法を説明しています。

AWS CLI

ディテクターモデルを作成するには

次のcreate-detector-model例では、ディテクターモデルを作成します。

```
aws iotevents create-detector-model \
  --cli-input-json file://motorDetectorModel.json
```

motorDetectorModel.json の内容:

```
{
  "detectorModelName": "motorDetectorModel",
  "detectorModelDefinition": {
    "states": [
      {
        "stateName": "Normal",
        "onEnter": {
          "events": [
            {
              "eventName": "init",
              "condition": "true",
              "actions": [
                {
```

```

                "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "0"
                }
            }
        ]
    },
    "onInput": {
        "transitionEvents": [
            {
                "eventName": "Overpressurized",
                "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
                "actions": [
                    {
                        "setVariable": {
                            "variableName": "pressureThresholdBreached",
                            "value":
"$variable.pressureThresholdBreached + 3"
                        }
                    }
                ],
                "nextState": "Dangerous"
            }
        ]
    },
},
{
    "stateName": "Dangerous",
    "onEnter": {
        "events": [
            {
                "eventName": "Pressure Threshold Breached",
                "condition": "$variable.pressureThresholdBreached &gt;
1",
                "actions": [
                    {
                        "sns": {
                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                        }
                    }
                ]
            }
        ]
    }
}

```

```

        ]
      }
    ]
  },
  "onInput": {
    "events": [
      {
        "eventName": "Overpressurized",
        "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
        "actions": [
          {
            "setVariable": {
              "variableName": "pressureThresholdBreached",
              "value": "3"
            }
          }
        ]
      },
      {
        "eventName": "Pressure Okay",
        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70",
        "actions": [
          {
            "setVariable": {
              "variableName": "pressureThresholdBreached",
              "value":
"$variable.pressureThresholdBreached - 1"
            }
          }
        ]
      }
    ],
    "transitionEvents": [
      {
        "eventName": "BackToNormal",
        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70 &amp; $variable.pressureThresholdBreached &lt;= 1",
        "nextState": "Normal"
      }
    ]
  },
  "onExit": {

```

```
    "events": [
      {
        "eventName": "Normal Pressure Restored",
        "condition": "true",
        "actions": [
          {
            "sns": {
              "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
            }
          }
        ]
      }
    ],
    "initialStateName": "Normal"
  },
  "key": "motorid",
  "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}
```

出力:

```
{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560796816.077,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560796816.077,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
}
```

詳細については、AWS IoT Events デベロッパーガイド* [CreateDetectorModel](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDetectorModel](#)」の「」を参照してください。

create-input

次のコード例は、create-input を使用する方法を示しています。

AWS CLI

入力を作成するには

次のcreate-input例では、入力を作成します。

```
aws iotevents create-input \  
  --cli-input-json file://pressureInput.json
```

pressureInput.json の内容:

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

出力:

```
{  
  "inputConfiguration": {  
    "status": "ACTIVE",  
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
    "lastUpdateTime": 1560795312.542,  
    "creationTime": 1560795312.542,  
    "inputName": "PressureInput",  
    "inputDescription": "Pressure readings from a motor"  
  }  
}
```

詳細については、AWS IoT Events デベロッパーガイド* [CreateInput](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInput](#)」の「」を参照してください。

delete-detector-model

次の例は、delete-detector-model を使用する方法を説明しています。

AWS CLI

ディテクターモデルを削除するには

次のdelete-detector-model例では、ディテクターモデルを削除します。ディテクターモデルのアクティブなインスタンスも削除されます。

```
aws iotevents delete-detector-model \  
  --detector-model-name motorDetectorModel*
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Events デベロッパーガイド[DeleteDetectorModel](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDetectorModel](#)」の「」を参照してください。

delete-input

次のコード例は、delete-input を使用する方法を示しています。

AWS CLI

入力を削除するには

次のdelete-input例では、入力を削除します。

```
aws iotevents delete-input \  
  --input-name PressureInput
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Events デベロッパーガイド* [DeleteInput](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInput](#)」の「」を参照してください。

describe-detector-model

次の例は、describe-detector-model を使用する方法を説明しています。

AWS CLI

ディテクターモデルに関する情報を取得するには

次のdescribe-detector-model例では、ディテクターモデルについて説明します。versionパラメータが指定されていない場合、コマンドは最新バージョンに関する情報を返します。

```
aws iotevents describe-detector-model \  
--detector-model-name motorDetectorModel
```

出力:

```
{  
  "detectorModel": {  
    "detectorModelConfiguration": {  
      "status": "ACTIVE",  
      "lastUpdateTime": 1560796816.077,  
      "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
      "creationTime": 1560796816.077,  
      "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/motorDetectorModel",  
      "key": "motorid",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    },  
    "detectorModelDefinition": {  
      "states": [  
        {  
          "onInput": {  
            "transitionEvents": [  
              {  
                "eventName": "Overpressurized",  
                "actions": [  
                  {  
                    "setVariable": {
```

```

        "variableName":
"pressureThresholdBreached",
        "value":
"$variable.pressureThresholdBreached + 3"
    }
    ],
    "condition":
"$input.PressureInput.sensorData.pressure > 70",
    "nextState": "Dangerous"
    }
    ],
    "events": []
},
"stateName": "Normal",
"onEnter": {
    "events": [
        {
            "eventName": "init",
            "actions": [
                {
                    "setVariable": {
                        "variableName":
"pressureThresholdBreached",
                        "value": "0"
                    }
                }
            ],
            "condition": "true"
        }
    ]
},
"onExit": {
    "events": []
}
},
{
    "onInput": {
        "transitionEvents": [
            {
                "eventName": "BackToNormal",
                "actions": [],

```

```
        "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
        "nextState": "Normal"
    }
],
"events": [
    {
        "eventName": "Overpressurized",
        "actions": [
            {
                "setVariable": {
                    "variableName":
"pressureThresholdBreached",
                    "value": "3"
                }
            }
        ],
        "condition":
"$input.PressureInput.sensorData.pressure > 70"
    },
    {
        "eventName": "Pressure Okay",
        "actions": [
            {
                "setVariable": {
                    "variableName":
"pressureThresholdBreached",
                    "value":
"$variable.pressureThresholdBreached - 1"
                }
            }
        ],
        "condition":
"$input.PressureInput.sensorData.pressure <= 70"
    }
],
"stateName": "Dangerous",
"onEnter": {
    "events": [
        {
            "eventName": "Pressure Threshold Breached",
            "actions": [
```

```

        {
            "sns": {
                "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
            }
        },
        "condition": "$variable.pressureThresholdBreached >
1"
    }
],
"onExit": {
    "events": [
        {
            "eventName": "Normal Pressure Restored",
            "actions": [
                {
                    "sns": {
                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                    }
                }
            ],
            "condition": "true"
        }
    ]
}
],
"initialStateName": "Normal"
}
}
}

```

詳細については、AWS IoT Events デベロッパーガイド* [DescribeDetectorModel](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDetectorModel](#)」の「」を参照してください。

describe-detector

次の例は、describe-detector を使用する方法を説明しています。

AWS CLI

ディテクターに関する情報を取得するには (インスタンス)

次のdescribe-detector例では、指定されたディテクター (インスタンス) に関する情報を返します。

```
aws iotevents-data describe-detector \  
  --detector-model-name motorDetectorModel \  
  --key-value "Fulton-A32"
```

出力:

```
{  
  "detector": {  
    "lastUpdateTime": 1560797852.776,  
    "creationTime": 1560797852.775,  
    "state": {  
      "variables": [  
        {  
          "name": "pressureThresholdBreached",  
          "value": "3"  
        }  
      ],  
      "stateName": "Dangerous",  
      "timers": []  
    },  
    "keyValue": "Fulton-A32",  
    "detectorModelName": "motorDetectorModel",  
    "detectorModelVersion": "1"  
  }  
}
```

詳細については、AWS IoT Events デベロッパーガイド [DescribeDetector](#) の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDetector](#)」の「」を参照してください。

describe-input

次のコード例は、describe-input を使用する方法を示しています。

AWS CLI

入力に関する情報を取得するには

次のdescribe-input例では、入力の詳細を取得します。

```
aws iotevents describe-input \  
  --input-name PressureInput
```

出力:

```
{  
  "input": {  
    "inputConfiguration": {  
      "status": "ACTIVE",  
      "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/  
PressureInput",  
      "lastUpdateTime": 1560795312.542,  
      "creationTime": 1560795312.542,  
      "inputName": "PressureInput",  
      "inputDescription": "Pressure readings from a motor"  
    },  
    "inputDefinition": {  
      "attributes": [  
        {  
          "jsonPath": "sensorData.pressure"  
        },  
        {  
          "jsonPath": "motorid"  
        }  
      ]  
    }  
  }  
}
```

詳細については、AWS IoT Events デベロッパーガイド* [DescribeInput](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInput](#)」の「」を参照してください。

describe-logging-options

次の例は、describe-logging-options を使用する方法を説明しています。

AWS CLI

ログ記録設定に関する情報を取得するには

次のdescribe-logging-options例では、現在の AWS IoT Events ログ記録オプションを取得します。

```
aws iotevents describe-logging-options
```

出力:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "enabled": false,
    "level": "ERROR"
  }
}
```

詳細については、AWS IoT Events デベロッパーガイド*[DescribeLoggingOptions](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoggingOptions](#)」の「」を参照してください。

list-detector-model-versions

次の例は、list-detector-model-versions を使用する方法を説明しています。

AWS CLI

ディテクターモデルのバージョンに関する情報を取得するには

次のlist-detector-model-versions例では、ディテクターモデルのすべてのバージョンを一覧表示します。各探知器モデルのバージョンに関連付けられているメタデータのみが返されます。

```
aws iotevents list-detector-model-versions \  
  --detector-model-name motorDetectorModel
```

出力:

```
{  
  "detectorModelVersionSummaries": [  
    {  
      "status": "ACTIVE",  
      "lastUpdateTime": 1560796816.077,  
      "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
      "creationTime": 1560796816.077,  
      "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/motorDetectorModel",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    }  
  ]  
}
```

詳細については、AWS IoT Events デベロッパーガイド*[ListDetectorModelVersions](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDetectorModelVersions](#)」の「」を参照してください。

list-detector-models

次の例は、list-detector-models を使用する方法を説明しています。

AWS CLI

ディテクターモデルのリストを取得するには

次のlist-detector-models例では、作成したディテクターモデルを一覧表示します。各探知器モデルに関連付けられているメタデータのみが返されます。

```
aws iotevents list-detector-models
```

出力:

```
{
  "detectorModelSummaries": [
    {
      "detectorModelName": "motorDetectorModel",
      "creationTime": 1552072424.212
      "detectorModelDescription": "Detect overpressure in a motor."
    }
  ]
}
```

詳細については、AWS IoT Events デベロッパーガイド「[ListDetectorModels](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDetectorModels](#)」の「」を参照してください。

list-detectors

次の例は、list-detectors を使用する方法を説明しています。

AWS CLI

ディテクターモデルのディテクターのリストを取得するには

次のlist-detectors例では、ディテクター (ディテクターモデルのインスタンス) を一覧表示します。

```
aws iotevents-data list-detectors \
  --detector-model-name motorDetectorModel
```

出力:

```
{
  "detectorSummaries": [
    {
      "lastUpdateTime": 1558129925.2,
      "creationTime": 1552073155.527,
```

```
    "state": {
      "stateName": "Normal"
    },
    "keyValue": "Fulton-A32",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
]
```

詳細については、AWS IoT Events デベロッパーガイド* [ListDetectors](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDetectors](#)」の「」を参照してください。

list-inputs

次のコード例は、list-inputs を使用する方法を示しています。

AWS CLI

入力を一覧表示するには

次のlist-inputs例では、作成した入力を一覧表示します。

```
aws iotevents list-inputs
```

出力:

```
{
  "status": "ACTIVE",
  "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
  "lastUpdateTime": 1551742986.768,
  "creationTime": 1551742986.768,
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor"
}
```

詳細については、AWS IoT Events デベロッパーガイド* [ListInputs](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInputs](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースに割り当てられたタグを一覧表示するには

次のlist-tags-for-resource例では、リソースに割り当てたタグ (メタデータ) を一覧表示します。

```
aws iotevents list-tags-for-resource \  
  --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

出力:

```
{  
  "tags": [  
    {  
      "value": "motor",  
      "key": "deviceType"  
    }  
  ]  
}
```

詳細については、AWS IoT Events デベロッパーガイド* [ListTagsForResource](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-logging-options

次の例は、put-logging-options を使用する方法を説明しています。

AWS CLI

ログ記録オプションを設定するには

次のlist-tags-for-resource例では、AWS IoT Events のログ記録オプションを設定または更新します。いずれかの loggingOptions フィールドの値を更新する場合、変更が有効になるまでに最大で1分かかることに注意してください。また、roleArnフィールドで指定したロール

にアタッチされたポリシーを変更する場合 (無効なポリシーを修正する場合など)、その変更が有効になるまでに最大 5 分かかります。

```
aws iotevents put-logging-options \  
  --cli-input-json file://logging-options.json
```

logging-options.json の内容:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "level": "DEBUG",  
    "enabled": true,  
    "detectorDebugOptions": [  
      {  
        "detectorModelName": "motorDetectorModel",  
        "keyValue": "Fulton-A32"  
      }  
    ]  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Events デベロッパーガイド* [PutLoggingOptions](#) の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLoggingOptions](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを追加するには

次のtag-resource例では、特定のリソースのタグを に追加または変更します。タグは、リソースの管理に使用できるメタデータです。

```
aws iotevents tag-resource \  
  --cli-input-json file://tag-resource.json
```

```
--cli-input-json file://pressureInput.tag.json
```

pressureInput.tag.json の内容:

```
{
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
  "tags": [
    {
      "key": "deviceType",
      "value": "motor"
    }
  ]
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Events デベロッパーガイド* [TagResource](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースからタグを削除する

次のuntag-resource例では、リソースから指定されたタグを削除します。

```
aws iotevents untag-resource \  
  --cli-input-json file://pressureInput.untag.json
```

pressureInput.untag.json の内容:

```
{
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
  "tagKeys": [
    "deviceType"
  ]
}
```

```
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Events デベロッパーガイド* [UntagResource](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-detector-model

次の例は、update-detector-model を使用する方法を説明しています。

AWS CLI

ディテクターモデルを更新するには

次のupdate-detector-model例では、ディテクターモデルを更新します。以前のバージョンによって生成されたディテクター (インスタンス) が削除され、新しい入力が到着すると再作成されます。

```
aws iotevents update-detector-model \  
  --cli-input-json file://motorDetectorModel.update.json
```

motorDetectorModel.update.json の内容 :

```
{  
  "detectorModelName": "motorDetectorModel",  
  "detectorModelDefinition": {  
    "states": [  
      {  
        "stateName": "Normal",  
        "onEnter": {  
          "events": [  
            {  
              "eventName": "init",  
              "condition": "true",  
              "actions": [  
                {  
                  "setVariable": {  
                    "variableName": "pressureThresholdBreach",
```



```
        "value": "0"
      }
    }
  ]
},
"onInput": {
  "transitionEvents": [
    {
      "eventName": "Overpressurized",
      "condition": "$input.PressureInput.sensorData.pressure > 70",
      "actions": [
        {
          "setVariable": {
            "variableName": "pressureThresholdBreach",
            "value": "$variable.pressureThresholdBreach + 3"
          }
        }
      ],
      "nextState": "Dangerous"
    }
  ]
},
{
  "stateName": "Dangerous",
  "onEnter": {
    "events": [
      {
        "eventName": "Pressure Threshold Breached",
        "condition": "$variable.pressureThresholdBreach > 1",
        "actions": [
          {
            "sns": {
              "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
            }
          }
        ]
      }
    ]
  },
  "onInput": {
```

```
"events": [
  {
    "eventName": "Overpressurized",
    "condition": "$input.PressureInput.sensorData.pressure > 70",
    "actions": [
      {
        "setVariable": {
          "variableName": "pressureThresholdBreached",
          "value": "3"
        }
      }
    ]
  },
  {
    "eventName": "Pressure Okay",
    "condition": "$input.PressureInput.sensorData.pressure <= 70",
    "actions": [
      {
        "setVariable": {
          "variableName": "pressureThresholdBreached",
          "value": "$variable.pressureThresholdBreached - 1"
        }
      }
    ]
  }
],
"transitionEvents": [
  {
    "eventName": "BackToNormal",
    "condition": "$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
    "nextState": "Normal"
  }
],
"onExit": {
  "events": [
    {
      "eventName": "Normal Pressure Restored",
      "condition": "true",
      "actions": [
        {
          "sns": {
```

```

        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
      }
    }
  ]
}
],
"initialStateName": "Normal"
},
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

出力:

```

{
  "detectorModelConfiguration": {
    "status": "ACTIVATING",
    "lastUpdateTime": 1560799387.719,
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "creationTime": 1560799387.719,
    "detectorModelArn": "arn:aws:iotevents:us-west-2:123456789012:detectorModel/
motorDetectorModel",
    "key": "motorid",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "2"
  }
}

```

詳細については、AWS IoT Events デベロッパーガイド[UpdateDetectorModel](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDetectorModel](#)」の「」を参照してください。

update-input

次のコード例は、update-input を使用する方法を示しています。

AWS CLI

入力を更新するには

次のupdate-input例では、入力を更新します。

```
aws iotevents update-input \  
  --cli-input-json file://pressureInput.json
```

pressureInput.json の内容:

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

出力:

```
{  
  "inputConfiguration": {  
    "status": "ACTIVE",  
    "inputArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
    "lastUpdateTime": 1560795976.458,  
    "creationTime": 1560795312.542,  
    "inputName": "PressureInput",  
    "inputDescription": "Pressure readings from a motor"  
  }  
}
```

詳細については、AWS IoT Events デベロッパーガイド* [UpdateInput](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateInput](#)」の「」を参照してください。

AWS IoT Greengrass を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Greengrass。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-role-to-group

次の例は、associate-role-to-group を使用する方法を説明しています。

AWS CLI

ロールを Greengrass グループに関連付けるには

次のassociate-role-to-group例では、指定されたIAMロールを Greengrass グループに関連付けます。グループロールは、ローカルの Lambda 関数とコネクタが AWS サービスにアクセスするために使用します。例えば、グループロールが CloudWatch Logs 統合に必要なアクセス許可を付与する場合があります。

```
aws greengrass associate-role-to-group \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b \  
  --role-arn arn:aws:iam::123456789012:role/GG-Group-Role
```

出力:

```
{  
  "AssociatedAt": "2019-09-10T20:03:30Z"  
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの [「グループロールの設定」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateRoleToGroup](#)」の「」を参照してください。

associate-service-role-to-account

次の例は、associate-service-role-to-account を使用する方法を説明しています。

AWS CLI

サービスロールを AWS アカウントに関連付けるには

次のassociate-service-role-to-account例では、で指定されたIAMサービスロールをARN、AWS アカウントのAWS IoT Greengrass に関連付けます。でサービスロールを事前に作成しIAM、AWS IoT Greengrass がこのロールを引き受けることができるポリシードキュメントを関連付ける必要があります。

```
aws greengrass associate-service-role-to-account \  
  --role-arn "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
```

出力:

```
{  
  "AssociatedAt": "2019-06-25T18:12:45Z"  
}
```

詳細については、AWS IoT [Greengrass デベロッパーガイド](#)の「[Greengrass サービスロール](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateServiceRoleToAccount](#)」の「」を参照してください。

create-connector-definition-version

次のコード例は、create-connector-definition-version を使用する方法を示しています。

AWS CLI

コネクタ定義バージョンを作成するには

次のcreate-connector-definition-version例では、コネクタ定義バージョンを作成し、指定されたコネクタ定義に関連付けます。バージョン内のすべてのコネクタは、パラメータの値を定義します。

```
aws greengrass create-connector-definition-version \  
  --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \  
  --connectors "[{"Id": "MyTwilioNotificationsConnector",  
  "ConnectorArn": "arn:aws:greengrass:us-west-2:/connectors/  
TwilioNotifications/versions/2", "Parameters": {"TWILIO_ACCOUNT_SID  
": "AC1a8d4204890840d7fc482aab38090d57", "TwilioAuthTokenSecretArn":  
"arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-  
ntSlp6", "TwilioAuthTokenSecretArn-ResourceId": "TwilioAuthToken",  
"DefaultFromPhoneNumber": "4254492999"}}]"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
connectors/55d0052b-0d7d-44d6-b56f-21867215e118/versions/33f709a0-c825-49cb-9eea-  
dc8964fbd635",  
  "CreationTimestamp": "2019-06-24T20:46:30.134Z",  
  "Id": "55d0052b-0d7d-44d6-b56f-21867215e118",  
  "Version": "33f709a0-c825-49cb-9eea-dc8964fbd635"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConnectorDefinitionVersion](#)」の「」を参照してください。

create-connector-definition

次のコード例は、create-connector-definition を使用する方法を示しています。

AWS CLI

コネクタ定義を作成するには

次のcreate-connector-definition例では、コネクタ定義と初期コネクタ定義バージョンを作成します。初期バージョンにはコネクタが 1 つ含まれています。バージョン内のすべてのコネクタは、パラメータの値を定義します。

```
aws greengrass create-connector-definition \  
  --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \  
  --connectors [{"Id": "MyTwilioNotificationsConnector",  
  "ConnectorArn": "arn:aws:greengrass:us-west-2:/connectors/  
TwilioNotifications/versions/2", "Parameters": {"TWILIO_ACCOUNT_SID  
": "AC1a8d4204890840d7fc482aab38090d57", "TwilioAuthTokenSecretArn":  
"arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-  
ntSlp6", "TwilioAuthTokenSecretArn-ResourceId": "TwilioAuthToken",  
"DefaultFromPhoneNumber": "4254492999"}}]"
```

```
--name MySNSConnector \  
--initial-version "{\"Connectors\": [{\"Id\": \"MySNSConnector\", \"ConnectorArn\": \"arn:aws:greengrass:us-west-2:/connectors/SNS/versions/1\", \"Parameters\": {\"DefaultSNSArn\": \"arn:aws:sns:us-west-2:123456789012:GGConnectorTopic\"}}]}
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",  
  "CreationTimestamp": "2019-06-19T19:30:01.300Z",  
  "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",  
  "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",  
  "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-c7c2-4a26-a7e2-7bf478ea2623",  
  "Name": "MySNSConnector"  
}
```

詳細については、AWS IoT [Greengrass デベロッパーガイドの「Greengrass コネクタの開始方法 \(CLI\)」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateConnectorDefinition](#)」の「」を参照してください。

create-core-definition-version

次の例は、create-core-definition-version を使用する方法を説明しています。

AWS CLI

コア定義バージョンを作成するには

次のcreate-core-definition-version例では、コア定義バージョンを作成し、指定したコア定義に関連付けます。バージョンには 1 つのコアのみを含めることができます。コアを作成する前に、対応する AWS IoT モノを作成してプロビジョニングする必要があります。このプロセスには、iot コマンドCertificateArnに必要な ThingArnと を返す次のcreate-core-definition-versionコマンドが含まれます。

コアデバイスに対応する AWS IoT モノを作成します。


```
aws iot create-thing \  
  --thing-name "MyCoreDevice"
```

出力:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",  
  "thingName": "MyCoreDevice",  
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"  
}
```

パブリックキーとプライベートキー、およびモノのコアデバイス証明書を作成します。この例では `create-keys-and-certificate` コマンドを使用し、現在のディレクトリへの書き込みアクセス許可が必要です。または、`create-certificate-from-csr` コマンドを使用できます。

```
aws iot create-keys-and-certificate \  
  --set-as-active \  
  --certificate-pem-outfile "myCore.cert.pem" \  
  --public-key-outfile "myCore.public.key" \  
  --private-key-outfile "myCore.private.key"
```

出力:

```
{  
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",  
  "certificatePem": "-----BEGIN CERTIFICATE-----  
  \nMIIDWTCAkGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END  
  CERTIFICATE-----\n",  
  "keyPair": {  
    "PublicKey": "-----BEGIN PUBLIC KEY-----  
    \nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END  
    PUBLIC KEY-----\n",  
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----  
    \nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjBMXDcNOK4rMCxDR...fvY4+te\n-----END  
    RSA PRIVATE KEY-----\n"  
  },  
  "certificateId":  
  "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"  
}
```

iot および greengrass アクションを許可する AWS IoT ポリシーを作成します。簡単にするために、次のポリシーではすべてのリソースに対するアクションを許可しますが、ポリシーはより制限的である必要があります。

```
aws iot create-policy \  
  --policy-name "Core_Devices" \  
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\  
  \":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect\  
  \",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":\  
  [\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot>DeleteThingShadow\"],\  
  \"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource\  
  \":[\"*\"]}]}"
```

出力:

```
{  
  "policyName": "Core_Devices",  
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",  
  "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\  
  \":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect\  
  \",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":\  
  [\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot>DeleteThingShadow\"],\  
  \"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource\  
  \":[\"*\"]}]}",  
  "policyVersionId": "1"  
}
```

ポリシーを証明書にアタッチします。

```
aws iot attach-policy \  
  --policy-name "Core_Devices" \  
  --target "arn:aws:iot:us-  
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

このコマンドでは何も出力されません。

モノを証明書にアタッチします。

```
aws iot attach-thing-principal \  
  --thing-name "MyCoreDevice" \  
  --policy-name "Core_Devices"
```

```
--principal "arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

このコマンドでは何も出力されません。

コア定義バージョンを作成します。

```
aws greengrass create-core-definition-version \  
  --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \  
  --cores "[{\"Id\":\"MyCoreDevice\"},{\"ThingArn\":\"arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice\"},{\"CertificateArn\":\"arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz\"},{\"SyncShadow\":true}]"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/3fdc1190-2ce5-44de-b98b-eec8f9571014",  
  "Version": "3fdc1190-2ce5-44de-b98b-eec8f9571014",  
  "CreationTimestamp": "2019-09-18T00:15:09.838Z",  
  "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12"  
}
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「IoT Greengrass Core の設定」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCoreDefinitionVersion](#)」の「」を参照してください。

create-core-definition

次のコード例は、create-core-definition を使用する方法を示しています。

AWS CLI

例 1: 空のコア定義を作成するには

次のcreate-core-definition例では、空の(初期バージョンなし) Greengrass コア定義を作成します。コアを使用する前に、create-core-definition-version コマンドを使用してコアの他のパラメータを指定する必要があります。

```
aws greengrass create-core-definition \  
  --name cliGroup_Core
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/  
b5c08008-54cb-44bd-9eec-c121b04283b5",  
  "CreationTimestamp": "2019-06-25T18:23:22.106Z",  
  "Id": "b5c08008-54cb-44bd-9eec-c121b04283b5",  
  "LastUpdatedTimestamp": "2019-06-25T18:23:22.106Z",  
  "Name": "cliGroup_Core"  
}
```

例 2: 初期バージョンでコア定義を作成するには

次の `create-core-definition` 例では、最初のコア定義バージョンを含むコア定義を作成します。バージョンには 1 つのコアのみを含めることができます。コアを作成する前に、対応する AWS IoT モノを作成してプロビジョニングする必要があります。このプロセスには、`iot` コマンド `CertificateArn` に必要な `ThingArn` とを返す次の `create-core-definition` コマンドが含まれます。

コアデバイスに対応する AWS IoT モノを作成します。

```
aws iot create-thing \  
  --thing-name "MyCoreDevice"
```

出力:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",  
  "thingName": "MyCoreDevice",  
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"  
}
```

パブリックキーとプライベートキー、およびモノのコアデバイス証明書を作成します。この例では `create-keys-and-certificate` コマンドを使用し、現在のディレクトリへの書き込みアクセス許可が必要です。または、`create-certificate-from-csr` コマンドを使用できます。

```
aws iot create-keys-and-certificate \
  --set-as-active \
  --certificate-pem-outfile "myCore.cert.pem" \
  --public-key-outfile "myCore.public.key" \
  --private-key-outfile "myCore.private.key"
```

出力:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCakGgAwIBATgIUCgq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbmXDCnOK4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}
```

iot および greengrass アクションを許可する AWS IoT ポリシーを作成します。簡単にするために、次のポリシーではすべてのリソースに対するアクションを許可しますが、ポリシーはより制限的である必要があります。

```
aws iot create-policy \
  --policy-name "Core_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\": \"Allow\", \"Action\": [\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect
\", \"iot:Receive\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\":
[\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot:DeleteThingShadow\"],
\"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"greengrass:*\"], \"Resource
\": [\"*\"]}]}"
```

出力:

```
{
  "policyName": "Core_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
  "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource\":[\"*\"]}]}\",
  "policyVersionId": "1"
}
```

ポリシーを証明書にアタッチします。

```
aws iot attach-policy \
  --policy-name "Core_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

このコマンドでは何も出力されません。

モノを証明書にアタッチします。

```
aws iot attach-thing-principal \
  --thing-name "MyCoreDevice" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

このコマンドでは何も出力されません。

コア定義を作成します。

```
aws greengrass create-core-definition \
  --name "MyCores" \
  --initial-version "{\"Cores\":[{\"Id\":\"MyCoreDevice\",\"ThingArn\":
\"arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice\",\"CertificateArn\":
\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz
\",\"SyncShadow\":true}]}"
```

出力:

```
{
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
  "Name": "MyCores",
  "LastUpdatedTimestamp": "2019-09-18T00:11:06.197Z",
  "LatestVersion": "cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
  "CreationTimestamp": "2019-09-18T00:11:06.197Z",
  "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12",
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「[IoT Greengrass Core の設定](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCoreDefinition](#)」の「」を参照してください。

create-deployment

次の例は、create-deployment を使用する方法を説明しています。

AWS CLI

Greengrass グループのバージョンのデプロイを作成するには

次のcreate-deployment例では、指定されたバージョンの Greengrass グループをデプロイします。

```
aws greengrass create-deployment \
  --deployment-type NewDeployment \
  --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
  --group-version-id "dc40c1e9-e8c8-4d28-a84d-a9cad5f599c9"
```

出力:

```
{
  "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/deployments/bfceb608-4e97-45bc-af5c-460144270308",
  "DeploymentId": "bfceb608-4e97-45bc-af5c-460144270308"
}
```

```
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの「[コネクタの開始方法 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDeployment](#)」の「」を参照してください。

create-device-definition-version

次の例は、create-device-definition-version を使用する方法を説明しています。

AWS CLI

デバイス定義バージョンを作成するには

次のcreate-device-definition-version例では、デバイス定義バージョンを作成し、指定したデバイス定義に関連付けます。バージョンは 2 つのデバイスを定義します。Greengrass デバイスを作成する前に、対応する AWS IoT モノを作成してプロビジョニングする必要があります。このプロセスには、Greengrass iot コマンドに必要な情報を取得するために実行する必要があります。次のコマンドが含まれています。

デバイスに対応する AWS IoT モノを作成します。

```
aws iot create-thing \  
  --thing-name "InteriorTherm"
```

出力:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",  
  "thingName": "InteriorTherm",  
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"  
}
```

モノのパブリックキーとプライベートキー、およびデバイス証明書を作成します。この例では create-keys-and-certificate コマンドを使用し、現在のディレクトリへの書き込みアクセス許可が必要です。または、create-certificate-from-csr コマンドを使用できます。

```
aws iot create-keys-and-certificate \  
  --set-as-active \  
  --set-as-active
```



```
--certificate-pem-outfile "myDevice.cert.pem" \
--public-key-outfile "myDevice.public.key" \
--private-key-outfile "myDevice.private.key"
```

出力:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCGq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjBMXDcNOK4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

iot および greengrass アクションを許可する AWS IoT ポリシーを作成します。簡単にするために、次のポリシーではすべてのリソースに対するアクションを許可しますが、ポリシーはより制限的になる可能性があります。

```
aws iot create-policy \
  --policy-name "GG_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\": \"Allow\", \"Action\": [\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect\", \"iot:Receive\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"greengrass:*\"], \"Resource\": [\"*\"]}]}\""
```

出力:

```
{
  "policyName": "GG_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
```

```

    "policyDocument": "{\\"Version\\":\\"2012-10-17\\",\\"Statement\\":[{\\"Effect\\":\\"Allow\\",\\"Action\\":[\\"iot:Publish\\",\\"iot:Subscribe\\",\\"iot:Connect\\",\\"iot:Receive\\"],\\"Resource\\":[\\"*\\"]},{\\"Effect\\":\\"Allow\\",\\"Action\\":[\\"iot:GetThingShadow\\",\\"iot:UpdateThingShadow\\",\\"iot:DeleteThingShadow\\"],\\"Resource\\":[\\"*\\"]},{\\"Effect\\":\\"Allow\\",\\"Action\\":[\\"greengrass:*\\"],\\"Resource\\":[\\"*\\"]}]}",
    "policyVersionId": "1"
  }

```

ポリシーを証明書にアタッチします。

```

aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

モノを証明書にアタッチする

```

aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

上記のように IoT モノを作成して設定したら、次の例の最初の 2 つのコマンド CertificateArn の ThingArn とを使用します。

```

aws greengrass create-device-definition-version \
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
  --devices "[{\\"Id\\":\\"InteriorTherm\\",\\"ThingArn\\":\\"arn:aws:iot:us-
west-2:123456789012:thing/InteriorTherm\\",\\"CertificateArn\\":\\"arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\\",
\\"SyncShadow\\":true},{\\"Id\\":\\"ExteriorTherm\\",\\"ThingArn\\":\\"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\\",\\"CertificateArn\\":\\"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\\",
\\"SyncShadow\\":true}]"

```

出力:

```

{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",

```

```
"Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
"CreationTimestamp": "2019-09-11T00:15:09.838Z",
"Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDeviceDefinitionVersion](#)」の「」を参照してください。

create-device-definition

次の例は、create-device-definition を使用する方法を説明しています。

AWS CLI

デバイス定義を作成するには

次のcreate-device-definition例では、最初のデバイス定義バージョンを含むデバイス定義を作成します。初期バージョンでは、2つのデバイスを定義します。Greengrass デバイスを作成する前に、対応する AWS IoT モノを作成してプロビジョニングする必要があります。このプロセスには、Greengrass iot コマンドに必要な情報を取得するために実行する必要がある次のコマンドが含まれています。

デバイスに対応する AWS IoT モノを作成します。

```
aws iot create-thing \  
  --thing-name "InteriorTherm"
```

出力:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",  
  "thingName": "InteriorTherm",  
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"  
}
```

モノのパブリックキーとプライベートキー、およびデバイス証明書を作成します。この例では create-keys-and-certificate コマンドを使用し、現在のディレクトリへの書き込みアクセス許可が必要です。または、create-certificate-from-csr コマンドを使用できます。

```
aws iot create-keys-and-certificate \  
  --thing-name "InteriorTherm"
```

```
--set-as-active \
--certificate-pem-outfile "myDevice.cert.pem" \
--public-key-outfile "myDevice.public.key" \
--private-key-outfile "myDevice.private.key"
```

出力:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCAkGgAwIBATgIUCGq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbmXDCnOK4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

iot および greengrass アクションを許可する AWS IoT ポリシーを作成します。簡単にするために、次のポリシーではすべてのリソースに対するアクションを許可しますが、ポリシーはより制限的になる可能性があります。

```
aws iot create-policy \
  --policy-name "GG_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\": \"Allow\", \"Action\": [\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect\", \"iot:Receive\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"], \"Resource\": [\"*\"]}, {\"Effect\": \"Allow\", \"Action\": [\"greengrass:*\"], \"Resource\": [\"*\"]}]}"
```

出力:

```
{
  "policyName": "GG_Devices",
```

```

    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/GG_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":"
    "\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect\","
    "\":\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":"
    "\":[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot>DeleteThingShadow\"],\":"
    "\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource\":"
    "\":[\"*\"]}]}",
    "policyVersionId": "1"
  }

```

ポリシーを証明書にアタッチします。

```

aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

モノを証明書にアタッチする

```

aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

上記のように IoT モノを作成して設定したら、次の例の最初の 2 つのコマンド CertificateArn の ThingArn とを使用します。

```

aws greengrass create-device-definition \
  --name "Sensors" \
  --initial-version "{\"Devices\":[{\"Id\":\"InteriorTherm\","
  "\":\"ThingArn\":\"arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm\","
  "\":\"CertificateArn\":\"arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\"},\":"
  "\":{\"SyncShadow\":true},{\"Id\":\"ExteriorTherm\",\"ThingArn\":\"arn:aws:iot:us-west-2:123456789012:thing/ExteriorTherm\","
  "\":\"CertificateArn\":\"arn:aws:iot:us-west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\"},\":"
  "\":{\"SyncShadow\":true}}}"

```

出力:

```
{
```

```

    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795",
    "Name": "Sensors",
    "LastUpdatedTimestamp": "2019-09-11T00:11:06.197Z",
    "LatestVersion": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
    "CreationTimestamp": "2019-09-11T00:11:06.197Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
  }

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDeviceDefinition](#)」の「」を参照してください。

create-function-definition-version

次のコード例は、create-function-definition-version を使用する方法を示しています。

AWS CLI

関数定義のバージョンを作成するには

次のcreate-function-definition-version例では、指定された関数定義の新しいバージョンを作成します。このバージョンでは、ID が の単一の関数を指定しHello-World-function、ファイルシステムへのアクセスを許可し、最大メモリサイズとタイムアウト期間を指定します。

```

aws greengrass create-function-definition-version \
  --cli-input-json '{"FunctionDefinitionId": "e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3","Functions": [{"Id": "Hello-World-function", "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:Greengrass_HelloWorld_Counter:gghw-alias", "FunctionConfiguration": {"Environment": {"AccessSysfs": true}, "Executable": "greengrassHelloWorldCounter.function_handler", "MemorySize": 16000, "Pinned": false, "Timeout": 25}}]}'

```

出力:

```
{
```

```

    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3/
versions/74abd1cc-637e-4abe-8684-9a67890f4043",
    "CreationTimestamp": "2019-06-25T22:03:43.376Z",
    "Id": "e626e8c9-3b8f-4bf3-9cdc-d26ecdeb9fa3",
    "Version": "74abd1cc-637e-4abe-8684-9a67890f4043"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateFunctionDefinitionVersion](#)」の「」を参照してください。

create-function-definition

次のコード例は、create-function-definition を使用する方法を示しています。

AWS CLI

Lambda 関数定義を作成するには

次のcreate-function-definition例では、Lambda 関数のリスト (この場合は、という名前の 1 つの関数のみのリストTempMonitorFunction) とその設定を指定して、Lambda 関数定義と初期バージョンを作成します。関数定義を作成する前に、Lambda 関数 が必要ですARN。関数とそのエイリアスを作成するには、Lambda の コマンドcreate-functionと publish-version コマンドを使用します。Lambda ARNのcreate-functionコマンドでは、アクセス許可が Greengrass グループロールで指定されているため、AWS IoT Greengrass はそのロールを使用しませんが、実行ロールの が必要です。IAM create-role コマンドを使用して空のロールを作成し、Lambda で使用する ARN を取得するcreate-functionか、既存の実行ロールを使用できます。

```

aws greengrass create-function-definition \
  --name MyGreengrassFunctions \
  --initial-version "{\"Functions\": [{\"Id\": \"TempMonitorFunction\",
  \"FunctionArn\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"FunctionConfiguration
\": {\"Executable\": \"temp_monitor.function_handler\", \"MemorySize\": 16000,
  \"Timeout\": 5}}]}"

```

出力:

```
{
```

```
"Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/3b0d0080-87e7-48c6-b182-503ec743a08b",
"CreationTimestamp": "2019-06-19T22:24:44.585Z",
"Id": "3b0d0080-87e7-48c6-b182-503ec743a08b",
"LastUpdatedTimestamp": "2019-06-19T22:24:44.585Z",
"LatestVersion": "67f918b9-efb4-40b0-b87c-de8c9faf085b",
"LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/versions/67f918b9-
efb4-40b0-b87c-de8c9faf085b",
"Name": "MyGreengrassFunctions"
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの[AWS 「コマンドラインインターフェイスを使用してローカルリソースアクセスを設定する方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFunctionDefinition](#)」の「」を参照してください。

create-group-certificate-authority

次のコード例は、create-group-certificate-authority を使用する方法を示しています。

AWS CLI

グループの認証局 (CA) を作成するには

次のcreate-group-certificate-authority例では、指定されたグループの CA を作成またはローテーションします。

```
aws greengrass create-group-certificate-authority \
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
```

出力:

```
{
  "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/8eaadd72-ce4b-4f15-892a-0cc4f3a343f1/certificateauthorities/
d31630d674c4437f6c5dbc0dca56312a902171ce2d086c38e509c8EXAMPLEecc5"
}
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)「AWS 」のIoT Greengrass セキュリティ」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateGroupCertificateAuthority](#)」の「」を参照してください。

create-group-version

次のコード例は、create-group-version を使用する方法を示しています。

AWS CLI

Greengrass グループのバージョンを作成するには

次のcreate-group-version例では、グループバージョンを作成し、指定したグループと関連付けます。バージョンは、このグループバージョンに含めるエンティティを含むコア、リソース、コネクタ、関数、サブスクリプションバージョンを参照します。グループバージョンを作成する前に、これらのエンティティを作成する必要があります。

初期バージョンでリソース定義を作成するには、create-resource-definitioncommand.Toを使用して初期バージョンでコネクタ定義を作成し、command.To create-connector-definitionを使用して初期バージョンで関数定義を作成し、command.To create-function-definitionを使用して初期バージョンでサブスクリプション定義を作成し、command.To create-subscription-definitionを使用してARN最新のコア定義バージョンの取得し、get-group-version コマンドを使用して最新のグループバージョンの ID を指定します。

```
aws greengrass create-group-version \  
  --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \  
  --core-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/6a630442-8708-4838-ad36-eb98849d975e/versions/6c87151b-1fb4-4cb2-8b31-6ee715d8f8ba" \  
  --resource-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1" \  
  --connector-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/connectors/55d0052b-0d7d-44d6-b56f-21867215e118/versions/78a3331b-895d-489b-8823-17b4f9f418a0" \  
  --function-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/3b0d0080-87e7-48c6-b182-503ec743a08b/versions/67f918b9-efb4-40b0-b87c-de8c9faf085b" \  
  --subscription-definition-version-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/subscriptions/9d611d57-5d5d-44bd-a3b4-fecbbdd69112/versions/aa645c47-ac90-420d-9091-8c7ffa4f103f"
```

出力:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/
ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca/versions/e10b0459-4345-4a09-88a4-1af1f5d34638",
  "CreationTimestamp": "2019-06-20T18:42:47.020Z",
  "Id": "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca",
  "Version": "e10b0459-4345-4a09-88a4-1af1f5d34638"
}
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「IoT Greengrass グループオブジェクトモデルの概要」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateGroupVersion](#)」の「」を参照してください。

create-group

次のコード例は、create-group を使用する方法を示しています。

AWS CLI

Greengrass グループを作成するには

次のcreate-group例では、 という名前のグループを作成しますcli-created-group。

```
aws greengrass create-group \
  --name cli-created-group
```

出力:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/4e22bd92-898c-436b-ade5-434d883ff749",
  "CreationTimestamp": "2019-06-25T18:07:17.688Z",
  "Id": "4e22bd92-898c-436b-ade5-434d883ff749",
  "LastUpdatedTimestamp": "2019-06-25T18:07:17.688Z",
  "Name": "cli-created-group"
}
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「IoT Greengrass グループオブジェクトモデルの概要」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateGroup](#)」の「」を参照してください。

create-logger-definition-version

次の例は、create-logger-definition-version を使用する方法を説明しています。

AWS CLI

ロガー定義バージョンを作成するには

次のcreate-logger-definition-version例では、ロガー定義バージョンを作成し、ロガー定義に関連付けます。このバージョンでは、1) コアデバイスのファイルシステム上のシステムコンポーネントログ、2) コアデバイスのファイルシステム上のユーザー定義の Lambda 関数ログ、3) Amazon CloudWatch Logs 内のシステムコンポーネントログ、および 4) Amazon CloudWatch Logs 内のユーザー定義の Lambda 関数ログの 4 つのログ記録設定を定義します。注: CloudWatch Logs 統合の場合、グループロールは適切なアクセス許可を付与する必要があります。

```
aws greengrass create-logger-definition-version \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \  
  --loggers [{"Id\":\"1\",\"Component\":\"GreengrassSystem\",\"Level\":\"ERROR \  
 \",\"Space\":\"10240\",\"Type\":\"FileSystem\"},{\"Id\":\"2\",\"Component\":\"Lambda \  
 \",\"Level\":\"INFO\",\"Space\":\"10240\",\"Type\":\"FileSystem\"},{\"Id\":\"3\", \  
 \"Component\":\"GreengrassSystem\",\"Level\":\"WARN\",\"Type\":\"AWSCloudWatch\"}, \  
 {\"Id\":\"4\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Type\":\"AWSCloudWatch \  
 \"}]
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/loggers/  
 a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/49aedb1e-01a3-4d39-9871-3a052573f1ea",  
  "Version": "49aedb1e-01a3-4d39-9871-3a052573f1ea",  
  "CreationTimestamp": "2019-07-24T00:04:48.523Z",  
  "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"  
}
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「[IoT Greengrass Logs によるモニタリング](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateLoggerDefinitionVersion](#)」の「」を参照してください。

create-logger-definition

次の例は、create-logger-definition を使用する方法を説明しています。

AWS CLI

ロガー定義を作成するには

次のcreate-logger-definition例では、最初のロガー定義バージョンを含むロガー定義を作成します。初期バージョンでは、1) コアデバイスのファイルシステム上のシステムコンポーネントログ、2) コアデバイスのファイルシステム上のユーザー定義の Lambda 関数ログ、および 3) Amazon CloudWatch Logs のユーザー定義の Lambda 関数ログの 3 つのログ記録設定を定義します。注: CloudWatch Logs 統合の場合、グループロールは適切なアクセス許可を付与する必要があります。

```
aws greengrass create-logger-definition \  
  --name "LoggingConfigs" \  
  --initial-version "{\"Loggers\":{\"Id\":\"1\",\"Component\":\"GreengrassSystem\  
\",\"Level\":\"ERROR\",\"Space\":\"10240\",\"Type\":\"FileSystem\"},{\"Id\":  
\"2\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Space\":\"10240\",\"Type\":  
\"FileSystem\"},{\"Id\":\"3\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Type\":  
\"AWSCloudWatch\"}}}"
```

出力:

```
{  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0/versions/de1d9854-1588-4525-  
b25e-b378f60f2322",  
  "Name": "LoggingConfigs",  
  "LastUpdatedTimestamp": "2019-07-23T23:52:17.165Z",  
  "LatestVersion": "de1d9854-1588-4525-b25e-b378f60f2322",  
  "CreationTimestamp": "2019-07-23T23:52:17.165Z",  
  "Id": "a454b62a-5d56-4ca9-bdc4-8254e1662cb0",  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
loggers/a454b62a-5d56-4ca9-bdc4-8254e1662cb0"  
}
```

詳細については、[AWS IoT Greengrass デベロッパガイド](#)の「IoT Greengrass Logs によるモニタリング」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoggerDefinition](#)」の「」を参照してください。

create-resource-definition-version

次の例は、create-resource-definition-version を使用する方法を説明しています。

AWS CLI

リソース定義のバージョンを作成するには

次のcreate-resource-definition-version例では、 の新しいバージョンを作成します TwilioAuthToken。

```
aws greengrass create-resource-definition-version \  
  --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \  
  --resources "[{"Id": "TwilioAuthToken"}, {"Name": "MyTwilioAuthToken", "ResourceDataContainer": {"SecretsManagerSecretResourceData": {"ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-ntS1p6"}}}]"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/b3bcada0-5fb6-42df-bf0b-1ee4f15e769e",  
  "CreationTimestamp": "2019-06-24T21:17:25.623Z",  
  "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",  
  "Version": "b3bcada0-5fb6-42df-bf0b-1ee4f15e769e"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResourceDefinitionVersion](#)」の「」を参照してください。

create-resource-definition

次のコード例は、create-resource-definition を使用する方法を示しています。

AWS CLI

リソース定義を作成するには

次のcreate-resource-definition例では、Greengrass グループで使用するリソースのリストを含むリソース定義を作成します。この例では、リソースのリストを提供することで、リソース定義の初期バージョンが含まれています。このリストには、Twilio 認証トークン用のリソースが1つと、AWS Secrets Manager に保存されているシークレットARN用のが含まれます。リソース定義を作成する前に、シークレットを作成する必要があります。

```
aws greengrass create-resource-definition \  
  --name MyGreengrassResources \  
  --initial-version "{\"Resources\": [{\"Id\": \"TwilioAuthToken  
\", \"Name\": \"MyTwilioAuthToken\", \"ResourceDataContainer\":  
  {\"SecretsManagerSecretResourceData\": {\"ARN\": \"arn:aws:secretsmanager:us-  
west-2:123456789012:secret:greengrass-TwilioAuthToken-ntSlp6\"}}}]}"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",  
  "CreationTimestamp": "2019-06-19T21:51:28.212Z",  
  "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",  
  "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",  
  "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/a5f94d0b-  
f6bc-40f4-bb78-7a1c5fe13ba1",  
  "Name": "MyGreengrassResources"  
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの[AWS 「コマンドラインインターフェイスを使用してローカルリソースアクセスを設定する方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResourceDefinition](#)」の「」を参照してください。

create-software-update-job

次の例は、create-software-update-job を使用する方法を説明しています。

AWS CLI

コアのソフトウェア更新ジョブを作成するには

次のcreate-software-update-job例では、over-the-air名前が のコアで AWS IoT Greengrass Core ソフトウェアを更新する (OTA) 更新ジョブを作成しますMyFirstGroup_Core。このコマンドには、Amazon S3 のソフトウェア更新パッケージへのアクセスを許可するIAMロールが必要で、 を信頼されたエンティティiot.amazonaws.comとして含めます。

```
aws greengrass create-software-update-job \  
  --update-targets-architecture armv7l \  
  --update-targets ["arn:aws:iot:us-west-2:123456789012:thing/MyFirstGroup_Core \  
  \"] \  
  --update-targets-operating-system raspbian \  
  --software-to-update core \  
  --s3-url-signer-role arn:aws:iam::123456789012:role/OTA_signer_role \  
  --update-agent-log-level WARN
```

出力:

```
{  
  "IotJobId": "GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",  
  "IotJobArn": "arn:aws:iot:us-west-2:123456789012:job/  
GreengrassUpdateJob_30b353e3-3af7-4786-be25-4c446663c09e",  
  "PlatformSoftwareVersion": "1.9.3"  
}
```

詳細については、[OTA AWS IoT Greengrass デベロッパーガイド](#)の「IoT Greengrass コアソフトウェアの更新」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSoftwareUpdateJob](#)」の「」を参照してください。

create-subscription-definition-version

次のコード例は、create-subscription-definition-version を使用する方法を示しています。

AWS CLI

サブスクリプション定義の新しいバージョンを作成するには

次のcreate-subscription-definition-version例では、トリガー通知、温度入力、出力ステータスの3つのサブスクリプションを含むサブスクリプション定義の新しいバージョンを作成します。

```
aws greengrass create-subscription-definition-version \
  --subscription-definition-id "9d611d57-5d5d-44bd-a3b4-feccbdd69112" \
  --subscriptions "[{\\"Id\\": \\"TriggerNotification\\", \\"Source\\":
  \\"arn:aws:lambda:us-west-2:123456789012:function:TempMonitor:GG_TempMonitor
  \", \\"Subject\\": \\"twilio/txt\\", \\"Target\\": \\"arn:aws:greengrass:us-west-2:/:
  connectors/TwilioNotifications/versions/1\\\"},{\\"Id\\": \\"TemperatureInput\\", \\"Source
  \": \\"cloud\\", \\"Subject\\": \\"temperature/input\\", \\"Target\\": \\"arn:aws:lambda:us-
  west-2:123456789012:function:TempMonitor:GG_TempMonitor\\\"},{\\"Id\\": \\"OutputStatus
  \", \\"Source\\": \\"arn:aws:greengrass:us-west-2:/:connectors/TwilioNotifications/
  versions/1\\", \\"Subject\\": \\"twilio/message/status\\", \\"Target\\": \\"cloud\\\"}]"]
```

出力:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
  subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/7b65dfae-50b6-4d0f-
  b3e0-27728bfb0620",
  "CreationTimestamp": "2019-06-24T21:21:33.837Z",
  "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
  "Version": "7b65dfae-50b6-4d0f-b3e0-27728bfb0620"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSubscriptionDefinitionVersion](#)」の「」を参照してください。

create-subscription-definition

次の例は、create-subscription-definition を使用する方法を説明しています。

AWS CLI

サブスクリプション定義を作成するには

次のcreate-subscription-definition例では、サブスクリプション定義を作成し、その初期バージョンを指定します。最初のバージョンには、コネクタがサブスクリライブするMQTTトピック用のサブスクリプションと、関数が AWS IoT から温度測定値を受信できるようにするサブスクリプション、および AWS IoT がコネクタからステータス情報を受信できるようにするサ

ブスクリプションの 3 つが含まれています。この例では、ARNLambda の `create-alias` コマンドを使用して以前に作成された Lambda 関数エイリアスの を提供します。

```
aws greengrass create-subscription-definition \
  --initial-version "{\"Subscriptions\": [{\"Id\":
  \"TriggerNotification\", \"Source\": \"arn:aws:lambda:us-
  west-2:123456789012:function:TempMonitor:GG_TempMonitor\", \"Subject\":
  \"twilio/txt\", \"Target\": \"arn:aws:greengrass:us-west-2:/connectors/
  TwilioNotifications/versions/1\"},{\"Id\": \"TemperatureInput\", \"Source\":
  \"cloud\", \"Subject\": \"temperature/input\", \"Target\": \"arn:aws:lambda:us-
  west-2:123456789012:function:TempMonitor:GG_TempMonitor\"},{\"Id\": \"OutputStatus
  \", \"Source\": \"arn:aws:greengrass:us-west-2:/connectors/TwilioNotifications/
  versions/1\", \"Subject\": \"twilio/message/status\", \"Target\": \"cloud\"}]}"
```

出力:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
  subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112",
  "CreationTimestamp": "2019-06-19T22:34:26.677Z",
  "Id": "9d611d57-5d5d-44bd-a3b4-feccbdd69112",
  "LastUpdatedTimestamp": "2019-06-19T22:34:26.677Z",
  "LatestVersion": "aa645c47-ac90-420d-9091-8c7ffa4f103f",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
  definition/subscriptions/9d611d57-5d5d-44bd-a3b4-feccbdd69112/versions/aa645c47-
  ac90-420d-9091-8c7ffa4f103f"
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの「[コネクタの開始方法 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubscriptionDefinition](#)」の「」を参照してください。

delete-connector-definition

次のコード例は、`delete-connector-definition` を使用する方法を示しています。

AWS CLI

コネクタ定義を削除するには

次のdelete-connector-definition例では、指定された Greengrass コネクタ定義を削除します。グループが使用するコネクタ定義を削除すると、そのグループを正常にデプロイすることはできません。

```
aws greengrass delete-connector-definition \  
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConnectorDefinition](#)」の「」を参照してください。

delete-core-definition

次のコード例は、delete-core-definition を使用する方法を示しています。

AWS CLI

コア定義を削除するには

次のdelete-core-definition例では、すべてのバージョンを含む、指定された Greengrass コア定義を削除します。Greengrass グループに関連付けられているコアを削除すると、そのグループは正常にデプロイできません。

```
aws greengrass delete-core-definition \  
  --core-definition-id "ff36cc5f-9f98-4994-b468-9d9b6dc52abd"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCoreDefinition](#)」の「」を参照してください。

delete-device-definition

次のコード例は、delete-device-definition を使用する方法を示しています。

AWS CLI

デバイス定義を削除するには

次のdelete-device-definition例では、すべてのバージョンを含む、指定されたデバイス定義を削除します。グループバージョンで使用されるデバイス定義バージョンを削除すると、グループバージョンが正常にデプロイされません。

```
aws greengrass delete-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDeviceDefinition](#)」の「」を参照してください。

delete-function-definition

次の例は、delete-function-definition を使用する方法を説明しています。

AWS CLI

関数定義を削除するには

次のdelete-function-definition例では、指定された Greengrass 関数定義を削除します。グループが使用する関数定義を削除すると、そのグループを正常にデプロイすることはできません。

```
aws greengrass delete-function-definition \  
  --function-definition-id "fd4b906a-dff3-4c1b-96eb-52ebfcfac06a"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFunctionDefinition](#)」の「」を参照してください。

delete-group

次の例は、delete-group を使用する方法を説明しています。

AWS CLI

グループを削除するには

次のdelete-group例では、指定された Greengrass グループを削除します。

```
aws greengrass delete-group \  
  --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteGroup](#)」の「」を参照してください。

delete-logger-definition

次のコード例は、delete-logger-definition を使用する方法を示しています。

AWS CLI

ロガー定義を削除するには

次のdelete-logger-definition例では、すべてのロガー定義バージョンを含む、指定されたロガー定義を削除します。グループバージョンで使用されるロガー定義バージョンを削除すると、グループバージョンが正常にデプロイされません。

```
aws greengrass delete-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「[IoT Greengrass Logs によるモニタリング](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteLoggerDefinition](#)」の「」を参照してください。

delete-resource-definition

次のコード例は、delete-resource-definition を使用する方法を示しています。

AWS CLI

リソース定義を削除するには

次のdelete-resource-definition例では、すべてのリソースバージョンを含む、指定されたリソース定義を削除します。グループが使用するリソース定義を削除すると、そのグループを正常にデプロイすることはできません。

```
aws greengrass delete-resource-definition \  
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResourceDefinition](#)」の「」を参照してください。

delete-subscription-definition

次の例は、delete-subscription-definition を使用する方法を説明しています。

AWS CLI

サブスクリプション定義を削除するには

次のdelete-subscription-definition例では、指定された Greengrass サブスクリプション定義を削除します。グループで使用されているサブスクリプションを削除すると、そのグループを正常にデプロイすることはできません。

```
aws greengrass delete-subscription-definition \  
  --subscription-definition-id "cd6f1c37-d9a4-4e90-be94-01a7404f5967"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSubscriptionDefinition](#)」の「」を参照してください。

disassociate-role-from-group

次の例は、disassociate-role-from-group を使用する方法を説明しています。

AWS CLI

Greengrass グループからロールの関連付けを解除するには

次のdisassociate-role-from-group例では、指定された Greengrass グループからIAMロールの関連付けを解除します。

```
aws greengrass disassociate-role-from-group \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

出力:

```
{
  "DisassociatedAt": "2019-09-10T20:05:49Z"
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの「[グループロールの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateRoleFromGroup](#)」の「」を参照してください。

disassociate-service-role-from-account

次のコード例は、disassociate-service-role-from-account を使用する方法を示しています。

AWS CLI

AWS アカウントからサービスロールの関連付けを解除するには

次のdisassociate-service-role-from-account例では、AWS アカウントに関連付けられているサービスロールを削除します。どの AWS リージョンでもサービスロールを使用していない場合は、delete-role-policy コマンドを使用してロールからAWSGreengrassResourceAccessRolePolicy管理ポリシーをデタッチし、delete-role コマンドを使用してロールを削除します。

```
aws greengrass disassociate-service-role-from-account
```

出力:

```
{
  "DisassociatedAt": "2019-06-25T22:12:55Z"
}
```

詳細については、AWS IoT [Greengrass デベロッパーガイド](#)の「[Greengrass サービスロール](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateServiceRoleFromAccount](#)」の「」を参照してください。

get-associated-role

次の例は、get-associated-role を使用する方法を説明しています。

AWS CLI

Greengrass グループに関連付けられたロールを取得するには

次のget-associated-role例では、指定された Greengrass グループに関連付けられている IAMロールを取得します。グループロールは、ローカルの Lambda 関数とコネクタが AWS サービスにアクセスするために使用します。

```
aws greengrass get-associated-role \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

出力:

```
{  
  "RoleArn": "arn:aws:iam::123456789012:role/GG-Group-Role",  
  "AssociatedAt": "2019-09-10T20:03:30Z"  
}
```

詳細については、AWS IoT Greengrass デベロッパーガイド」の「[グループロールの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAssociatedRole](#)」の「」を参照してください。

get-bulk-deployment-status

次のコード例は、get-bulk-deployment-status を使用する方法を示しています。

AWS CLI

一括デプロイのステータスを確認するには

次のget-bulk-deployment-status例では、指定された一括デプロイオペレーションのステータス情報を取得します。この例では、デプロイするグループを指定したファイルの入力レコードが無効です。

```
aws greengrass get-bulk-deployment-status \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

```
--bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

出力:

```
{
  "BulkDeploymentMetrics": {
    "InvalidInputRecords": 1,
    "RecordsProcessed": 1,
    "RetryAttempts": 0
  },
  "BulkDeploymentStatus": "Completed",
  "CreatedAt": "2019-06-25T16:11:33.265Z",
  "tags": {}
}
```

詳細については、IoT Greengrass デベロッパーガイドの「[グループの一括デプロイの作成](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBulkDeploymentStatus](#)」の「」を参照してください。

get-connectivity-info

次のコード例は、get-connectivity-info を使用する方法を示しています。

AWS CLI

Greengrass コアの接続情報を取得するには

次のget-connectivity-info例は、デバイスが指定された Greengrass コアへの接続に使用できるエンドポイントを示しています。接続情報は、IP アドレスまたはドメイン名のリストで、対応するポート番号とオプションのユーザー定義メタデータが含まれます。

```
aws greengrass get-connectivity-info \
  --thing-name "MyGroup_Core"
```

出力:

```
{
  "ConnectivityInfo": [
    {
```



```
    "Metadata": "",
    "PortNumber": 8883,
    "HostAddress": "127.0.0.1",
    "Id": "AUTOIP_127.0.0.1_0"
  },
  {
    "Metadata": "",
    "PortNumber": 8883,
    "HostAddress": "192.168.1.3",
    "Id": "AUTOIP_192.168.1.3_1"
  },
  {
    "Metadata": "",
    "PortNumber": 8883,
    "HostAddress": "::1",
    "Id": "AUTOIP_::1_2"
  },
  {
    "Metadata": "",
    "PortNumber": 8883,
    "HostAddress": "fe80::1e69:ed93:f5b:f6d",
    "Id": "AUTOIP_fe80::1e69:ed93:f5b:f6d_3"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConnectivityInfo](#)」の「」を参照してください。

get-connector-definition-version

次の例は、get-connector-definition-version を使用する方法を説明しています。

AWS CLI

コネクタ定義の特定のバージョンに関する情報を取得するには

次のget-connector-definition-version例では、指定されたコネクタ定義の指定されたバージョンに関する情報を取得します。コネクタ定義IDsのすべてのバージョンの を取得するには、list-connector-definition-versions コマンドを使用します。コネクタ定義に追加された最新バージョンの ID を取得するには、get-connector-definition コマンドを使用して LatestVersionプロパティを確認します。

```
aws greengrass get-connector-definition-version \  
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8" \  
  --connector-definition-version-id "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-c7c2-4a26-  
a7e2-7bf478ea2623",  
  "CreationTimestamp": "2019-06-19T19:30:01.300Z",  
  "Definition": {  
    "Connectors": [  
      {  
        "ConnectorArn": "arn:aws:greengrass:us-west-2:./connectors/SNS/  
versions/1",  
        "Id": "MySNSConnector",  
        "Parameters": {  
          "DefaultSNSArn": "arn:aws:sns:us-  
west-2:123456789012:GGConnectorTopic"  
        }  
      }  
    ]  
  },  
  "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",  
  "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"  
}
```

詳細については、AWS IoT Greengrass [デベロッパーガイドの「Greengrass コネクタを使用したサービスとプロトコルとの統合」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConnectorDefinitionVersion](#)」の「」を参照してください。

get-connector-definition

次のコード例は、get-connector-definition を使用する方法を示しています。

AWS CLI

コネクタ定義に関する情報を取得するには

次の`get-connector-definition`例では、指定されたコネクタ定義に関する情報を取得します。コネクタ定義IDsの を取得するには、`list-connector-definitions` コマンドを使用します。

```
aws greengrass get-connector-definition \  
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",  
  "CreationTimestamp": "2019-06-19T19:30:01.300Z",  
  "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",  
  "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",  
  "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-  
c7c2-4a26-a7e2-7bf478ea2623",  
  "Name": "MySNSConnector",  
  "tags": {}  
}
```

詳細については、AWS IoT Greengrass [デベロッパーガイドの「Greengrass コネクタを使用したサービスとプロトコルとの統合」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConnectorDefinition](#)」の「」を参照してください。

get-core-definition-version

次のコード例は、`get-core-definition-version` を使用する方法を示しています。

AWS CLI

Greengrass コア定義の特定のバージョンに関する詳細を取得するには

次の`get-core-definition-version`例では、指定されたコア定義の指定されたバージョンに関する情報を取得します。コア定義IDsのすべてのバージョンの を取得するには、`list-core-definition-versions` コマンドを使用します。コア定義に追加された最新バージョンの ID を

取得するには、`get-core-definition` コマンドを使用して `LatestVersion` プロパティを確認します。

```
aws greengrass get-core-definition-version \  
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46" \  
  --core-definition-version-id "42aeaac3-fd9d-4312-a8fd-ffa9404a20e0"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/  
c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeaac3-fd9d-4312-a8fd-ffa9404a20e0",  
  "CreationTimestamp": "2019-06-18T16:21:21.351Z",  
  "Definition": {  
    "Cores": [  
      {  
        "CertificateArn": "arn:aws:iot:us-  
west-2:123456789012:cert/928dea7b82331b47c3ff77b0e763fc5e64e2f7c884e6ef391baed9b6b8e21b45",  
        "Id": "1a39aac7-0885-4417-91f6-23e4cea6c511",  
        "SyncShadow": false,  
        "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/  
GGGroup4Pi3_Core"  
      }  
    ]  
  },  
  "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",  
  "Version": "42aeaac3-fd9d-4312-a8fd-ffa9404a20e0"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetCoreDefinitionVersion](#)」の「」を参照してください。

get-core-definition

次のコード例は、`get-core-definition` を使用する方法を示しています。

AWS CLI

Greengrass コア定義の詳細を取得するには

次の`get-core-definition`例では、指定されたコア定義に関する情報を取得します。コア定義 IDs の取得するには、`list-core-definitions` コマンドを使用します。

```
aws greengrass get-core-definition \  
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd",  
  "CreationTimestamp": "2018-10-18T04:47:06.721Z",  
  "Id": "237d6916-27cf-457f-ba0c-e86cfb5d25cd",  
  "LastUpdatedTimestamp": "2018-10-18T04:47:06.721Z",  
  "LatestVersion": "bd2cd6d4-2bc5-468a-8962-39e071e34b68",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/cores/237d6916-27cf-457f-ba0c-e86cfb5d25cd/versions/  
bd2cd6d4-2bc5-468a-8962-39e071e34b68",  
  "tags": {}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCoreDefinition](#)」の「」を参照してください。

get-deployment-status

次のコード例は、get-deployment-status を使用する方法を示しています。

AWS CLI

デプロイのステータスを取得するには

次のget-deployment-status例では、指定された Greengrass グループの指定されたデプロイのステータスを取得します。デプロイ ID を取得するには、list-deployments コマンドを使用してグループ ID を指定します。

```
aws greengrass get-deployment-status \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \  
  --deployment-id "1065b8a0-812b-4f21-9d5d-e89b232a530f"
```

出力:

```
{  
  "DeploymentStatus": "Success",  
}
```

```

    "DeploymentType": "NewDeployment",
    "UpdatedAt": "2019-06-18T17:04:44.761Z"
  }

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDeploymentStatus](#)」の「」を参照してください。

get-device-definition-version

次の例は、get-device-definition-version を使用する方法を説明しています。

AWS CLI

デバイス定義バージョンを取得するには

次のget-device-definition-version例では、指定されたデバイス定義の指定されたバージョンに関する情報を取得します。デバイス定義IDsのすべてのバージョンの を取得するには、list-device-definition-versions コマンドを使用します。デバイス定義に追加された最新バージョンの ID を取得するには、get-device-definition コマンドを使用して LatestVersion プロパティを確認します。

```

aws greengrass get-device-definition-version \
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
  --device-definition-version-id "83c13984-6fed-447e-84d5-5b8aa45d5f71"

```

出力:

```

{
  "Definition": {
    "Devices": [
      {
        "CertificateArn": "arn:aws:iot:us-west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02",
        "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/ExteriorTherm",
        "SyncShadow": true,
        "Id": "ExteriorTherm"
      },
      {
        "CertificateArn": "arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",

```

```

        "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
InteriorTherm",
        "SyncShadow": true,
        "Id": "InteriorTherm"
    }
]
},
"Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
"CreationTimestamp": "2019-09-11T00:15:09.838Z",
"Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
"Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeviceDefinitionVersion](#)」の「」を参照してください。

get-device-definition

次の例は、get-device-definition を使用する方法を説明しています。

AWS CLI

デバイス定義を取得するには

次のget-device-definition例では、指定されたデバイス定義に関する情報を取得します。デバイス定義IDsの を取得するには、list-device-definitions コマンドを使用します。

```

aws greengrass get-device-definition \
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"

```

出力:

```

{
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/
versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
  "Name": "TemperatureSensors",
  "tags": {},
  "LastUpdatedTimestamp": "2019-09-11T00:19:03.698Z",
  "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",

```

```

    "CreationTimestamp": "2019-09-11T00:11:06.197Z",
    "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDeviceDefinition](#)」の「」を参照してください。

get-function-definition-version

次の例は、get-function-definition-version を使用する方法を説明しています。

AWS CLI

Lambda 関数の特定のバージョンに関する詳細を取得するには

以下は、指定された関数定義の指定されたバージョンに関する情報get-function-definition-versionを取得します。関数定義IDsのすべてのバージョンの を取得するには、list-function-definition-versions コマンドを使用します。関数定義に追加された最新バージョンの ID を取得するには、get-function-definition コマンドを使用してLatestVersionプロパティを確認します。

```

aws greengrass get-function-definition-version \
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85" \
  --function-definition-version-id "9748fda7-1589-4fcc-ac94-f5559e88678b"

```

出力:

```

{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/9748fda7-1589-4fcc-ac94-
f5559e88678b",
  "CreationTimestamp": "2019-06-18T17:04:30.776Z",
  "Definition": {
    "Functions": [
      {
        "FunctionArn": "arn:aws:lambda::function:GGIPDetector:1",
        "FunctionConfiguration": {
          "Environment": {},
          "MemorySize": 32768,
          "Pinned": true,

```



```
        "Timeout": 3
      },
      "Id": "26b69bdb-e547-46bc-9812-84ec04b6cc8c"
    },
    {
      "FunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",
      "FunctionConfiguration": {
        "EncodingType": "json",
        "Environment": {
          "Variables": {}
        },
        "MemorySize": 16384,
        "Pinned": true,
        "Timeout": 25
      },
      "Id": "384465a8-eebf-48c6-b793-4c35f7bfae9b"
    }
  ]
},
"Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
"Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFunctionDefinitionVersion](#)」の「」を参照してください。

get-function-definition

次のコード例は、get-function-definition を使用する方法を示しています。

AWS CLI

関数定義を取得するには

次のget-function-definition例では、指定された関数定義の詳細を表示します。関数定義IDsの を取得するには、list-function-definitions コマンドを使用します。

```
aws greengrass get-function-definition \
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

出力:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
  "CreationTimestamp": "2019-06-18T16:21:21.431Z",
  "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
  "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
  "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
  "tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFunctionDefinition](#)」の「」を参照してください。

get-group-certificate-authority

次の例は、get-group-certificate-authority を使用する方法を説明しています。

AWS CLI

Greengrass グループに関連付けられている CA を取得するには

次のget-group-certificate-authority例では、指定された Greengrass グループに関連付けられている認証局 (CA) を取得します。認証機関 ID を取得するには、list-group-certificate-authorities コマンドを使用してグループ ID を指定します。

```
aws greengrass get-group-certificate-authority \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
  --certificate-authority-
id "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
```

出力:

```
{
  "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
  "GroupCertificateAuthorityId":
  "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
```

```

    "PemEncodedCertificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBWEXAMPLEGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDEXAMPLEEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAEXAMPLESDB25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jEXAMPLENMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0EXAMPLEBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWEXAMPLEDASBgNVBA5TC01BTSDB25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWEXAMPLEGkqhkiG9w0BCQEWEG5vb25lQGFT
YXpvi5EXAMPLE8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLygVIk60CEXAMPLE93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswYEXAMPLEEgpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKEXAMPLEAQEFBQADgYEAtCu4
nUHVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n"
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroupCertificateAuthority](#)」の「」を参照してください。

get-group-certificate-configuration

次のコード例は、get-group-certificate-configuration を使用する方法を示しています。

AWS CLI

Greengrass グループが使用する認証機関の設定を取得するには

次のget-group-certificate-configuration例では、指定された Greengrass グループで使用される認証局 (CA) の設定を取得します。

```

aws greengrass get-group-certificate-configuration \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"

```

出力:

```

{
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
  "CertificateExpiryInMilliseconds": 604800000,
  "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"
}

```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroupCertificateConfiguration](#)」の「」を参照してください。

get-group-version

次の例は、get-group-version を使用する方法を説明しています。

AWS CLI

Greengrass グループのバージョンに関する情報を取得するには

次のget-group-version例では、指定されたグループの指定されたバージョンに関する情報を取得します。グループのIDsすべてのバージョンの を取得するには、list-group-versions コマンドを使用します。グループに最後に追加されたバージョンの ID を取得するには、get-group コマンドを使用して LatestVersionプロパティを確認します。

```
aws greengrass get-group-version \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \  
  --group-version-id "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-  
b77f-8741a4b00e5e",  
  "CreationTimestamp": "2019-06-18T17:04:30.915Z",  
  "Definition": {  
    "CoreDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-  
fd9d-4312-a8fd-ffa9404a20e0",  
    "FunctionDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/  
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",  
    "SubscriptionDefinitionVersionArn": "arn:aws:greengrass:us-  
west-2:123456789012:/greengrass/definition/subscriptions/70e49321-83d5-45d2-  
bc09-81f4917ae152/versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"  
  },  
  "Id": "1013db12-8b58-45ff-acc7-704248f66731",  
  "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroupVersion](#)」の「」を参照してください。

get-group

次のコード例は、get-group を使用する方法を示しています。

AWS CLI

Greengrass グループに関する情報を取得するには

次のget-group例では、指定された Greengrass グループに関する情報を取得します。グループの を取得するIDsには、list-groups コマンドを使用します。

```
aws greengrass get-group \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731",  
  "CreationTimestamp": "2019-06-18T16:21:21.457Z",  
  "Id": "1013db12-8b58-45ff-acc7-704248f66731",  
  "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",  
  "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-  
b77f-8741a4b00e5e",  
  "Name": "GGGroup4Pi3",  
  "tags": {}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroup](#)」の「」を参照してください。

get-logger-definition-version

次の例は、get-logger-definition-version を使用する方法を説明しています。

AWS CLI

ロガー定義のバージョンに関する情報を取得するには

次の`get-logger-definition-version`例では、指定されたロガー定義の指定されたバージョンに関する情報を取得します。ロガー定義IDsのすべてのバージョンの を取得するには、`list-logger-definition-versions` コマンドを使用します。ロガー定義に追加された最新バージョンの ID を取得するには、`get-logger-definition` コマンドを使用して `LatestVersion` プロパティを確認します。

```
aws greengrass get-logger-definition-version \  
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23" \  
  --logger-definition-version-id "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-  
a565-491e-8de0-3c0d8e0f2073",  
  "CreationTimestamp": "2019-05-08T16:10:13.866Z",  
  "Definition": {  
    "Loggers": []  
  },  
  "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",  
  "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLoggerDefinitionVersion](#)」の「」を参照してください。

get-logger-definition

次のコード例は、`get-logger-definition` を使用する方法を示しています。

AWS CLI

ロガー定義に関する情報を取得するには

次の`get-logger-definition`例では、指定されたロガー定義に関する情報を取得します。ロガー定義IDsの を取得するには、`list-logger-definitions` コマンドを使用します。

```
aws greengrass get-logger-definition \  
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",  
  "CreationTimestamp": "2019-05-08T16:10:13.809Z",  
  "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",  
  "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",  
  "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-  
a565-491e-8de0-3c0d8e0f2073",  
  "tags": {}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLoggerDefinition](#)」の「」を参照してください。

get-resource-definition-version

次のコード例は、get-resource-definition-version を使用する方法を示しています。

AWS CLI

リソース定義の特定のバージョンに関する情報を取得するには

次のget-resource-definition-version例では、指定されたリソース定義の指定されたバージョンに関する情報を取得します。リソース定義IDsのすべてのバージョンの を取得するには、list-resource-definition-versions コマンドを使用します。リソース定義に追加された最後のバージョンの ID を取得するには、get-resource-definition コマンドを使用して LatestVersion プロパティを確認します。

```
aws greengrass get-resource-definition-version \  
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658" \  
  --resource-definition-version-id "26e8829a-491a-464d-9c87-664bf6f6f2be"
```

出力:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
  "CreationTimestamp": "2019-06-19T16:40:59.392Z",
  "Definition": {
    "Resources": [
      {
        "Id": "26ff3f7b-839a-4217-9fdc-a218308b3963",
        "Name": "usb-port",
        "ResourceDataContainer": {
          "LocalDeviceResourceData": {
            "GroupOwnerSetting": {
              "AutoAddGroupOwner": false
            },
            "SourcePath": "/dev/bus/usb"
          }
        }
      }
    ]
  },
  "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
  "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceDefinitionVersion](#)」の「」を参照してください。

get-resource-definition

次の例は、get-resource-definition を使用する方法を説明しています。

AWS CLI

リソース定義に関する情報を取得するには

次のget-resource-definition例では、指定されたリソース定義に関する情報を取得します。リソース定義IDsの を取得するには、list-resource-definitions コマンドを使用します。

```
aws greengrass get-resource-definition \
```



```
--resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

出力:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
  "CreationTimestamp": "2019-06-19T16:40:59.261Z",
  "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
  "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
  "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
  "tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceDefinition](#)」の「」を参照してください。

get-service-role-for-account

次の例は、get-service-role-for-account を使用する方法を説明しています。

AWS CLI

アカウントにアタッチされているサービスロールの詳細を取得するには

次のget-service-role-for-account例では、AWS アカウントにアタッチされているサービスロールに関する情報を取得します。

```
aws greengrass get-service-role-for-account
```

出力:

```
{
  "AssociatedAt": "2018-10-18T15:59:20Z",
  "RoleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

詳細については、AWS IoT [Greengrass デベロッパーガイド](#) の「[Greengrass サービスロール](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceRoleForAccount](#)」の「」を参照してください。

get-subscription-definition-version

次のコード例は、get-subscription-definition-version を使用する方法を示しています。

AWS CLI

サブスクリプション定義の特定のバージョンに関する情報を取得するには

次のget-subscription-definition-version例では、指定されたサブスクリプション定義の指定されたバージョンに関する情報を取得します。サブスクリプション定義IDsのすべてのバージョンの を取得するには、list-subscription-definition-versions コマンドを使用します。サブスクリプション定義に追加された最新バージョンの ID を取得するには、get-subscription-definition コマンドを使用して LatestVersionプロパティを確認します。

```
aws greengrass get-subscription-definition-version \  
  --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152" \  
  --subscription-definition-version-id "88ae8699-12ac-4663-ba3f-4d7f0519140b"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/88ae8699-12ac-4663-  
ba3f-4d7f0519140b",  
  "CreationTimestamp": "2019-06-18T17:03:52.499Z",  
  "Definition": {  
    "Subscriptions": [  
      {  
        "Id": "692c4484-d89f-4f64-8edd-1a041a65e5b6",  
        "Source": "arn:aws:lambda:us-  
west-2:123456789012:function:Greengrass_HelloWorld:GG_HelloWorld",  
        "Subject": "hello/world",  
        "Target": "cloud"  
      }  
    ]  
  }  
}
```

```
  },  
  "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",  
  "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSubscriptionDefinitionVersion](#)」の「」を参照してください。

get-subscription-definition

次の例は、get-subscription-definition を使用する方法を説明しています。

AWS CLI

サブスクリプション定義に関する情報を取得するには

次のget-subscription-definition例では、指定されたサブスクリプション定義に関する情報を取得します。サブスクリプション定義IDsの を取得するには、list-subscription-definitions コマンドを使用します。

```
aws greengrass get-subscription-definition \  
--subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

出力:

```
{  
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/  
subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",  
  "CreationTimestamp": "2019-06-18T17:03:52.392Z",  
  "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",  
  "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",  
  "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",  
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/  
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",  
  "tags": {}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSubscriptionDefinition](#)」の「」を参照してください。

get-thing-runtime-configuration

次のコード例は、get-thing-runtime-configuration を使用する方法を示しています。

AWS CLI

Greengrass コアのランタイム設定を取得するには

次のget-thing-runtime-configuration例では、Greengrass コアのランタイム設定を取得します。ランタイム設定を取得する前に、update-thing-runtime-configuration コマンドを使用してコアのランタイム設定を作成する必要があります。

```
aws greengrass get-thing-runtime-configuration \  
  --thing-name SampleGreengrassCore
```

出力:

```
{  
  "RuntimeConfiguration": {  
    "TelemetryConfiguration": {  
      "ConfigurationSyncStatus": "OutOfSync",  
      "Telemetry": "On"  
    }  
  }  
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの「[テレメトリ設定の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetThingRuntimeConfiguration](#)」の「」を参照してください。

list-bulk-deployment-detailed-reports

次の例は、list-bulk-deployment-detailed-reports を使用する方法を説明しています。

AWS CLI

一括デプロイでの個々のデプロイに関する情報を一覧表示するには

次のlist-bulk-deployment-detailed-reports例では、ステータスを含む、一括デプロイオペレーションの個々のデプロイに関する情報を表示します。

```
aws greengrass list-bulk-deployment-detailed-reports \
  --bulk-deployment-id 42ce9c42-489b-4ed4-b905-8996aa50ef9d
```

出力:

```
{
  "Deployments": [
    {
      "DeploymentType": "NewDeployment",
      "DeploymentStatus": "Success",
      "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/versions/123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
      "CreatedAt": "2020-01-21T21:34:16.501Z"
    },
    {
      "DeploymentType": "NewDeployment",
      "DeploymentStatus": "InProgress",
      "DeploymentId": "123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/deployments/123456789012:123456789012:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-EXAMPLE55555/versions/a1b2c3d4-5678-90ab-cdef-EXAMPLE66666",
      "CreatedAt": "2020-01-21T21:34:16.486Z"
    },
    ...
  ]
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの [「グループの一括デプロイの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListBulkDeploymentDetailedReports](#)」の「」を参照してください。

list-bulk-deployments

次のコード例は、list-bulk-deployments を使用する方法を示しています。

AWS CLI

一括デプロイを一覧表示するには

次のlist-bulk-deployments例では、すべての一括デプロイを一覧表示します。

```
aws greengrass list-bulk-deployments
```

出力:

```
{
  "BulkDeployments": [
    {
      "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
      "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
      "CreatedAt": "2019-06-25T16:11:33.265Z"
    }
  ]
}
```

詳細については、AWS IoT Greengrass デベロッパーガイドの「[グループの一括デプロイの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBulkDeployments](#)」の「」を参照してください。

list-connector-definition-versions

次の例は、list-connector-definition-versions を使用する方法を説明しています。

AWS CLI

コネクタ定義で利用できるバージョンを一覧表示するには

次のlist-connector-definition-versions例では、指定されたコネクタ定義で利用できるバージョンを一覧表示します。list-connector-definitions コマンドを使用して、コネクタ定義 ID を取得します。

```
aws greengrass list-connector-definition-versions \  
--connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

出力:

```
{  
  "Versions": [  
    {  
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/versions/63c57963-  
c7c2-4a26-a7e2-7bf478ea2623",  
      "CreationTimestamp": "2019-06-19T19:30:01.300Z",  
      "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",  
      "Version": "63c57963-c7c2-4a26-a7e2-7bf478ea2623"  
    }  
  ]  
}
```

詳細については、AWS IoT Greengrass [デベロッパーガイドの「Greengrass コネクタを使用したサービスおよびプロトコルとの統合」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListConnectorDefinitionVersions](#)」の「」を参照してください。

list-connector-definitions

次の例は、list-connector-definitions を使用する方法を説明しています。

AWS CLI

定義されている Greengrass コネクタを一覧表示するには

次のlist-connector-definitions例では、AWS アカウントに定義されているすべての Greengrass コネクタを一覧表示します。

```
aws greengrass list-connector-definitions
```

出力:

```
{
```

```
"Definitions": [
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "CreationTimestamp": "2019-06-19T19:30:01.300Z",
    "Id": "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8",
    "LastUpdatedTimestamp": "2019-06-19T19:30:01.300Z",
    "LatestVersion": "63c57963-c7c2-4a26-a7e2-7bf478ea2623",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/connectors/b5c4ebfd-f672-49a3-83cd-31c7216a7bb8/
versions/63c57963-c7c2-4a26-a7e2-7bf478ea2623",
    "Name": "MySNSConnector"
  }
]
```

詳細については、AWS IoT Greengrass [デベロッパーガイドの「Greengrass コネクタを使用したサービスとプロトコルとの統合」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListConnectorDefinitions](#)」の「」を参照してください。

list-core-definition-versions

次の例は、list-core-definition-versions を使用する方法を説明しています。

AWS CLI

Greengrass コア定義のバージョンを一覧表示するには

次のlist-core-definitions例では、指定された Greengrass コア定義のすべてのバージョンを一覧表示します。list-core-definitions コマンドを使用してバージョン ID を取得できません。

```
aws greengrass list-core-definition-versions \
  --core-definition-id "eaf280cb-138c-4d15-af36-6f681a1348f7"
```

出力:

```
{
  "Versions": [
```



```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-c5da-440c-
a97b-084e62593b4c",
  "CreationTimestamp": "2019-06-18T16:14:17.709Z",
  "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
  "Version": "467c36e4-c5da-440c-a97b-084e62593b4c"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCoreDefinitionVersions](#)」の「」を参照してください。

list-core-definitions

次の例は、list-core-definitions を使用する方法を説明しています。

AWS CLI

Greengrass コア定義を一覧表示するには

次のlist-core-definitions例では、AWS アカウントの Greengrass コア定義をすべて一覧表示します。

```
aws greengrass list-core-definitions
```

出力:

```
{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d",
      "CreationTimestamp": "2018-10-17T04:30:32.786Z",
      "Id": "0507843c-c1ef-4f06-b051-817030df7e7d",
      "LastUpdatedTimestamp": "2018-10-17T04:30:32.786Z",
      "LatestVersion": "bcdf9e86-3793-491e-93af-3cdfbf4e22b7",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d/versions/
bcdf9e86-3793-491e-93af-3cdfbf4e22b7"
    }
  ]
}
```

```

    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/31c22500-3509-4271-bafd-cf0655cda438",
      "CreationTimestamp": "2019-06-18T16:24:16.064Z",
      "Id": "31c22500-3509-4271-bafd-cf0655cda438",
      "LastUpdatedTimestamp": "2019-06-18T16:24:16.064Z",
      "LatestVersion": "2f350395-6d09-4c8a-8336-9ae5b57ace84",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/31c22500-3509-4271-bafd-cf0655cda438/
versions/2f350395-6d09-4c8a-8336-9ae5b57ace84"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46",
      "CreationTimestamp": "2019-06-18T16:21:21.351Z",
      "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
      "LastUpdatedTimestamp": "2019-06-18T16:21:21.351Z",
      "LatestVersion": "42aeac3-fd9d-4312-a8fd-ffa9404a20e0",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeac3-
fd9d-4312-a8fd-ffa9404a20e0"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7",
      "CreationTimestamp": "2019-06-18T16:14:17.709Z",
      "Id": "eaf280cb-138c-4d15-af36-6f681a1348f7",
      "LastUpdatedTimestamp": "2019-06-18T16:14:17.709Z",
      "LatestVersion": "467c36e4-c5da-440c-a97b-084e62593b4c",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/eaf280cb-138c-4d15-af36-6f681a1348f7/versions/467c36e4-
c5da-440c-a97b-084e62593b4c"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListCoreDefinitions](#)」の「」を参照してください。

list-deployments

次のコード例は、list-deployments を使用する方法を示しています。

AWS CLI

Greengrass グループのデプロイを一覧表示するには

次のlist-deployments例では、指定された Greengrass グループのデプロイを一覧表示します。list-groups コマンドを使用して、グループ ID を検索できます。

```
aws greengrass list-deployments \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

出力:

```
{  
  "Deployments": [  
    {  
      "CreatedAt": "2019-06-18T17:04:32.702Z",  
      "DeploymentId": "1065b8a0-812b-4f21-9d5d-e89b232a530f",  
      "DeploymentType": "NewDeployment",  
      "GroupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-  
b77f-8741a4b00e5e"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeployments](#)」の「」を参照してください。

list-device-definition-versions

次のコード例は、list-device-definition-versions を使用する方法を示しています。

AWS CLI

デバイス定義のバージョンを一覧表示するには

次のlist-device-definition-versions例は、指定されたデバイス定義に関連付けられたデバイス定義バージョンを表示します。

```
aws greengrass list-device-definition-versions \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

出力:

```
{
  "Versions": [
    {
      "Version": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
      "CreationTimestamp": "2019-09-11T00:15:09.838Z",
      "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/83c13984-6fed-447e-84d5-5b8aa45d5f71"
    },
    {
      "Version": "3b5cc510-58c1-44b5-9d98-4ad858ffa795",
      "CreationTimestamp": "2019-09-11T00:11:06.197Z",
      "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/3b5cc510-58c1-44b5-9d98-4ad858ffa795"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeviceDefinitionVersions](#)」の「」を参照してください。

list-device-definitions

次の例は、list-device-definitions を使用する方法を説明しています。

AWS CLI

デバイス定義を一覧表示するには

次のlist-device-definitions例では、指定された AWS リージョンの AWS アカウント内のデバイス定義の詳細を表示します。

```
aws greengrass list-device-definitions \
  --region us-west-2
```

出力:

```
{
  "Definitions": [
    {
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab/versions/c777b0f5-1059-449b-beaa-f003ebc56c34",
      "LastUpdatedTimestamp": "2019-06-14T15:42:09.059Z",
      "LatestVersion": "c777b0f5-1059-449b-beaa-f003ebc56c34",
      "CreationTimestamp": "2019-06-14T15:42:09.059Z",
      "Id": "50f3274c-3f0a-4f57-b114-6f46085281ab",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/50f3274c-3f0a-4f57-b114-6f46085281ab"
    },
    {
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/e01951c9-6134-479a-969a-1a15cac11c40/versions/514d57aa-4ee6-401c-9fac-938a9f7a51e5",
      "Name": "TestDeviceDefinition",
      "LastUpdatedTimestamp": "2019-04-16T23:17:43.245Z",
      "LatestVersion": "514d57aa-4ee6-401c-9fac-938a9f7a51e5",
      "CreationTimestamp": "2019-04-16T23:17:43.245Z",
      "Id": "e01951c9-6134-479a-969a-1a15cac11c40",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/e01951c9-6134-479a-969a-1a15cac11c40"
    },
    {
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd/versions/83c13984-6fed-447e-84d5-5b8aa45d5f71",
      "Name": "TemperatureSensors",
      "LastUpdatedTimestamp": "2019-09-10T00:19:03.698Z",
      "LatestVersion": "83c13984-6fed-447e-84d5-5b8aa45d5f71",
      "CreationTimestamp": "2019-09-11T00:11:06.197Z",
      "Id": "f9ba083d-5ad4-4534-9f86-026a45df1ccd",
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/devices/f9ba083d-5ad4-4534-9f86-026a45df1ccd"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeviceDefinitions](#)」の「」を参照してください。

list-function-definition-versions

次の例は、list-function-definition-versions を使用する方法を説明しています。

AWS CLI

Lambda 関数のバージョンを一覧表示するには

次のlist-function-definition-versions例では、指定された Lambda 関数のすべてのバージョンを一覧表示します。list-function-definitions コマンドを使用して ID を取得できます。

```
aws greengrass list-function-definition-versions \  
--function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

出力:

```
{  
  "Versions": [  
    {  
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/  
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",  
      "CreationTimestamp": "2019-06-18T17:04:30.776Z",  
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",  
      "Version": "9748fda7-1589-4fcc-ac94-f5559e88678b"  
    },  
    {  
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/  
versions/9b08df77-26f2-4c29-93d2-769715edcfec",  
      "CreationTimestamp": "2019-06-18T17:02:44.087Z",  
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",  
      "Version": "9b08df77-26f2-4c29-93d2-769715edcfec"  
    },  
    {  
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/  
versions/4236239f-94f7-4b90-a2f8-2a24c829d21e",  
      "CreationTimestamp": "2019-06-18T17:01:42.284Z",  
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",  
      "Version": "4236239f-94f7-4b90-a2f8-2a24c829d21e"  
    }  
  ]  
}
```

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/versions/343408bb-549a-4fbe-b043-853643179a39",
  "CreationTimestamp": "2019-06-18T16:21:21.431Z",
  "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
  "Version": "343408bb-549a-4fbe-b043-853643179a39"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFunctionDefinitionVersions](#)」の「」を参照してください。

list-function-definitions

次の例は、list-function-definitions を使用する方法を説明しています。

AWS CLI

Lambda 関数を一覧表示するには

次のlist-function-definitions例では、AWS アカウントに定義されているすべての Lambda 関数を一覧表示します。

```
aws greengrass list-function-definitions
```

出力:

```
{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960",
      "CreationTimestamp": "2018-10-17T04:30:32.884Z",
      "Id": "017970a5-8952-46dd-b1c1-020b3ae8e960",
      "LastUpdatedTimestamp": "2018-10-17T04:30:32.884Z",
      "LatestVersion": "4380b302-790d-4ed8-92bf-02e88afecb15",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/functions/017970a5-8952-46dd-b1c1-020b3ae8e960/versions/4380b302-790d-4ed8-92bf-02e88afecb15"
    }
  ]
}
```

```

    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "CreationTimestamp": "2019-06-18T16:21:21.431Z",
      "Id": "063f5d1a-1dd1-40b4-9b51-56f8993d0f85",
      "LastUpdatedTimestamp": "2019-06-18T16:21:21.431Z",
      "LatestVersion": "9748fda7-1589-4fcc-ac94-f5559e88678b",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/6598e653-a262-440c-9967-e2697f64da7b",
      "CreationTimestamp": "2019-06-18T16:24:16.123Z",
      "Id": "6598e653-a262-440c-9967-e2697f64da7b",
      "LastUpdatedTimestamp": "2019-06-18T16:24:16.123Z",
      "LatestVersion": "38bc6ccd-98a2-4ce7-997e-16c84748fae4",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/6598e653-a262-440c-9967-e2697f64da7b/
versions/38bc6ccd-98a2-4ce7-997e-16c84748fae4"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/functions/c668df84-fad2-491b-95f4-655d2cad7885",
      "CreationTimestamp": "2019-06-18T16:14:17.784Z",
      "Id": "c668df84-fad2-491b-95f4-655d2cad7885",
      "LastUpdatedTimestamp": "2019-06-18T16:14:17.784Z",
      "LatestVersion": "37dd68c4-a64f-40ba-aa13-71fecc3ebded",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/c668df84-fad2-491b-95f4-655d2cad7885/
versions/37dd68c4-a64f-40ba-aa13-71fecc3ebded"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListFunctionDefinitions](#)」の「」を参照してください。

list-group-certificate-authorities

次のコード例は、list-group-certificate-authorities を使用する方法を示しています。

AWS CLI

グループの現在の CAsを一覧表示するには

次のlist-group-certificate-authorities例では、指定された Greengrass グループの現在の認証機関 (CAs) を一覧表示します。

```
aws greengrass list-group-certificate-authorities \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

出力:

```
{  
  "GroupCertificateAuthorities": [  
    {  
      "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-  
west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/  
certificateauthorities/  
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",  
      "GroupCertificateAuthorityId":  
      "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGroupCertificateAuthorities](#)」の「」を参照してください。

list-group-versions

次の例は、list-group-versions を使用する方法を説明しています。

AWS CLI

Greengrass グループのバージョンを一覧表示するには

次のlist-group-versions例では、指定された Greengrass グループのバージョンを一覧表示します。

```
aws greengrass list-group-versions \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

出力:

```
{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
      "CreationTimestamp": "2019-06-18T17:04:30.915Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/4340669d-
d14d-44e3-920c-46c928750750",
      "CreationTimestamp": "2019-06-18T17:03:52.663Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "4340669d-d14d-44e3-920c-46c928750750"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
versions/1b06e099-2d5b-4f10-91b9-78c4e060f5da",
      "CreationTimestamp": "2019-06-18T17:02:44.189Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "1b06e099-2d5b-4f10-91b9-78c4e060f5da"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/2d3f27f1-3b43-4554-
ab7a-73ec30477efe",
      "CreationTimestamp": "2019-06-18T17:01:42.401Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "2d3f27f1-3b43-4554-ab7a-73ec30477efe"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/d20f7ae9-3444-4c1c-b025-
e2ede23cdd31",
      "CreationTimestamp": "2019-06-18T16:21:21.457Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "Version": "d20f7ae9-3444-4c1c-b025-e2ede23cdd31"
    }
  ]
}
```

```
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGroupVersions](#)」の「」を参照してください。

list-groups

次の例は、list-groups を使用する方法を説明しています。

AWS CLI

Greengrass グループを一覧表示するには

次のlist-groups例では、AWS アカウントで定義されているすべての Greengrass グループを一覧表示します。

```
aws greengrass list-groups
```

出力:

```
{
  "Groups": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731",
      "CreationTimestamp": "2019-06-18T16:21:21.457Z",
      "Id": "1013db12-8b58-45ff-acc7-704248f66731",
      "LastUpdatedTimestamp": "2019-06-18T16:21:21.457Z",
      "LatestVersion": "115136b3-cfd7-4462-b77f-8741a4b00e5e",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-b77f-8741a4b00e5e",
      "Name": "GGGroup4Pi3"
    },
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
      "CreationTimestamp": "2018-10-31T21:52:46.603Z",
      "Id": "1402daf9-71cf-4cfe-8be0-d5e80526d0d8",
      "LastUpdatedTimestamp": "2018-10-31T21:52:46.603Z",

```

```

    "LatestVersion": "749af901-60ab-456f-a096-91b12d983c29",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/versions/749af901-60ab-456f-
a096-91b12d983c29",
    "Name": "MyTestGroup"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/504b5c8d-bbed-4635-aff1-48ec5b586db5",
    "CreationTimestamp": "2018-12-31T21:39:36.771Z",
    "Id": "504b5c8d-bbed-4635-aff1-48ec5b586db5",
    "LastUpdatedTimestamp": "2018-12-31T21:39:36.771Z",
    "LatestVersion": "46911e8e-f9bc-4898-8b63-59c7653636ec",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/504b5c8d-bbed-4635-aff1-48ec5b586db5/versions/46911e8e-
f9bc-4898-8b63-59c7653636ec",
    "Name": "smp-ggrass-group"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGroups](#)」の「」を参照してください。

list-logger-definition-versions

次のコード例は、list-logger-definition-versions を使用する方法を示しています。

AWS CLI

ロガー定義のバージョンのリストを取得するには

次のlist-logger-definition-versions例では、指定されたロガー定義のすべてのバージョンのリストを取得します。

```

aws greengrass list-logger-definition-versions \
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"

```

出力:

```
{
```

```
"Versions": [
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/5e3f6f64-
a565-491e-8de0-3c0d8e0f2073",
    "CreationTimestamp": "2019-05-08T16:10:13.866Z",
    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "Version": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/versions/3ec6d3af-eb85-48f9-
a16d-1c795fe696d7",
    "CreationTimestamp": "2019-05-08T16:10:13.809Z",
    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "Version": "3ec6d3af-eb85-48f9-a16d-1c795fe696d7"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLoggerDefinitionVersions](#)」の「」を参照してください。

list-logger-definitions

次の例は、list-logger-definitions を使用する方法を説明しています。

AWS CLI

ロガー定義のリストを取得するには

次のlist-logger-definitions例では、AWS アカウントのすべてのロガー定義を一覧表示します。

```
aws greengrass list-logger-definitions
```

出力:

```
{
  "Definitions": [
    {
```

```

    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "CreationTimestamp": "2019-05-08T16:10:13.809Z",
    "Id": "49eeeb66-f1d3-4e34-86e3-3617262abf23",
    "LastUpdatedTimestamp": "2019-05-08T16:10:13.809Z",
    "LatestVersion": "5e3f6f64-a565-491e-8de0-3c0d8e0f2073",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/loggers/49eeeb66-f1d3-4e34-86e3-3617262abf23/
versions/5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLoggerDefinitions](#)」の「」を参照してください。

list-resource-definition-versions

次のコード例は、list-resource-definition-versions を使用する方法を示しています。

AWS CLI

リソース定義のバージョンを一覧表示するには

次のlist-resource-definition-versions例では、指定された Greengrass リソースのバージョンを一覧表示します。

```

aws greengrass list-resource-definition-versions \
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"

```

出力:

```

{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/
versions/26e8829a-491a-464d-9c87-664bf6f6f2be",
      "CreationTimestamp": "2019-06-19T16:40:59.392Z",
      "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
      "Version": "26e8829a-491a-464d-9c87-664bf6f6f2be"
    },
  ],
}

```

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/versions/432d92f6-12de-4ec9-a704-619a942a62aa",
  "CreationTimestamp": "2019-06-19T16:40:59.261Z",
  "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
  "Version": "432d92f6-12de-4ec9-a704-619a942a62aa"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceDefinitionVersions](#)」の「」を参照してください。

list-resource-definitions

次のコード例は、list-resource-definitions を使用する方法を示しています。

AWS CLI

定義されているリソースを一覧表示するには

次のlist-resource-definitions例では、AWS IoT Greengrass が使用するように定義されているリソースを一覧表示します。

```
aws greengrass list-resource-definitions
```

出力:

```
{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658",
      "CreationTimestamp": "2019-06-19T16:40:59.261Z",
      "Id": "ad8c101d-8109-4b0e-b97d-9cc5802ab658",
      "LastUpdatedTimestamp": "2019-06-19T16:40:59.261Z",
      "LatestVersion": "26e8829a-491a-464d-9c87-664bf6f6f2be",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658/versions/26e8829a-491a-464d-9c87-664bf6f6f2be"
    },
  ],
}
```

```

    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
      "CreationTimestamp": "2019-06-19T21:51:28.212Z",
      "Id": "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38",
      "LastUpdatedTimestamp": "2019-06-19T21:51:28.212Z",
      "LatestVersion": "a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
      "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/resources/c8bb9ebc-c3fd-40a4-9c6a-568d75569d38/versions/
a5f94d0b-f6bc-40f4-bb78-7a1c5fe13ba1",
      "Name": "MyGreengrassResources"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceDefinitions](#)」の「」を参照してください。

list-subscription-definition-versions

次の例は、list-subscription-definition-versions を使用する方法を説明しています。

AWS CLI

サブスクリプション定義のバージョンを一覧表示するには

次のlist-subscription-definition-versions例では、指定されたサブスクリプションのすべてのバージョンを一覧表示します。list-subscription-definitions コマンドを使用して、サブスクリプション ID を検索できます。

```

aws greengrass list-subscription-definition-versions \
  --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"

```

出力:

```

{
  "Versions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/
versions/88ae8699-12ac-4663-ba3f-4d7f0519140b",

```



```
    "CreationTimestamp": "2019-06-18T17:03:52.499Z",
    "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
    "Version": "88ae8699-12ac-4663-ba3f-4d7f0519140b"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/7e320ba3-
c369-4069-a2f0-90acb7f219d6",
    "CreationTimestamp": "2019-06-18T17:03:52.392Z",
    "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
    "Version": "7e320ba3-c369-4069-a2f0-90acb7f219d6"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSubscriptionDefinitionVersions](#)」の「」を参照してください。

list-subscription-definitions

次のコード例は、list-subscription-definitions を使用する方法を示しています。

AWS CLI

リストサブスクリプション定義を取得するには

次のlist-subscription-definitions例では、AWS アカウントで定義されている AWS IoT Greengrass サブスクリプションをすべて一覧表示します。

```
aws greengrass list-subscription-definitions
```

出力:

```
{
  "Definitions": [
    {
      "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152",
      "CreationTimestamp": "2019-06-18T17:03:52.392Z",
      "Id": "70e49321-83d5-45d2-bc09-81f4917ae152",
      "LastUpdatedTimestamp": "2019-06-18T17:03:52.392Z",
```

```

    "LatestVersion": "88ae8699-12ac-4663-ba3f-4d7f0519140b",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/subscriptions/70e49321-83d5-45d2-bc09-81f4917ae152/versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967",
    "CreationTimestamp": "2018-10-18T15:45:34.024Z",
    "Id": "cd6f1c37-d9a4-4e90-be94-01a7404f5967",
    "LastUpdatedTimestamp": "2018-10-18T15:45:34.024Z",
    "LatestVersion": "d1cf8fac-284f-4f6a-98fe-a2d36d089373",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/subscriptions/cd6f1c37-d9a4-4e90-be94-01a7404f5967/versions/d1cf8fac-284f-4f6a-98fe-a2d36d089373"
  },
  {
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b",
    "CreationTimestamp": "2018-10-22T17:09:31.429Z",
    "Id": "fa81bc84-3f59-4377-a84b-5d0134da359b",
    "LastUpdatedTimestamp": "2018-10-22T17:09:31.429Z",
    "LatestVersion": "086d1b08-b25a-477c-a16f-6f9b3a9c295a",
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/subscriptions/fa81bc84-3f59-4377-a84b-5d0134da359b/versions/086d1b08-b25a-477c-a16f-6f9b3a9c295a"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSubscriptionDefinitions](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースにアタッチされたタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたリソースにアタッチされているタグとその値を一覧表示します。

```
aws greengrass list-tags-for-resource \  
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
  definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

出力:

```
{  
  "tags": {  
    "ResourceSubType": "USB",  
    "ResourceType": "Device"  
  }  
}
```

詳細については、AWS IoT [Greengrass デベロッパーガイドの「Greengrass リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

reset-deployments

次のコード例は、reset-deployments を使用する方法を示しています。

AWS CLI

Greengrass グループのデプロイ情報をクリーンアップするには

次のreset-deployments例では、指定された Greengrass グループのデプロイ情報をクリーンアップします。を追加すると--force option、コアデバイスが応答するのを待たずにデプロイ情報がリセットされます。

```
aws greengrass reset-deployments \  
  --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \  
  --force
```

出力:

```
{  
  "DeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/  
  greengrass/groups/1402daf9-71cf-4cfe-8be0-d5e80526d0d8/  
  deployments/7dd4e356-9882-46a3-9e28-6d21900c011a",
```

```
"DeploymentId": "7dd4e356-9882-46a3-9e28-6d21900c011a"
}
```

詳細については、[「IoT Greengrass デベロッパーガイド」の「デプロイのリセット」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetDeployments](#)」の「」を参照してください。

start-bulk-deployment

次の例は、start-bulk-deployment を使用する方法を説明しています。

AWS CLI

一括デプロイオペレーションを開始するには

次のstart-bulk-deployment例では、S3 バケットに保存されているファイルを使用して、一括デプロイオペレーションを開始し、デプロイするグループを指定します。

```
aws greengrass start-bulk-deployment \
  --cli-input-json "{\"InputFileUri\":\"https://gg-group-deployment1.s3-us-west-2.amazonaws.com/MyBulkDeploymentInputFile.txt\", \"ExecutionRoleArn\": \"arn:aws:iam::123456789012:role/ggCreateDeploymentRole\", \"AmznClientToken\": \"yourAmazonClientToken\"}"
```

出力:

```
{
  "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
  "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
}
```

詳細については、IoT Greengrass デベロッパーガイドの[「グループの一括デプロイの作成」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[StartBulkDeployment](#)」の「」を参照してください。

stop-bulk-deployment

次の例は、stop-bulk-deployment を使用する方法を説明しています。

AWS CLI

一括デプロイを停止するには

次のstop-bulk-deployment例では、指定された一括デプロイを停止します。完了した一括デプロイを停止しようとする、エラーが表示されます。InvalidInputException: Cannot change state of finished execution.

```
aws greengrass stop-bulk-deployment \  
  --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

このコマンドでは何も出力されません。

詳細については、[「IoT Greengrass デベロッパーガイド」の「グループの一括デプロイの作成」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[StopBulkDeployment](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを適用するには

次のtag-resource例では、指定された Greengrass リソースに ResourceSubType2 つのタグ ResourceTypeと を適用します。このオペレーションでは、新しいタグと値を追加したり、既存のタグの値を更新したりできます。untag-resource コマンドを使用してタグを削除します。

```
aws greengrass tag-resource \  
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
  definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658" \  
  --tags "ResourceType=Device,ResourceSubType=USB"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT [Greengrass デベロッパーガイドの「Greengrass リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグとその値を削除するには

次のuntag-resource例では、キーが指定された Greengrass グループCategoryにあるタグを削除します。指定されたリソースに キーCategoryが存在しない場合、エラーは返されません。

```
aws greengrass untag-resource \  
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731" \  
  --tag-keys "Category"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT [Greengrass デベロッパーガイドの「Greengrass リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-connectivity-info

次のコード例は、update-connectivity-info を使用する方法を示しています。

AWS CLI

Greengrass コアの接続情報を更新するには

次のupdate-connectivity-info例では、デバイスが指定された Greengrass コアへの接続に使用できるエンドポイントを変更します。接続情報は、IP アドレスまたはドメイン名のリストで、対応するポート番号とオプションのカスタマー定義メタデータが含まれます。ローカルネットワークが変更されると、接続情報を更新する必要がある場合があります。

```
aws greengrass update-connectivity-info \  
  --thing-name "MyGroup_Core" \  
  --connectivity-info "[{"Metadata\":"\", \"PortNumber\":8883, \"HostAddress\":"127.0.0.1\", \"Id\":"localhost_127.0.0.1_0\"}, {"Metadata\":"\", \"PortNumber\":"8883, \"HostAddress\":"192.168.1.3\", \"Id\":"localIP_192.168.1.3\"}]"
```

出力:

```
{  
  "Version": "312de337-59af-4cf9-a278-2a23bd39c300"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateConnectivityInfo](#)」の「」を参照してください。

update-connector-definition

次の例は、update-connector-definition を使用する方法を説明しています。

AWS CLI

コネクタ定義の名前を更新するには

次のupdate-connector-definition例では、指定されたコネクタ定義の名前を更新します。コネクタの詳細を更新する場合は、create-connector-definition-version コマンドを使用して新しいバージョンを作成します。

```
aws greengrass update-connector-definition \  
  --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \  
  --name "GreengrassConnectors2019"
```

詳細については、AWS IoT Greengrass デベロッパーガイドの「[コネクタを使用した サービスとプロトコルとの統合](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateConnectorDefinition](#)」の「」を参照してください。

update-core-definition

次の例は、update-core-definition を使用する方法を説明しています。

AWS CLI

コア定義を更新するには

次のupdate-core-definition例では、指定されたコア定義の名前を変更します。更新できるのは、コア定義の name プロパティのみです。

```
aws greengrass update-core-definition \  
  --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \  
  --name "MyCoreDevices"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「IoT Greengrass Core の設定」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCoreDefinition](#)」の「」を参照してください。

update-device-definition

次の例は、update-device-definition を使用する方法を説明しています。

AWS CLI

デバイス定義を更新するには

次のupdate-device-definition例では、指定されたデバイス定義の名前を変更します。デバイス定義の name プロパティのみを更新できます。

```
aws greengrass update-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \  
  --name "TemperatureSensors"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDeviceDefinition](#)」の「」を参照してください。

update-function-definition

次の例は、update-function-definition を使用する方法を説明しています。

AWS CLI

関数定義の名前を更新するには

次のupdate-function-definition例では、指定された関数定義の名前を更新します。関数の詳細を更新する場合は、create-function-definition-version コマンドを使用して新しいバージョンを作成します。

```
aws greengrass update-function-definition \  
  --function-definition-id "e47952bd-dea9-4e2c-a7e1-37bbe8807f46" \  
  --name ObsoleteFunction
```

このコマンドでは何も出力されません。

詳細については、IoT Greengrass デベロッパーガイドの「[ローカル Lambda 関数の実行](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFunctionDefinition](#)」の「」を参照してください。

update-group-certificate-configuration

次の例は、update-group-certificate-configuration を使用する方法を説明しています。

AWS CLI

グループの証明書の有効期限を更新するには

次のupdate-group-certificate-configuration例では、指定されたグループに生成された証明書の 10 日間の有効期限を設定します。

```
aws greengrass update-group-certificate-configuration \  
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1" \  
  --certificate-expiry-in-milliseconds 864000000
```

出力:

```
{  
  "CertificateExpiryInMilliseconds": 864000000,  
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,  
  "GroupId": "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"  
}
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)「AWS 」のIoT Greengrass セキュリティ」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateGroupCertificateConfiguration](#)」の「」を参照してください。

update-group

次のコード例は、update-group を使用する方法を示しています。

AWS CLI

グループ名を更新するには

次のupdate-group例では、指定された Greengrass グループの名前を更新します。グループの詳細を更新する場合は、create-group-version コマンドを使用して新しいバージョンを作成します。

```
aws greengrass update-group \  
  --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \  
  --name TestGroup4of6
```

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)」の AWS IoT Greengrass onIoT の設定」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateGroup](#)」の「」を参照してください。

update-logger-definition

次の例は、update-logger-definition を使用する方法を説明しています。

AWS CLI

ロガ一定義を更新するには

次のupdate-logger-definition例では、指定されたロガ一定義の名前を変更します。更新できるのは、ロガ一定義の name プロパティのみです。

```
aws greengrass update-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \  
  --name "LoggingConfigsForSensors"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT Greengrass デベロッパーガイド](#)の「[IoT Greengrass Logs によるモニタリング](#)」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLoggerDefinition](#)」の「」を参照してください。

update-resource-definition

次の例は、update-resource-definition を使用する方法を説明しています。

AWS CLI

リソース定義の名前を更新するには

次のupdate-resource-definition例では、指定されたリソース定義の名前を更新します。リソースの詳細を変更する場合は、create-resource-definition-version コマンドを使用して新しいバージョンを作成します。

```
aws greengrass update-resource-definition \  
  --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \  
  --name GreengrassConnectorResources
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Greengrass デベロッパーガイドの「[Lambda 関数とコネクタを使用してローカルリソースにアクセスする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResourceDefinition](#)」の「」を参照してください。

update-subscription-definition

次のコード例は、update-subscription-definition を使用する方法を示しています。

AWS CLI

サブスクリプション定義の名前を更新するには

次のupdate-subscription-definition例では、指定されたサブスクリプション定義の名前を更新します。サブスクリプションの詳細を変更する場合は、create-subscription-definition-version コマンドを使用して新しいバージョンを作成します。

```
aws greengrass update-subscription-definition \  
  --subscription-definition-id "fa81bc84-3f59-4377-a84b-5d0134da359b" \  
  --name "ObsoleteSubscription"
```

このコマンドでは何も出力されません。

詳細については、ガイドのタイトルを参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSubscriptionDefinition](#)」の「」を参照してください。

update-thing-runtime-configuration

次の例は、update-thing-runtime-configuration を使用する方法を説明しています。

AWS CLI

Greengrass コアのランタイム設定でテレメトリを有効にするには

次のupdate-thing-runtime-configuration例では、Greengrass コアのランタイム設定を更新してテレメトリを有効にします。

```
aws greengrass update-thing-runtime-configuration \  
  --thing-name SampleGreengrassCore \  
  --telemetry-configuration {"Telemetry\":"On\"}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Greengrass デベロッパーガイド」の「[テレメトリ設定の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateThingRuntimeConfiguration](#)」の「」を参照してください。

AWS IoT Greengrass V2 を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Greengrass V2。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-service-role-to-account

次のコード例は、`associate-service-role-to-account` を使用する方法を示しています。

AWS CLI

Greengrass サービスロールを AWS アカウントに関連付けるには

次の`associate-service-role-to-account`例では、AWS サービスロールをアカウントの AWS IoT Greengrass に関連付けます。

```
aws greengrassv2 associate-service-role-to-account \  
  --role-arn arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole
```

出力:

```
{  
  "associatedAt": "2022-01-19T19:21:53Z"  
}
```

詳細については、AWS IoT V2 [Greengrass V2 デベロッパーガイドの「Greengrass サービスロール」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateServiceRoleToAccount](#)」の「」を参照してください。

batch-associate-client-device-with-core-device

次のコード例は、batch-associate-client-device-with-core-device を使用する方法を示しています。

AWS CLI

クライアントデバイスをコアデバイスに関連付けるには

次のbatch-associate-client-device-with-core-device例では、2つのクライアントデバイスを1つのコアデバイスに関連付けます。

```
aws greengrassv2 batch-associate-client-device-with-core-device \  
  --core-device-thing-name MyGreengrassCore \  
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

出力:

```
{  
  "errorEntries": []  
}
```

詳細については、[IoT Greengrass V2 デベロッパーガイドの「ローカル IoT デバイスの操作」](#)を参照してください。AWS IoT V2

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchAssociateClientDeviceWithCoreDevice](#)」の「」を参照してください。

batch-disassociate-client-device-from-core-device

次のコード例は、batch-disassociate-client-device-from-core-device を使用する方法を示しています。

AWS CLI

コアデバイスからクライアントデバイスの関連付けを解除するには

次のbatch-disassociate-client-device-from-core-device例では、2つのクライアントデバイスをコアデバイスから関連付け解除します。

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \  
  --core-device-thing-name MyGreengrassCore \  
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

出力:

```
{  
  "errorEntries": []  
}
```

詳細については、[IoT Greengrass V2 デベロッパーガイドの「ローカル IoT デバイスの操作AWS IoT V2」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDisassociateClientDeviceFromCoreDevice](#)」の「」を参照してください。

cancel-deployment

次のコード例は、cancel-deployment を使用する方法を示しています。

AWS CLI

デプロイをキャンセルするには

次のcancel-deployment例では、モノのグループへの継続的なデプロイを停止します。

```
aws greengrassv2 cancel-deployment \  
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "message": "SUCCESS"  
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイドの「[デプロイをキャンセルする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelDeployment](#)」の「」を参照してください。

create-component-version

次の例は、create-component-version を使用する方法を説明しています。

AWS CLI

例 1: レシピからコンポーネントバージョンを作成するには

次のcreate-component-version例では、レシピファイルから Hello World コンポーネントのバージョンを作成します。

```
aws greengrassv2 create-component-version \  
  --inline-recipe fileb://com.example.HelloWorld-1.0.0.json
```

com.example.HelloWorld-1.0.0.json の内容:

```
{  
  "RecipeFormatVersion": "2020-01-25",  
  "ComponentName": "com.example.HelloWorld",  
  "ComponentVersion": "1.0.0",  
  "ComponentDescription": "My first AWS IoT Greengrass component.",  
  "ComponentPublisher": "Amazon",  
  "ComponentConfiguration": {  
    "DefaultConfiguration": {  
      "Message": "world"  
    }  
  },  
  "Manifests": [  
    {  
      "Platform": {  
        "os": "linux"  
      },  
      "Lifecycle": {  
        "Run": "echo 'Hello {configuration:/Message}'"  
      }  
    }  
  ]  
}
```


出力:

```
{
  "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
  "componentName": "com.example.HelloWorld",
  "componentVersion": "1.0.0",
  "creationTimestamp": "2021-01-07T16:24:33.650000-08:00",
  "status": {
    "componentState": "REQUESTED",
    "message": "NONE",
    "errors": {}
  }
}
```

詳細については、IoT Greengrass V2 デベロッパーガイドの「デプロイする [カスタムコンポーネントの作成](#)」および「コンポーネントのアップロード」を参照してください。 <https://docs.aws.amazon.com/greengrass/v2/developerguide/upload-components.html> AWS IoT V2

例 2: AWS Lambda 関数からコンポーネントバージョンを作成するには

次の `create-component-version` 例では、AWS Lambda 関数から Hello World コンポーネントのバージョンを作成します。

```
aws greengrassv2 create-component-version \
  --cli-input-json file://lambda-function-component.json
```

`lambda-function-component.json` の内容:

```
{
  "lambdaFunction": {
    "lambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:HelloWorldPythonLambda:1",
    "componentName": "com.example.HelloWorld",
    "componentVersion": "1.0.0",
    "componentLambdaParameters": {
      "eventSources": [
        {
          "topic": "hello/world/+",
          "type": "IOT_CORE"
        }
      ]
    }
  }
}
```

```
    ]
  }
}
```

出力:

```
{
  "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0",
  "componentName": "com.example.HelloWorld",
  "componentVersion": "1.0.0",
  "creationTimestamp": "2021-01-07T17:05:27.347000-08:00",
  "status": {
    "componentState": "REQUESTED",
    "message": "NONE",
    "errors": {}
  }
}
```

詳細については、IoT Greengrass V2 デベロッパーガイドの[AWS 「Lambda 関数の実行」](#)を参照してください。AWS IoT V2

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateComponentVersion](#)」の「」を参照してください。

create-deployment

次の例は、create-deployment を使用する方法を説明しています。

AWS CLI

例 1: デプロイを作成するには

次のcreate-deployment例では、AWS IoT Greengrass コマンドラインインターフェイスをコアデバイスにデプロイします。

```
aws greengrassv2 create-deployment \
  --cli-input-json file://cli-deployment.json
```

cli-deployment.json の内容:

```
{
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/MyGreengrassCore",
  "deploymentName": "Deployment for MyGreengrassCore",
  "components": {
    "aws.greengrass.Cli": {
      "componentVersion": "2.0.3"
    }
  },
  "deploymentPolicies": {
    "failureHandlingPolicy": "DO_NOTHING",
    "componentUpdatePolicy": {
      "timeoutInSeconds": 60,
      "action": "NOTIFY_COMPONENTS"
    },
    "configurationValidationPolicy": {
      "timeoutInSeconds": 60
    }
  },
  "iotJobConfiguration": {}
}
```

出力:

```
{
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

詳細については、[「IoT Greengrass V2 デベロッパーガイド」の「デプロイの作成」](#)を参照してください。AWS IoT V2

例 2: コンポーネント設定を更新するデプロイを作成するには

次のcreate-deployment例では、AWS IoT Greengrass nucleus コンポーネントをコアデバイスのグループにデプロイします。このデプロイでは、nucleus コンポーネントに次の設定更新が適用されます。

ターゲットデバイスのプロキシ設定をデフォルトのプロキシなし設定にリセットします。ターゲットデバイスのMQTT設定をデフォルトにリセットします。nucleus JVMのオプションを設定しますJVM。nucleus のログ記録レベルを設定します。

```
aws greengrassv2 create-deployment \
```

```
--cli-input-json file://nucleus-deployment.json
```

nucleus-deployment.json の内容:

```
{
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
  "deploymentName": "Deployment for MyGreengrassCoreGroup",
  "components": {
    "aws.greengrass.Nucleus": {
      "componentVersion": "2.0.3",
      "configurationUpdate": {
        "reset": [
          "/networkProxy",
          "/mqtt"
        ],
        "merge": "{\"jvmOptions\": \"-Xmx64m\", \"logging\": {\"level\": \"WARN
\\\"}}\"
      }
    }
  },
  "deploymentPolicies": {
    "failureHandlingPolicy": "ROLLBACK",
    "componentUpdatePolicy": {
      "timeoutInSeconds": 60,
      "action": "NOTIFY_COMPONENTS"
    },
    "configurationValidationPolicy": {
      "timeoutInSeconds": 60
    }
  },
  "iotJobConfiguration": {}
}
```

出力:

```
{
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
}
```

詳細については、[「IoT Greengrass V2 デベロッパーガイド」の「デプロイの作成」と「コンポーネント設定の更新」](#)を参照してください。AWS IoT V2

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDeployment](#)」の「」を参照してください。

delete-component

次の例は、delete-component を使用する方法を説明しています。

AWS CLI

コンポーネントバージョンを削除するには

次のdelete-component例では、Hello World コンポーネントを削除します。

```
aws greengrassv2 delete-component \  
  --arn arn:aws:greengrass:us-west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の[「コンポーネントの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteComponent](#)」の「」を参照してください。

delete-core-device

次のコード例は、delete-core-device を使用する方法を示しています。

AWS CLI

コアデバイスを削除するには

次のdelete-core-device例では、AWS IoT Greengrass コアデバイスを削除します。

```
aws greengrassv2 delete-core-device \  
  --core-device-thing-name MyGreengrassCore
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT Greengrass V2 デベロッパーガイド](#)の「[IoT Greengrass Core ソフトウェアのアンインストール](#)」を参照してください。AWS IoT V2

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCoreDevice](#)」の「」を参照してください。

describe-component

次のコード例は、describe-component を使用する方法を示しています。

AWS CLI

コンポーネントバージョンを記述するには

次のdescribe-component例では、Hello World コンポーネントについて説明します。

```
aws greengrassv2 describe-component \  
  --arn arn:aws:greengrass:us-  
west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0
```

出力:

```
{  
  "arn": "arn:aws:greengrass:us-  
west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0",  
  "componentName": "com.example>HelloWorld",  
  "componentVersion": "1.0.0",  
  "creationTimestamp": "2021-01-07T17:12:11.133000-08:00",  
  "publisher": "Amazon",  
  "description": "My first AWS IoT Greengrass component.",  
  "status": {  
    "componentState": "DEPLOYABLE",  
    "message": "NONE",  
    "errors": {}  
  },  
  "platforms": [  
    {  
      "attributes": {  
        "os": "linux"  
      }  
    }  
  ]  
}
```

```
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[コンポーネントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeComponent](#)」の「」を参照してください。

disassociate-service-role-from-account

次の例は、disassociate-service-role-from-account を使用する方法を説明しています。

AWS CLI

Greengrass サービスロールを AWS アカウントから関連付け解除するには

次のdisassociate-service-role-from-account例では、AWS アカウントの Greengrass サービスロールと AWS IoT Greengrass の関連付けを解除します。

```
aws greengrassv2 disassociate-service-role-from-account
```

出力:

```
{
  "disassociatedAt": "2022-01-19T19:26:09Z"
}
```

詳細については、AWS IoT V2 [Greengrass V2 デベロッパーガイド](#)の「[Greengrass サービスロール](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateServiceRoleFromAccount](#)」の「」を参照してください。

get-component-version-artifact

次の例は、get-component-version-artifact を使用する方法を説明しています。

AWS CLI

を取得URLしてコンポーネントアーティファクトをダウンロードするには

次のget-component-version-artifact例ではURL、ローカルデバッグコンソールコンポーネントのJAR ファイルをダウンロードする を取得します。

```
aws greengrassv2 get-component-version-artifact \
  --arn arn:aws:greengrass:us-west-2:aws:components:aws.greengrass.LocalDebugConsole:versions:2.0.3 \
  --artifact-name "Uvt6ZEzQ9TKiAuLbfXBX_APdY0TWks3uc46tHFHTzBM=/aws.greengrass.LocalDebugConsole.jar"
```

出力:

```
{
  "preSignedUrl": "https://evergreencomponentmanageme-
artifactbucket7410c9ef-g18n1iya8kwr.s3.us-west-2.amazonaws.com/public/
aws.greengrass.LocalDebugConsole/2.0.3/s3/ggv2-component-releases-prod-pdx/
EvergreenHttpDebugView/2ffc496ba41b39568968b22c582b4714a937193ee7687a45527238e696672521/
aws.greengrass.LocalDebugConsole/aws.greengrass.LocalDebugConsole.jar?X-Amz-
Security-Token=KwflKSdEXAMPLE..."
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[コンポーネントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetComponentVersionArtifact](#)」の「」を参照してください。

get-component

次のコード例は、get-component を使用する方法を示しています。

AWS CLI

例 1: コンポーネントのレシピをYAML形式 (Linux、macOSまたは Unix) でダウンロードするには

次のget-component例では、Hello World コンポーネントのレシピを YAML 形式のファイルにダウンロードします。このコマンドは次のことを行います。

--output および --queryパラメータを使用して、コマンドの出力を制御します。これらのパラメータは、コマンドの出力からレシピ BLOB を抽出します。出力の制御の詳細については、「[コマンドAWS ラインインターフェイスユーザーガイド](#)」の「[コマンド出力の制御](#)」を参照してください。base64ユーティリティを使用します。このユーティリティは、抽出された BLOB

を元のテキストにデコードします。成功した `get-component` コマンドによって返される blob は base64 でエンコードされたテキストです。元のテキストを取得するには、この BLOB をデコードする必要があります。デコードされたテキストをファイルに保存してください。コマンド (`> com.example.HelloWorld-1.0.0.json`) の最後のセクションでは、デコードされたテキストがファイルに保存されます。

```
aws greengrassv2 get-component \  
  --arn arn:aws:greengrass:us-west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0 \  
  --recipe-output-format YAML \  
  --query recipe \  
  --output text | base64 --decode > com.example.HelloWorld-1.0.0.json
```

詳細については、「AWS IoT Greengrass V2 デベロッパーガイド」の「[コンポーネントの管理](#)」を参照してください。

例 2: コンポーネントのレシピを YAML 形式でダウンロードするには (Windows CMD)

次の `get-component` 例では、Hello World コンポーネントのレシピを YAML 形式のファイルにダウンロードします。このコマンドは `certutil` ユーティリティを使用します。

```
aws greengrassv2 get-component ^  
  --arn arn:aws:greengrass:us-west-2:675946970638:components:com.example.HelloWorld:versions:1.0.0 ^  
  --recipe-output-format YAML ^  
  --query recipe ^  
  --output text > com.example.HelloWorld-1.0.0.yaml.b64  
  
certutil -  
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

詳細については、「AWS IoT Greengrass V2 デベロッパーガイド」の「[コンポーネントの管理](#)」を参照してください。

例 3: コンポーネントのレシピを YAML 形式でダウンロードするには (Windows PowerShell)

次の `get-component` 例では、Hello World コンポーネントのレシピを YAML 形式のファイルにダウンロードします。このコマンドは `certutil` ユーティリティを使用します。

```
aws greengrassv2 get-component `  
  --arn arn:aws:greengrass:us-west-2:675946970638:components:com.example.HelloWorld:versions:1.0.0 `
```

```
--recipe-output-format YAML \  
--query recipe \  
--output text > com.example.HelloWorld-1.0.0.yaml.b64
```

```
certutil -  
decode com.example.HelloWorld-1.0.0.yaml.b64 com.example.HelloWorld-1.0.0.yaml
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[コンポーネントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetComponent](#)」の「」を参照してください。

get-connectivity-info

次の例は、get-connectivity-info を使用する方法を説明しています。

AWS CLI

Greengrass コアデバイスの接続情報を取得するには

次のget-connectivity-info例では、Greengrass コアデバイスの接続情報を取得します。クライアントデバイスは、この情報を使用して、このコアデバイスで実行されるMQTTブローカーに接続します。

```
aws greengrassv2 get-connectivity-info \  
--thing-name MyGreengrassCore
```

出力:

```
{  
  "connectivityInfo": [  
    {  
      "id": "localIP_192.0.2.0",  
      "hostAddress": "192.0.2.0",  
      "portNumber": 8883  
    }  
  ]  
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[コアデバイスエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConnectivityInfo](#)」の「」を参照してください。

get-core-device

次のコード例は、get-core-device を使用する方法を示しています。

AWS CLI

コアデバイスを取得するには

次のget-core-device例では、AWS IoT Greengrass コアデバイスに関する情報を取得します。

```
aws greengrassv2 get-core-device \  
  --core-device-thing-name MyGreengrassCore
```

出力:

```
{  
  "coreDeviceThingName": "MyGreengrassCore",  
  "coreVersion": "2.0.3",  
  "platform": "linux",  
  "architecture": "amd64",  
  "status": "HEALTHY",  
  "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00",  
  "tags": {}  
}
```

詳細については、「AWS IoT Greengrass V2 デベロッパーガイド」の「[コアデバイスのステータスを確認する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCoreDevice](#)」の「」を参照してください。

get-deployment

次のコード例は、get-deployment を使用する方法を示しています。

AWS CLI

デプロイを取得するには

次のget-deployment例では、AWS IoT Greengrass nucleus コンポーネントのコアデバイスのグループにデプロイする情報を取得します。

```
aws greengrassv2 get-deployment \  
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/  
MyGreengrassCoreGroup",  
  "revisionId": "14",  
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "deploymentName": "Deployment for MyGreengrassCoreGroup",  
  "deploymentStatus": "ACTIVE",  
  "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE22222",  
  "components": {  
    "aws.greengrass.Nucleus": {  
      "componentVersion": "2.0.3",  
      "configurationUpdate": {  
        "merge": "{\"jvmOptions\": \"-Xmx64m\", \"logging\": {\"level\": \"WARN  
\"}}\",  
        "reset": [  
          "/networkProxy",  
          "/mqtt"  
        ]  
      }  
    }  
  },  
  "deploymentPolicies": {  
    "failureHandlingPolicy": "ROLLBACK",  
    "componentUpdatePolicy": {  
      "timeoutInSeconds": 60,  
      "action": "NOTIFY_COMPONENTS"  
    },  
    "configurationValidationPolicy": {  
      "timeoutInSeconds": 60  
    }  
  },  
  "iotJobConfiguration": {},  
  "creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
```

```
"isLatestForTarget": false,  
"tags": {}  
}
```

詳細については、IoT Greengrass V2 デベロッパーガイドの [「デバイスにコンポーネントをデプロイする」](#) を参照してください。AWS IoT V2

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDeployment](#)」の「」を参照してください。

get-service-role-for-account

次の例は、get-service-role-for-account を使用する方法を説明しています。

AWS CLI

AWS アカウントの Greengrass サービスロールを取得するには

次のget-service-role-for-account例では、AWS アカウントの AWS IoT Greengrass に関連付けられているサービスロールを取得します。

```
aws greengrassv2 get-service-role-for-account
```

出力:

```
{  
  "associatedAt": "2022-01-19T19:21:53Z",  
  "roleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"  
}
```

詳細については、AWS IoT V2 [Greengrass V2 デベロッパーガイドの「Greengrass サービスロール」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetServiceRoleForAccount](#)」の「」を参照してください。

list-client-devices-associated-with-core-device

次の例は、list-client-devices-associated-with-core-device を使用する方法を説明しています。

AWS CLI

コアデバイスに関連付けられたクライアントデバイスを一覧表示するには

次のlist-client-devices-associated-with-core-device例では、コアデバイスに関連付けられているすべてのクライアントデバイスを一覧表示します。

```
aws greengrassv2 list-client-devices-associated-with-core-device \
  --core-device-thing-name MyTestGreengrassCore
```

出力:

```
{
  "associatedClientDevices": [
    {
      "thingName": "MyClientDevice2",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    },
    {
      "thingName": "MyClientDevice1",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    }
  ]
}
```

詳細については、[IoT Greengrass V2 デベロッパーガイドの「ローカル IoT デバイス进行操作するAWS IoT V2」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListClientDevicesAssociatedWithCoreDevice](#)」の「」を参照してください。

list-component-versions

次のコード例は、list-component-versions を使用する方法を示しています。

AWS CLI

コンポーネントのバージョンを一覧表示するには

次のlist-component-versions例では、Hello World コンポーネントのすべてのバージョンを一覧表示します。

```
aws greengrassv2 list-component-versions \  
  --arn arn:aws:greengrass:us-  
west-2:123456789012:components:com.example.HelloWorld
```

出力:

```
{  
  "componentVersions": [  
    {  
      "componentName": "com.example.HelloWorld",  
      "componentVersion": "1.0.1",  
      "arn": "arn:aws:greengrass:us-  
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.1"  
    },  
    {  
      "componentName": "com.example.HelloWorld",  
      "componentVersion": "1.0.0",  
      "arn": "arn:aws:greengrass:us-  
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0"  
    }  
  ]  
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[コンポーネントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListComponentVersions](#)」の「」を参照してください。

list-components

次の例は、list-components を使用する方法を説明しています。

AWS CLI

コンポーネントを一覧表示するには

次のlist-components例では、現在のリージョンの AWS アカウントで定義されている各コンポーネントとその最新バージョンを一覧表示します。

```
aws greengrassv2 list-components
```

出力:

```
{
  "components": [
    {
      "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld",
      "componentName": "com.example.HelloWorld",
      "latestVersion": {
        "arn": "arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.1",
        "componentVersion": "1.0.1",
        "creationTimestamp": "2021-01-08T16:51:07.352000-08:00",
        "description": "My first AWS IoT Greengrass component.",
        "publisher": "Amazon",
        "platforms": [
          {
            "attributes": {
              "os": "linux"
            }
          }
        ]
      }
    }
  ]
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[コンポーネントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListComponents](#)」の「」を参照してください。

list-core-devices

次の例は、list-core-devices を使用する方法を説明しています。

AWS CLI

コアデバイスを一覧表示するには

次のlist-core-devices例では、現在のリージョンの AWS アカウントにある AWS IoT Greengrass コアデバイスを一覧表示します。


```
aws greengrassv2 list-core-devices
```

出力:

```
{
  "coreDevices": [
    {
      "coreDeviceThingName": "MyGreengrassCore",
      "status": "HEALTHY",
      "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00"
    }
  ]
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイドの「[コアデバイスのステータスを確認する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCoreDevices](#)」の「」を参照してください。

list-deployments

次のコード例は、list-deployments を使用する方法を示しています。

AWS CLI

デプロイを一覧表示するには

次のlist-deployments例では、現在のリージョンの AWS アカウントで定義された各デプロイの最新リビジョンを一覧表示します。

```
aws greengrassv2 list-deployments
```

出力:

```
{
  "deployments": [
    {
      "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/MyGreengrassCoreGroup",
      "revisionId": "14",
    }
  ]
}
```

```
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "deploymentName": "Deployment for MyGreengrassCoreGroup",
    "creationTimestamp": "2021-01-07T17:21:20.691000-08:00",
    "deploymentStatus": "ACTIVE",
    "isLatestForTarget": false
  },
  {
    "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/
MyGreengrassCore",
    "revisionId": "1",
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "deploymentName": "Deployment for MyGreengrassCore",
    "creationTimestamp": "2021-01-06T16:10:42.407000-08:00",
    "deploymentStatus": "COMPLETED",
    "isLatestForTarget": false
  }
]
}
```

詳細については、IoT Greengrass V2 デベロッパーガイドの [「デバイスにコンポーネントをデプロイする」](#) を参照してください。AWS IoT V2

- API 詳細については、AWS CLI 「コマンドリファレンス [ListDeployments](#)」の「」を参照してください。

list-effective-deployments

次のコード例は、list-effective-deployments を使用する方法を示しています。

AWS CLI

デプロイジョブを一覧表示するには

次のlist-effective-deployments例では、AWS IoT Greengrass コアデバイスに適用されるデプロイを一覧表示します。

```
aws greengrassv2 list-effective-deployments \
  --core-device-thing-name MyGreengrassCore
```

出力:

```
{
  "effectiveDeployments": [
```

```
{
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "deploymentName": "Deployment for MyGreengrassCore",
  "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thing/
MyGreengrassCore",
  "coreDeviceExecutionStatus": "COMPLETED",
  "reason": "SUCCESSFUL",
  "creationTimestamp": "2021-01-06T16:10:42.442000-08:00",
  "modifiedTimestamp": "2021-01-08T17:21:27.830000-08:00"
},
{
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "deploymentName": "Deployment for MyGreengrassCoreGroup",
  "iotJobId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE44444",
  "iotJobArn": "arn:aws:iot:us-west-2:123456789012:job/a1b2c3d4-5678-90ab-
cdef-EXAMPLE44444",
  "targetArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
MyGreengrassCoreGroup",
  "coreDeviceExecutionStatus": "SUCCEEDED",
  "reason": "SUCCESSFUL",
  "creationTimestamp": "2021-01-07T17:19:20.394000-08:00",
  "modifiedTimestamp": "2021-01-07T17:21:20.721000-08:00"
}
]
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイドの「[コアデバイスのステータスを確認する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEffectiveDeployments](#)」の「」を参照してください。

list-installed-components

次の例は、list-installed-components を使用する方法を説明しています。

AWS CLI

コアデバイスにインストールされているコンポーネントを一覧表示するには

次のlist-installed-components例では、AWS IoT Greengrass コアデバイスにインストールされているコンポーネントを一覧表示します。

```
aws greengrassv2 list-installed-components \  
  --core-device-thing-name MyGreengrassCore
```

出力:

```
{  
  "installedComponents": [  
    {  
      "componentName": "aws.greengrass.Cli",  
      "componentVersion": "2.0.3",  
      "lifecycleState": "RUNNING",  
      "isRoot": true  
    },  
    {  
      "componentName": "aws.greengrass.Nucleus",  
      "componentVersion": "2.0.3",  
      "lifecycleState": "FINISHED",  
      "isRoot": true  
    }  
  ]  
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイドの「[コアデバイスのステータスを確認する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInstalledComponents](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、AWS IoT Greengrass コアデバイスのすべてのタグを一覧表示します。

```
aws greengrassv2 list-tags-for-resource \  
  --resource-id
```

```
--resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore
```

出力:

```
{
  "tags": {
    "Owner": "richard-roe"
  }
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[リソースにタグを付ける](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを追加

次のtag-resource例では、AWS IoT Greengrass コアデバイスに所有者タグを追加します。このタグを使用して、コアデバイスを所有するユーザーに基づいてコアデバイスへのアクセスを制御できます。

```
aws greengrassv2 tag-resource \  
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tags Owner=richard-roe
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[リソースにタグを付ける](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、`untag-resource` を使用する方法を説明しています。

AWS CLI

リソースからタグを削除するには

次の`untag-resource`例では、AWS IoT Greengrass コアデバイスから所有者タグを削除します。

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tag-keys Owner
```

このコマンドでは何も出力されません。

詳細については、「AWS IoT Greengrass V2 デベロッパーガイド」の「[リソースにタグを付ける](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-connectivity-info

次のコード例は、`update-connectivity-info` を使用する方法を示しています。

AWS CLI

Greengrass コアデバイスの接続情報を更新するには

次の`update-connectivity-info`例では、Greengrass コアデバイスの接続情報を取得します。クライアントデバイスは、この情報を使用して、このコアデバイスで実行されるMQTTブローカーに接続します。

```
aws greengrassv2 update-connectivity-info \  
  --thing-name MyGreengrassCore \  
  --cli-input-json file://core-device-connectivity-info.json
```

`core-device-connectivity-info.json` の内容:

```
{
  "connectivityInfo": [
    {
      "hostAddress": "192.0.2.0",
      "portNumber": 8883,
      "id": "localIP_192.0.2.0"
    }
  ]
}
```

出力:

```
{
  "version": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

詳細については、AWS IoT Greengrass V2 デベロッパーガイド」の「[コアデバイスエンドポイントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateConnectivityInfo](#)」の「」を参照してください。

AWS IoT Jobs SDK release を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Jobs SDK release。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-job-execution

次の例は、describe-job-execution を使用する方法を説明しています。

AWS CLI

ジョブ実行の詳細を取得するには

次のdescribe-job-execution例では、指定されたジョブとモノの最新の実行の詳細を取得します。

```
aws iot-jobs-data describe-job-execution \  
  --job-id SampleJob \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

出力:

```
{  
  "execution": {  
    "approximateSecondsBeforeTimedOut": 88,  
    "executionNumber": 2939653338,  
    "jobId": "SampleJob",  
    "lastUpdatedAt": 1567701875.743,  
    "queuedAt": 1567701902.444,  
    "status": "QUEUED",  
    "thingName": "MotionSensor1 ",  
    "versionNumber": 3  
  }  
}
```

詳細については、AWS IoT デベロッパーガイドの「[デバイスとジョブ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeJobExecution](#)」の「」を参照してください。

get-pending-job-executions

次の例は、get-pending-job-executions を使用する方法を説明しています。

AWS CLI

モノのターミナルステータスではないすべてのジョブのリストを取得するには

次のget-pending-job-executions例では、指定されたモノのターミナル状態にないすべてのジョブのリストを表示します。

```
aws iot-jobs-data get-pending-job-executions \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

出力:

```
{  
  "InProgressJobs": [  
  ],  
  "queuedJobs": [  
    {  
      "executionNumber": 2939653338,  
      "jobId": "SampleJob",  
      "lastUpdatedAt": 1567701875.743,  
      "queuedAt": 1567701902.444,  
      "versionNumber": 3  
    }  
  ]  
}
```

詳細については、AWS IoT デベロッパーガイドの「[デバイスとジョブ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPendingJobExecutions](#)」の「」を参照してください。

start-next-pending-job-execution

次のコード例は、start-next-pending-job-execution を使用する方法を示しています。

AWS CLI

モノの次の保留中のジョブ実行を取得して開始するには

次のstart-next-pending-job-execution例では、ステータスが IN_PROGRESS または指定されたモノQUEUEDの次のジョブ実行を取得して開始します。

```
aws iot-jobs-data start-next-pending-job-execution \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

出力:

```
{  
  "execution": {  
    "approximateSecondsBeforeTimedOut": 88,  
    "executionNumber": 2939653338,  
    "jobId": "SampleJob",  
    "lastUpdatedAt": 1567714853.743,  
    "queuedAt": 1567701902.444,  
    "startedAt": 1567714871.690,  
    "status": "IN_PROGRESS",  
    "thingName": "MotionSensor1 ",  
    "versionNumber": 3  
  }  
}
```

詳細については、AWS IoT デベロッパーガイドの「[デバイスとジョブ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartNextPendingJobExecution](#)」の「」を参照してください。

update-job-execution

次の例は、update-job-execution を使用する方法を説明しています。

AWS CLI

ジョブ実行のステータスを更新するには

次のupdate-job-execution例では、指定されたジョブとモノのステータスを更新します。

```
aws iot-jobs-data update-job-execution \  
  --job-id SampleJob \  
  --thing-name MotionSensor1 \  
  --status REMOVED \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

出力:

```
{
  "executionState": {
    "status": "REMOVED",
    "versionNumber": 3
  },
}
```

詳細については、AWS 「IoT デベロッパーガイド」の [「デバイスとジョブ」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateJobExecution](#)」の「」を参照してください。

AWS IoT SiteWise を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT SiteWise。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-assets

次の例は、associate-assets を使用する方法を説明しています。

AWS CLI

子アセットを親アセットに関連付けるには

次のassociate-assets例では、風力タービンアセットを風力発電所アセットに関連付けます。風力タービンアセットモデルは、風力発電所アセットモデルの階層として存在します。

```
aws iotsitewise associate-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \  
  --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイド」の「[アセットの関連付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateAssets](#)」の「」を参照してください。

batch-associate-project-assets

次の例は、batch-associate-project-assets を使用する方法を説明しています。

AWS CLI

アセットをプロジェクトに関連付けるには

次のbatch-associate-project-assets例では、風力発電所アセットをプロジェクトに関連付けます。

```
aws iotsitewise batch-associate-project-assets \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
  --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「IoT Monitor アプリケーションガイド」の「[プロジェクトへのアセットの追加](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchAssociateProjectAssets](#)」の「」を参照してください。

batch-disassociate-project-assets

次の例は、batch-disassociate-project-assets を使用する方法を説明しています。

AWS CLI

プロジェクトからアセットの関連付けを解除するには

次のbatch-disassociate-project-assets例では、風力発電所アセットとプロジェクトとの関連付けを解除します。

```
aws iotsitewise batch-disassociate-project-assets \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「IoT Monitor アプリケーションガイド」の「[プロジェクトへのアセットの追加](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDisassociateProjectAssets](#)」の「」を参照してください。

batch-put-asset-property-value

次のコード例は、batch-put-asset-property-value を使用する方法を示しています。

AWS CLI

アセットプロパティにデータを送信するには

次のbatch-put-asset-property-value例では、プロパティエイリアスによって識別されるアセットプロパティに電力と温度データを送信します。

```
aws iotsitewise batch-put-asset-property-value \  
  --cli-input-json file://batch-put-asset-property-value.json
```

batch-put-asset-property-value.json の内容:

```
{  
  "entries": [  
    {  
      "entryId": "1575691200-company-windfarm-3-turbine-7-power",  
      "propertyAlias": "company-windfarm-3-turbine-7-power",  
      "propertyValues": [  
        {
```

```

        "value": {
            "doubleValue": 4.92
        },
        "timestamp": {
            "timeInSeconds": 1575691200
        },
        "quality": "GOOD"
    }
]
},
{
    "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",
    "propertyAlias": "company-windfarm-3-turbine-7-temperature",
    "propertyValues": [
        {
            "value": {
                "integerValue": 38
            },
            "timestamp": {
                "timeInSeconds": 1575691200
            }
        }
    ]
}
]
}

```

出力:

```

{
  "errorEntries": []
}

```

詳細については、[AWS IoT SiteWise API ユーザーガイドの「IoT を使用したデータの取り込み」](#)を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchPutAssetPropertyValue](#)」の「」を参照してください。

create-access-policy

次のコード例は、create-access-policy を使用する方法を示しています。

AWS CLI

例 1: ポータルへのユーザー管理アクセスを許可するには

次のcreate-access-policy例では、風力発電所会社のウェブポータルへのユーザー管理アクセスを許可するアクセスポリシーを作成します。

```
aws iotsitewise create-access-policy \  
  --cli-input-json file://create-portal-administrator-access-policy.json
```

create-portal-administrator-access-policy.json の内容:

```
{  
  "accessPolicyIdentity": {  
    "user": {  
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE"  
    }  
  },  
  "accessPolicyPermission": "ADMINISTRATOR",  
  "accessPolicyResource": {  
    "portal": {  
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"  
    }  
  }  
}
```

出力:

```
{  
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",  
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/  
a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE"  
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[ポータル管理者の追加または削除](#)」を参照してください。

例 2: ユーザーにプロジェクトへの読み取り専用アクセスを許可するには

次のcreate-access-policy例では、風力発電所プロジェクトへの読み取り専用アクセスをユーザーに許可するアクセスポリシーを作成します。

```
aws iotsitewise create-access-policy \  
  --cli-input-json file://create-project-viewer-access-policy.json
```

create-project-viewer-access-policy.json の内容:

```
{  
  "accessPolicyIdentity": {  
    "user": {  
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"  
    }  
  },  
  "accessPolicyPermission": "VIEWER",  
  "accessPolicyResource": {  
    "project": {  
      "id": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE"  
    }  
  }  
}
```

出力:

```
{  
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE",  
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/  
a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE"  
}
```

詳細については、AWS IoT SiteWise Monitor アプリケーションガイド」の「[プロジェクトビューワーの割り当て](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAccessPolicy](#)」の「」を参照してください。

create-asset-model

次の例は、create-asset-model を使用する方法を説明しています。

AWS CLI

アセットモデルを作成するには

次のcreate-asset-model例では、次のプロパティを持つ風力タービンを定義するアセットモデルを作成します。

シリアル番号 - 風力のシリアル番号 turbineGenerated - 風力からの生成された電力データストリーム turbineTemperature C - CelsiusTemperature F の風力タービンからの温度データストリーム - マッピングされた温度データは摂氏から華氏にポイントします

```
aws iotsitewise create-asset-model \  
  --cli-input-json file://create-wind-turbine-model.json
```

create-wind-turbine-model.json の内容:

```
{  
  "assetModelName": "Wind Turbine Model",  
  "assetModelDescription": "Represents a wind turbine",  
  "assetModelProperties": [  
    {  
      "name": "Serial Number",  
      "dataType": "STRING",  
      "type": {  
        "attribute": {}  
      }  
    },  
    {  
      "name": "Generated Power",  
      "dataType": "DOUBLE",  
      "unit": "kW",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature C",  
      "dataType": "DOUBLE",  
      "unit": "Celsius",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature F",  
      "dataType": "DOUBLE",  
      "unit": "Fahrenheit",
```

```
    "type": {
      "transform": {
        "expression": "temp_c * 9 / 5 + 32",
        "variables": [
          {
            "name": "temp_c",
            "value": {
              "propertyId": "Temperature C"
            }
          }
        ]
      }
    },
    {
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "metric": {
          "expression": "sum(power)",
          "variables": [
            {
              "name": "power",
              "value": {
                "propertyId": "Generated Power"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "1h"
            }
          }
        }
      }
    }
  ]
}
```

出力:

```
{
```

```
"assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
"assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
"assetModelStatus": {
  "state": "CREATING"
}
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[アセットモデルの定義](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAssetModel](#)」の「」を参照してください。

create-asset

次の例は、create-asset を使用する方法を説明しています。

AWS CLI

アセットを作成するには

次のcreate-asset例では、風力タービンアセットモデルから風力タービンアセットを作成します。

```
aws iotsitewise create-asset \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --asset-name "Wind Turbine 1"
```

出力:

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "assetStatus": {
    "state": "CREATING"
  }
}
```

詳細については、「IoT ユーザーガイド」の「[アセットの作成](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAsset](#)」の「」を参照してください。

create-dashboard

次の例は、create-dashboard を使用する方法を説明しています。

AWS CLI

ダッシュボードを作成するには

次のcreate-dashboard例では、風力発電所の合計発電量を表示する折れ線グラフを含むダッシュボードを作成します。

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --dashboard-name "Wind Farm" \  
  --dashboard-definition file://create-wind-farm-dashboard.json
```

create-wind-farm-dashboard.json の内容:

```
{  
  "widgets": [  
    {  
      "type": "monitor-line-chart",  
      "title": "Generated Power",  
      "x": 0,  
      "y": 0,  
      "height": 3,  
      "width": 3,  
      "metrics": [  
        {  
          "label": "Power",  
          "type": "iotsitewise",  
          "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",  
          "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"  
        }  
      ]  
    }  
  ]  
}
```

出力:

```
{
  "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
  "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE"
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[ダッシュボードの作成 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDashboard](#)」の「」を参照してください。

create-gateway

次のコード例は、create-gateway を使用する方法を示しています。

AWS CLI

ゲートウェイを作成するには

次のcreate-gateway例では、AWS IoT Greengrass で実行されるゲートウェイを作成します。

```
aws iotsitewise create-gateway \
  --gateway-name ExampleCorpGateway \
  --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}
```

出力:

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[ゲートウェイの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateGateway](#)」の「」を参照してください。

create-portal

次のコード例は、create-portal を使用する方法を示しています。

AWS CLI

ポータルを作成するには

次のcreate-portal例では、風力発電所会社のウェブポータルを作成します。ポータルは、AWS シングルサインオンを有効にした同じリージョンでのみ作成できます。

```
aws iotsitewise create-portal \  
  --portal-name WindFarmPortal \  
  --portal-description "A portal that contains wind farm projects for Example Corp." \  
  --portal-contact-email support@example.com \  
  --role-arn arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole
```

出力:

```
{  
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",  
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",  
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",  
  "portalStatus": {  
    "state": "CREATING"  
  },  
  "ssoApplicationId": "ins-a1b2c3d4-EXAMPLE"  
}
```

詳細については、[AWS IoT SiteWise ユーザーガイド](#)の「IoT Monitor の開始方法」とAWS IoT SiteWise ユーザーガイドの「[有効化 AWS SSO](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePortal](#)」の「」を参照してください。

create-project

次の例は、create-project を使用する方法を説明しています。

AWS CLI

プロジェクトを作成するには

次のcreate-project例では、風力発電所プロジェクトを作成します。

```
aws iotsitewise create-project \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \  
  --project-name "Wind Farm 1" \  
  --project-description "Contains asset visualizations for Wind Farm #1 for Example Corp."
```

出力:

```
{  
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",  
  "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/  
a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"  
}
```

詳細については、「IoT Monitor アプリケーションガイド」の[「プロジェクトの作成」](#)を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProject](#)」の「」を参照してください。

delete-access-policy

次のコード例は、delete-access-policy を使用する方法を示しています。

AWS CLI

プロジェクトまたはポータルへのユーザーのアクセスを取り消すには

次のdelete-access-policy例では、ポータルへのユーザー管理アクセスを許可するアクセスポリシーを削除します。

```
aws iotsitewise delete-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイドの「[ポータル管理者の追加または削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccessPolicy](#)」の「」を参照してください。

delete-asset-model

次の例は、delete-asset-model を使用する方法を説明しています。

AWS CLI

アセットモデルを削除するには

次のdelete-asset-model例では、風力タービンアセットモデルを削除します。

```
aws iotsitewise delete-asset-model \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

出力:

```
{  
  "assetModelStatus": {  
    "state": "DELETING"  
  }  
}
```

詳細については、「IoT ユーザーガイド」の「[アセットモデルの削除](#)」を参照してください。

AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAssetModel](#)」の「」を参照してください。

delete-asset

次の例は、delete-asset を使用する方法を説明しています。

AWS CLI

アセットを削除するには

次のdelete-asset例では、風力タービンアセットを削除します。


```
aws iotsitewise delete-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

出力:

```
{  
  "assetStatus": {  
    "state": "DELETING"  
  }  
}
```

詳細については、「IoT ユーザーガイド」の [「アセットの削除」](#) を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAsset](#)」の「」を参照してください。

delete-dashboard

次のコード例は、delete-dashboard を使用する方法を示しています。

AWS CLI

ダッシュボードを削除するには

次のdelete-dashboard例では、風力タービンダッシュボードを削除します。

```
aws iotsitewise delete-dashboard \  
  --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise Monitor アプリケーションガイドの [「ダッシュボードの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDashboard](#)」の「」を参照してください。

delete-gateway

次のコード例は、delete-gateway を使用する方法を示しています。

AWS CLI

ゲートウェイを削除するには

次のdelete-gateway例では、ゲートウェイを削除します。

```
aws iotsitewise delete-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイドの「[ゲートウェイを使用したデータの取り込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGateway](#)」の「」を参照してください。

delete-portal

次のコード例は、delete-portal を使用する方法を示しています。

AWS CLI

ポータルを削除するには

次のdelete-portal例では、風力発電所会社のウェブポータルを削除します。

```
aws iotsitewise delete-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

出力:

```
{  
  "portalStatus": {  
    "state": "DELETING"  
  }  
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[ポータルの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePortal](#)」の「」を参照してください。

delete-project

次の例は、delete-project を使用する方法を説明しています。

AWS CLI

プロジェクトを削除するには

次のdelete-project例では、風力発電所プロジェクトを削除します。

```
aws iotsitewise delete-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「IoT Monitor アプリケーションガイド」の「[プロジェクトの削除](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProject](#)」の「」を参照してください。

describe-access-policy

次のコード例は、describe-access-policy を使用する方法を示しています。

AWS CLI

アクセスポリシーを記述するには

次のdescribe-access-policy例では、風力発電所会社のウェブポータルへのユーザー管理アクセスを許可するアクセスポリシーについて説明します。

```
aws iotsitewise describe-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE
```

出力:

```
{  
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/  
a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
  "accessPolicyIdentity": {  
    "user": {
```

```
        "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE"
      }
    },
    "accessPolicyResource": {
      "portal": {
        "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
      }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyCreationDate": "2020-02-20T22:35:15.552880124Z",
    "accessPolicyLastUpdateDate": "2020-02-20T22:35:15.552880124Z"
  }
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[ポータル管理者の追加または削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccessPolicy](#)」の「」を参照してください。

describe-asset-model

次のコード例は、describe-asset-model を使用する方法を示しています。

AWS CLI

アセットモデルを記述するには

次のdescribe-asset-model例では、風力発電所のアセットモデルについて説明します。

```
aws iotsitewise describe-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

出力:

```
{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelName": "Wind Farm Model",
  "assetModelDescription": "Represents a wind farm that comprises many wind turbines",
  "assetModelProperties": [
    {
```

```

    "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
    "name": "Total Generated Power",
    "dataType": "DOUBLE",
    "unit": "kW",
    "type": {
      "metric": {
        "expression": "sum(power)",
        "variables": [
          {
            "name": "power",
            "value": {
              "propertyId": "a1b2c3d4-5678-90ab-
cdef-66666EXAMPLE",
              "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
      "name": "Region",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": " "
        }
      }
    }
  ],
  "assetModelHierarchies": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
  ],

```

```
"assetModelCreationDate": 1575671284.0,
"assetModelLastUpdateDate": 1575671988.0,
"assetModelStatus": {
  "state": "ACTIVE"
}
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[特定のアセットモデルの説明](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAssetModel](#)」の「」を参照してください。

describe-asset-property

次のコード例は、describe-asset-property を使用する方法を示しています。

AWS CLI

アセットプロパティを記述するには

次のdescribe-asset-property例では、風力発電所アセットの合計生成電力プロパティについて説明します。

```
aws iotsitewise describe-asset-property \
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-99999EXAMPLE
```

出力:

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
  "assetName": "Wind Farm 1",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetProperty": {
    "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
    "name": "Total Generated Power",
    "notification": {
      "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "state": "DISABLED"
    }
  }
}
```

```
    },
    "dataType": "DOUBLE",
    "unit": "kW",
    "type": {
      "metric": {
        "expression": "sum(power)",
        "variables": [
          {
            "name": "power",
            "value": {
              "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE",
              "hierarchyId": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE"
            }
          }
        ]
      },
      "window": {
        "tumbling": {
          "interval": "1h"
        }
      }
    }
  }
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[特定のアセットプロパティの説明](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAssetProperty](#)」の「」を参照してください。

describe-asset

次のコード例は、describe-asset を使用する方法を示しています。

AWS CLI

アセットを記述するには

次のdescribe-asset例では、風力発電所アセットについて説明します。

```
aws iotsitewise describe-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

出力:

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
  "assetName": "Wind Farm 1",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetProperties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
      "name": "Region",
      "dataType": "STRING"
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW"
    }
  ],
  "assetHierarchies": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
      "name": "Wind Turbines"
    }
  ],
  "assetCreationDate": 1575672453.0,
  "assetLastUpdateDate": 1575672453.0,
  "assetStatus": {
    "state": "ACTIVE"
  }
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[特定のアセットの説明](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAsset](#)」の「」を参照してください。

describe-dashboard

次のコード例は、describe-dashboard を使用する方法を示しています。

AWS CLI

ダッシュボードを記述するには

次のdescribe-dashboard例では、指定された風力発電所ダッシュボードについて説明します。

```
aws iotsitewise describe-dashboard \  
  --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

出力:

```
{  
  "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",  
  "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/  
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",  
  "dashboardName": "Wind Farm",  
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",  
  "dashboardDefinition": "{\"widgets\": [{\"type\": \"monitor-line-chart\", \"title  
\": \"Generated Power\", \"x\": 0, \"y\": 0, \"height\": 3, \"width\": 3, \"metrics\":  
[ {\"label\": \"Power\", \"type\": \"iotsitewise\", \"assetId\": \"a1b2c3d4-5678-90ab-  
cdef-44444EXAMPLE\", \"propertyId\": \"a1b2c3d4-5678-90ab-cdef-99999EXAMPLE\" } ] } ]\",  
  "dashboardCreationDate": "2020-05-01T20:32:12.228476348Z",  
  "dashboardLastUpdateDate": "2020-05-01T20:32:12.228476348Z"  
}
```

詳細については、AWS IoT SiteWise Monitor アプリケーションガイド」の「[ダッシュボードの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDashboard](#)」の「」を参照してください。

describe-gateway-capability-configuration

次のコード例は、describe-gateway-capability-configuration を使用する方法を示しています。

AWS CLI

ゲートウェイ機能を記述するには

次のdescribe-gateway-capability-configuration例では、OPC-UA ソース機能について説明します。

```
aws iotsitewise describe-gateway-capability-configuration \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --capability-namespace "iotsitewise:opcuacollector:1"
```

出力:

```
{  
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "capabilityNamespace": "iotsitewise:opcuacollector:1",  
  "capabilityConfiguration": "{\n\"sources\":[\n\"name\":\n\"Wind Farm #1\",  
  \n\"endpoint\":{\n\"certificateTrust\":{\n\"type\":\n\"TrustAny\",  
  \n\"endpointUri\n\":\n\"opc.tcp://203.0.113.0:49320\",  
  \n\"securityPolicy\":\n\"BASIC256\",  
  \n\"messageSecurityMode\":\n\"SIGN_AND_ENCRYPT\",  
  \n\"identityProvider\":  
{\n\"type\":\n\"Username\",  
  \n\"usernameSecretArn\":\n\"arn:aws:secretsmanager:us-east-1:123456789012:secret:greenrass-factory1-auth-3QNDmM\",  
  \n\"nodeFilterRules\":  
[]},  
  \n\"measurementDataStreamPrefix\":\n\"\"}}],  
  \n\"capabilitySyncStatus\": \"IN_SYNC\"  
  }  
}
```

詳細については、「IoT ユーザーガイド」の「[データソースの設定](#)」を参照してください。
AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGatewayCapabilityConfiguration](#)」の「」を参照してください。

describe-gateway

次のコード例は、describe-gateway を使用する方法を示しています。

AWS CLI

ゲートウェイを記述するには

次のdescribe-gateway例では、ゲートウェイについて説明します。

```
aws iotsitewise describe-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

出力:

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayName": "ExampleCorpGateway",
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayPlatform": {
    "greengrass": {
      "groupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE"
    }
  },
  "gatewayCapabilitySummaries": [
    {
      "capabilityNamespace": "iotsitewise:opcuacollector:1",
      "capabilitySyncStatus": "IN_SYNC"
    }
  ],
  "creationDate": 1588369971.457,
  "lastUpdateDate": 1588369971.457
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの [「ゲートウェイを使用したデータの取り込み」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeGateway](#)」の「」を参照してください。

describe-logging-options

次のコード例は、describe-logging-options を使用する方法を示しています。

AWS CLI

current AWS IoT SiteWise ログ記録オプションを取得するには

次のdescribe-logging-options例では、現在のリージョンの AWS アカウントの現在の AWS IoT SiteWise ログ記録オプションを取得します。

```
aws iotsitewise describe-logging-options
```

出力:

```
{
  "loggingOptions": {
    "level": "INFO"
  }
}
```

詳細については、[AWS IoT SiteWise ユーザーガイド](#)の「[Amazon CloudWatch Logs を使用した IoT のモニタリング](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoggingOptions](#)」の「」を参照してください。

describe-portal

次の例は、describe-portal を使用する方法を説明しています。

AWS CLI

ポータルを記述するには

次のdescribe-portal例では、風力発電所会社のウェブポータルについて説明します。

```
aws iotsitewise describe-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

出力:

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp.",
  "portalClientId": "E-a1b2c3d4e5f6_a1b2c3d4e5f6EXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalContactEmail": "support@example.com",
  "portalStatus": {
    "state": "ACTIVE"
  }
}
```

```
  },
  "portalCreationDate": "2020-02-04T23:01:52.90248068Z",
  "portalLastUpdateDate": "2020-02-04T23:01:52.90248078Z",
  "roleArn": "arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole"
}
```

詳細については、[「IoT ユーザーガイド」の「ポータル管理」](#)を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePortal](#)」の「」を参照してください。

describe-project

次の例は、describe-project を使用する方法を説明しています。

AWS CLI

プロジェクトを記述するには

次のdescribe-project例では、風力発電所プロジェクトについて説明します。

```
aws iotsitewise describe-project \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE
```

出力:

```
{
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",
  "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/
a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",
  "projectName": "Wind Farm 1",
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "projectDescription": "Contains asset visualizations for Wind Farm #1 for
Example Corp.",
  "projectCreationDate": "2020-02-20T21:58:43.362246001Z",
  "projectLastUpdateDate": "2020-02-20T21:58:43.362246095Z"
}
```

詳細については、AWS IoT SiteWise Monitor [アプリケーションガイド](#)の「[プロジェクトの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeProject](#)」の「」を参照してください。

disassociate-assets

次の例は、disassociate-assets を使用する方法を説明しています。

AWS CLI

親アセットから子アセットの関連付けを解除するには

次のdisassociate-assets例では、風力タービンアセットと風力発電所アセットの関連付けを解除します。

```
aws iotsitewise disassociate-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \  
  --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「AWS IoT SiteWise ユーザーガイド」の「[アセットの関連付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateAssets](#)」の「」を参照してください。

get-asset-property-aggregates

次のコード例は、get-asset-property-aggregates を使用する方法を示しています。

AWS CLI

アセットプロパティの集計平均とカウント値を取得するには

次のget-asset-property-aggregates例では、風力タービンアセットの平均総電力と 1 時間の合計電力データポイント数を取得します。

```
aws iotsitewise get-asset-property-aggregates \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \  
  --start-time 2019-01-01T00:00:00.000Z \  
  --end-time 2019-01-01T00:00:00.000Z
```

```
--start-date 1580849400 \  
--end-date 1580853000 \  
--aggregate-types AVERAGE COUNT \  
--resolution 1h
```

出力:

```
{  
  "aggregatedValues": [  
    {  
      "timestamp": 1580850000.0,  
      "quality": "GOOD",  
      "value": {  
        "average": 8723.46538886233,  
        "count": 12.0  
      }  
    }  
  ]  
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[アセットプロパティ集計のクエリ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAssetPropertyAggregates](#)」の「」を参照してください。

get-asset-property-value-history

次の例は、get-asset-property-value-history を使用する方法を説明しています。

AWS CLI

アセットプロパティの履歴値を取得するには

次のget-asset-property-value-history例では、風力タービンアセットの合計電力値を20分間取得します。

```
aws iotsitewise get-asset-property-value-history \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \  
  --start-date 1580851800 \  
  --end-date 1580853000
```

出力:

```
{
  "assetPropertyValueHistory": [
    {
      "value": {
        "doubleValue": 7217.787046814844
      },
      "timestamp": {
        "timeInSeconds": 1580852100,
        "offsetInNanos": 0
      },
      "quality": "GOOD"
    },
    {
      "value": {
        "doubleValue": 6941.242811875451
      },
      "timestamp": {
        "timeInSeconds": 1580852400,
        "offsetInNanos": 0
      },
      "quality": "GOOD"
    },
    {
      "value": {
        "doubleValue": 6976.797662266717
      },
      "timestamp": {
        "timeInSeconds": 1580852700,
        "offsetInNanos": 0
      },
      "quality": "GOOD"
    },
    {
      "value": {
        "doubleValue": 6890.8677520453875
      },
      "timestamp": {
        "timeInSeconds": 1580853000,
        "offsetInNanos": 0
      },
      "quality": "GOOD"
    }
  ]
}
```



```
]
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[過去のアセットプロパティ値のクエリ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAssetPropertyValueHistory](#)」の「」を参照してください。

get-asset-property-value

次の例は、get-asset-property-value を使用する方法を説明しています。

AWS CLI

アセットプロパティの現在の値を取得するには

次のget-asset-property-value例では、風力タービンアセットの現在の合計電力を取得します。

```
aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE
```

出力:

```
{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },
    "timestamp": {
      "timeInSeconds": 1580853000,
      "offsetInNanos": 0
    },
    "quality": "GOOD"
  }
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[現在のアセットプロパティ値のクエリ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAssetPropertyValue](#)」の「」を参照してください。

list-access-policies

次のコード例は、list-access-policies を使用する方法を示しています。

AWS CLI

すべてのアクセスポリシーを一覧表示するには

次のlist-access-policies例では、ポータル管理者であるユーザーのすべてのアクセスポリシーを一覧表示します。

```
aws iotsitewise list-access-policies \  
  --identity-type USER \  
  --identity-id a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE
```

出力:

```
{  
  "accessPolicySummaries": [  
    {  
      "id": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
      "identity": {  
        "user": {  
          "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"  
        }  
      },  
      "resource": {  
        "portal": {  
          "id": "a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE"  
        }  
      },  
      "permission": "ADMINISTRATOR"  
    }  
  ]  
}
```

詳細については、AWS IoT SiteWise ユーザーガイド」の「[ポータルの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccessPolicies](#)」の「」を参照してください。

list-asset-models

次のコード例は、list-asset-models を使用する方法を示しています。

AWS CLI

すべてのアセットモデルを一覧表示するには

次のlist-asset-models例では、現在のリージョンの AWS アカウントで定義されているすべてのアセットモデルを一覧表示します。

```
aws iotsitewise list-asset-models
```

出力:

```
{
  "assetModelSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "name": "Wind Farm Model",
      "description": "Represents a wind farm that comprises many wind turbines",
      "creationDate": 1575671284.0,
      "lastUpdateDate": 1575671988.0,
      "status": {
        "state": "ACTIVE"
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "name": "Wind Turbine Model",
      "description": "Represents a wind turbine manufactured by Example Corp",
      "creationDate": 1575671207.0,
      "lastUpdateDate": 1575686273.0,
      "status": {
```

```
        "state": "ACTIVE"
      }
    ]
  }
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[すべてのアセットモデルの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssetModels](#)」の「」を参照してください。

list-assets

次のコード例は、list-assets を使用する方法を示しています。

AWS CLI

例 1: すべての最上位アセットを一覧表示するには

次のlist-assets例では、アセット階層ツリーの最上位で、現在のリージョンの AWS アカウントで定義されているすべてのアセットを一覧表示します。

```
aws iotsitewise list-assets \
  --filter TOP_LEVEL
```

出力:

```
{
  "assetSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
      "name": "Wind Farm 1",
      "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "creationDate": 1575672453.0,
      "lastUpdateDate": 1575672453.0,
      "status": {
        "state": "ACTIVE"
      },
      "hierarchies": [
```

```
    {
      "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
      "name": "Wind Turbines"
    }
  ]
}
```

詳細については、AWS IoT SiteWise ユーザーガイド」の「[アセットの一覧表示](#)」を参照してください。

例 2: アセットモデルに基づいてすべてのアセットを一覧表示するには

次のlist-assets例では、アセットモデルに基づいて、現在のリージョンの AWS アカウントで定義されているすべてのアセットを一覧表示します。

```
aws iotsitewise list-assets \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

出力:

```
{
  "assetSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "name": "Wind Turbine 1",
      "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "creationDate": 1575671550.0,
      "lastUpdateDate": 1575686308.0,
      "status": {
        "state": "ACTIVE"
      },
      "hierarchies": []
    }
  ]
}
```

詳細については、AWS IoT SiteWise ユーザーガイド」の「[アセットの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssets](#)」の「」を参照してください。

list-associated-assets

次のコード例は、list-associated-assets を使用する方法を示しています。

AWS CLI

特定の階層内のアセットに関連付けられているすべてのアセットを一覧表示するには

次のlist-associated-assets例では、指定された風力発電所アセットに関連付けられているすべての風力タービンアセットを一覧表示します。

```
aws iotsitewise list-associated-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE
```

出力:

```
{  
  "assetSummaries": [  
    {  
      "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
      "name": "Wind Turbine 1",  
      "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "creationDate": 1575671550.0,  
      "lastUpdateDate": 1575686308.0,  
      "status": {  
        "state": "ACTIVE"  
      },  
      "hierarchies": []  
    }  
  ]  
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[特定のアセットに関連付けられたアセットの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssociatedAssets](#)」の「」を参照してください。

list-dashboards

次のコード例は、list-dashboards を使用する方法を示しています。

AWS CLI

プロジェクト内のすべてのダッシュボードを一覧表示するには

次のlist-dashboards例では、プロジェクトで定義されているすべてのダッシュボードを一覧表示します。

```
aws iotsitewise list-dashboards \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

出力:

```
{  
  "dashboardSummaries": [  
    {  
      "id": "a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE",  
      "name": "Wind Farm",  
      "creationDate": "2020-05-01T20:32:12.228476348Z",  
      "lastUpdateDate": "2020-05-01T20:32:12.228476348Z"  
    }  
  ]  
}
```

詳細については、AWS IoT SiteWise Monitor アプリケーションガイド」の「[ダッシュボードの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDashboards](#)」の「」を参照してください。

list-gateways

次の例は、list-gateways を使用する方法を説明しています。

AWS CLI

すべてのゲートウェイを一覧表示するには

次のlist-gateways例では、現在のリージョンの AWS アカウントで定義されているすべてのゲートウェイを一覧表示します。

```
aws iotsitewise list-gateways
```

出力:

```
{
  "gatewaySummaries": [
    {
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
      "gatewayName": "ExampleCorpGateway",
      "gatewayCapabilitySummaries": [
        {
          "capabilityNamespace": "iotsitewise:opcuacollector:1",
          "capabilitySyncStatus": "IN_SYNC"
        }
      ],
      "creationDate": 1588369971.457,
      "lastUpdateDate": 1588369971.457
    }
  ]
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの [「ゲートウェイを使用したデータの取り込み」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListGateways](#)」の「」を参照してください。

list-portals

次のコード例は、list-portals を使用する方法を示しています。

AWS CLI

すべてのポータルを一覧表示するには

次のlist-portals例では、現在のリージョンの AWS アカウントで定義されているすべてのポータルを一覧表示します。

```
aws iotsitewise list-portals
```

出力:

```
{
  "portalSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
      "name": "WindFarmPortal",
      "description": "A portal that contains wind farm projects for Example Corp.",
      "startUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
      "creationDate": "2020-02-04T23:01:52.90248068Z",
      "lastUpdateDate": "2020-02-04T23:01:52.90248078Z",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole"
    }
  ]
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[ポータルの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPortals](#)」の「」を参照してください。

list-project-assets

次のコード例は、list-project-assets を使用する方法を示しています。

AWS CLI

プロジェクトに関連付けられているすべてのアセットを一覧表示するには

次のlist-project-assets例では、風力発電所プロジェクトに関連付けられているすべてのアセットを一覧表示します。

```
aws iotsitewise list-projects \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE
```

出力:

```
{
  "assetIds": [
    "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
  ]
}
```

詳細については、「IoT Monitor アプリケーションガイド」の[「プロジェクトへのアセットの追加」](#)を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProjectAssets](#)」の「」を参照してください。

list-projects

次の例は、list-projects を使用する方法を説明しています。

AWS CLI

ポータル内のすべてのプロジェクトを一覧表示するには

次のlist-projects例では、ポータルで定義されているすべてのプロジェクトを一覧表示します。

```
aws iotsitewise list-projects \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

出力:

```
{
  "projectSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",
      "name": "Wind Farm 1",
      "description": "Contains asset visualizations for Wind Farm #1 for Example Corp.",
      "creationDate": "2020-02-20T21:58:43.362246001Z",
      "lastUpdateDate": "2020-02-20T21:58:43.362246095Z"
    }
  ]
}
```

詳細については、AWS IoT SiteWise Monitor アプリケーションガイド」の「[プロジェクトの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProjects](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのすべてのタグを一覧表示するには

次のlist-tags-for-resource例では、風力タービンアセットのすべてのタグを一覧表示します。

```
aws iotsitewise list-tags-for-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

出力:

```
{  
  "tags": {  
    "Owner": "richard-roe"  
  }  
}
```

詳細については、AWS IoT SiteWise ユーザーガイド」の「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-logging-options

次の例は、put-logging-options を使用する方法を説明しています。

AWS CLI

ログ記録のレベルを指定するには

次のput-logging-options例では、AWS IoT SiteWiseでのINFOレベルのログ記録を有効にします。その他のレベルには、DEBUGおよび が含まれますOFF。

```
aws iotsitewise put-logging-options \  
  --logging-options level=INFO
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT SiteWise ユーザーガイド](#)の「[Amazon CloudWatch Logs を使用したIoTのモニタリング](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLoggingOptions](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを追加

次のtag-resource例では、風力タービンアセットに所有者タグを追加します。これにより、アセットの所有者に基づいてアセットへのアクセスを制御できます。

```
aws iotsitewise tag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --tags Owner=richard-roe
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイドの「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグを削除するには

次の`untag-resource`例では、風力タービンアセットから所有者タグを削除します。

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --tag-keys Owner
```

このコマンドでは何も出力されません。

詳細については、「AWS IoT SiteWise ユーザーガイド」の「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-access-policy

次の例は、`update-access-policy` を使用する方法を説明しています。

AWS CLI

プロジェクトビューワーにプロジェクトの所有権を付与するには

次の`update-access-policy`例では、プロジェクトの所有権をプロジェクトビューワーに付与するアクセスポリシーを更新します。

```
aws iotsitewise update-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE \  
  --cli-input-json file://update-project-viewer-access-policy.json
```

`update-project-viewer-access-policy.json` の内容:

```
{  
  "accessPolicyIdentity": {  
    "user": {  
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"  
    }  
  },  
}
```

```
"accessPolicyPermission": "ADMINISTRATOR",
"accessPolicyResource": {
  "project": {
    "id": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"
  }
}
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise Monitor アプリケーションガイド」の「[プロジェクト所有者の割り当て](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAccessPolicy](#)」の「」を参照してください。

update-asset-model

次の例は、update-asset-model を使用する方法を説明しています。

AWS CLI

アセットモデルを更新するには

次のupdate-asset-model例では、風力発電所アセットモデルの説明を更新します。この例では、既存のモデルを新しいモデルでupdate-asset-model上書きするため、モデルの既存のIDsと定義が含まれています。

```
aws iotsitewise update-asset-model \
  --cli-input-json file://update-wind-farm-model.json
```

update-wind-farm-model.json の内容:

```
{
  "assetModelName": "Wind Farm Model",
  "assetModelDescription": "Represents a wind farm that comprises many wind turbines",
  "assetModelProperties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
      "name": "Region",
      "dataType": "STRING",
```

```

        "type": {
            "attribute": {}
        }
    },
    {
        "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
        "name": "Total Generated Power",
        "dataType": "DOUBLE",
        "unit": "kW",
        "type": {
            "metric": {
                "expression": "sum(power)",
                "variables": [
                    {
                        "name": "power",
                        "value": {
                            "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE",
                            "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE"
                        }
                    }
                ],
                "window": {
                    "tumbling": {
                        "interval": "1h"
                    }
                }
            }
        }
    }
],
"assetModelHierarchies": [
    {
        "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
        "name": "Wind Turbines",
        "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
]
}

```

出力:

```
{
```

```
"assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
"assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
"assetModelStatus": {
  "state": "CREATING"
}
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[アセットモデルの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAssetModel](#)」の「」を参照してください。

update-asset-property

次のコード例は、update-asset-property を使用する方法を示しています。

AWS CLI

例 1: アセットプロパティのエイリアスを更新するには

次のupdate-asset-property例では、風力タービンアセットの電力プロパティエイリアスを更新します。

```
aws iotsitewise update-asset-property \
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-55555EXAMPLE \
  --property-alias "/examplecorp/windfarm/1/turbine/1/power" \
  --property-notification-state DISABLED
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイドの「[産業用データストリームをアセットプロパティにマッピングする](#)」を参照してください。

例 2: アセットプロパティ通知を有効にするには

次のupdate-asset-property例では、風力タービンアセットの電力プロパティのアセットプロパティ更新通知を有効にします。プロパティ値の更新はMQTTトピックに発行され\$aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/

<propertyId>、各 ID はアセットプロパティのプロパティ、アセット、モデル ID に置き換えられます。

```
aws iotsitewise update-asset-property \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \  
  --property-notification-state ENABLED \  
  --property-alias "/examplecorp/windfarm/1/turbine/1/power"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイドの [「他の サービスとのやり取り」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateAssetProperty](#)」の「」を参照してください。

update-asset

次のコード例は、update-asset を使用する方法を示しています。

AWS CLI

アセット名を更新するには

次のupdate-asset例では、風力タービンアセットの名前を更新します。

```
aws iotsitewise update-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --asset-name "Wind Turbine 2"
```

出力:

```
{  
  "assetStatus": {  
    "state": "UPDATING"  
  }  
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの [「アセットの更新」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAsset](#)」の「」を参照してください。

update-dashboard

次の例は、update-dashboard を使用する方法を説明しています。

AWS CLI

ダッシュボードを更新するには

次のupdate-dashboard例では、風力発電所の合計発電量を表示するダッシュボードの折れ線グラフのタイトルを変更します。

```
aws iotsitewise update-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE \  
  --dashboard-name "Wind Farm" \  
  --dashboard-definition file://update-wind-farm-dashboard.json
```

update-wind-farm-dashboard.json の内容:

```
{  
  "widgets": [  
    {  
      "type": "monitor-line-chart",  
      "title": "Total Generated Power",  
      "x": 0,  
      "y": 0,  
      "height": 3,  
      "width": 3,  
      "metrics": [  
        {  
          "label": "Power",  
          "type": "iotsitewise",  
          "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",  
          "propertyId": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE"  
        }  
      ]  
    }  
  ]  
}
```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイドの [「ダッシュボードの作成 \(CLI\)」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateDashboard](#)」の「」を参照してください。

update-gateway-capability-configuration

次のコード例は、update-gateway-capability-configuration を使用する方法を示しています。

AWS CLI

ゲートウェイ機能を更新するには

次のupdate-gateway-capability-configuration例では、次のプロパティを使用して OPC-UA ソースを設定します。

任意の証明書を信頼します。Basic256 アルゴリズムを使用してメッセージを保護します。SignAndEncrypt モードを使用して接続を保護します。AWS Secrets Manager シークレットに保存されている認証情報を使用します。

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --capability-namespace "iotsitewise:opcuacollector:1" \  
  --capability-configuration file://opc-ua-capability-configuration.json
```

opc-ua-capability-configuration.json の内容:

```
{  
  "sources": [  
    {  
      "name": "Wind Farm #1",  
      "endpoint": {  
        "certificateTrust": {  
          "type": "TrustAny"  
        },  
        "endpointUri": "opc.tcp://203.0.113.0:49320",  
        "securityPolicy": "BASIC256",  
      }  
    }  
  ]  
}
```

```

        "messageSecurityMode": "SIGN_AND_ENCRYPT",
        "identityProvider": {
            "type": "Username",
            "usernameSecretArn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:green-grass-windfarm1-auth-1ABCDE"
        },
        "nodeFilterRules": []
    },
    "measurementDataStreamPrefix": ""
}
]
}

```

出力:

```

{
  "capabilityNamespace": "iotsitewise:opcuacollector:1",
  "capabilitySyncStatus": "OUT_OF_SYNC"
}

```

詳細については、「IoT ユーザーガイド」の「[データソースの設定](#)」を参照してください。

AWS IoT SiteWise

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateGatewayCapabilityConfiguration](#)」の「」を参照してください。

update-gateway

次のコード例は、update-gateway を使用する方法を示しています。

AWS CLI

ゲートウェイの名前を更新するには

次のupdate-gateway例では、ゲートウェイの名前を更新します。

```

aws iotsitewise update-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --gateway-name ExampleCorpGateway1

```

このコマンドでは何も出力されません。

詳細については、AWS IoT SiteWise ユーザーガイドの「[ゲートウェイを使用したデータの取り込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateGateway](#)」の「」を参照してください。

update-portal

次の例は、update-portal を使用する方法を説明しています。

AWS CLI

ポータルの詳細を更新するには

次のupdate-portal例では、風力発電所会社のウェブポータルを更新します。

```
aws iotsitewise update-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \  
  --portal-name WindFarmPortal \  
  --portal-description "A portal that contains wind farm projects for Example Corp." \  
  --portal-contact-email support@example.com \  
  --role-arn arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole
```

出力:

```
{  
  "portalStatus": {  
    "state": "UPDATING"  
  }  
}
```

詳細については、AWS IoT SiteWise ユーザーガイドの「[ポータルの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePortal](#)」の「」を参照してください。

update-project

次のコード例は、update-project を使用する方法を示しています。

AWS CLI

プロジェクトの詳細を更新するには

次のupdate-project例では、風力発電所プロジェクトを更新します。

```
aws iotsitewise update-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --project-name "Wind Farm 1" \  
  --project-description "Contains asset visualizations for Wind Farm #1 for Example Corp."
```

このコマンドでは何も出力されません。

詳細については、「IoT Monitor アプリケーションガイド」の「[プロジェクトの詳細の変更](#)」を参照してください。AWS IoT SiteWise

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateProject](#)」の「」を参照してください。

AWS IoT Things Graph を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Things Graph。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-entity-to-thing

次の例は、associate-entity-to-thing を使用する方法を説明しています。

AWS CLI

モノをデバイスに関連付けるには

次のassociate-entity-to-thing例では、モノをデバイスに関連付けられています。この例では、パブリック名前空間にあるモーションセンサーデバイスを使用しています。

```
aws iotthingsgraph associate-entity-to-thing \  
  --thing-name "MotionSensorName" \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

このコマンドでは何も出力されません。

詳細については、AWS IoT Things Graph ユーザーガイドの「[モデルの作成とアップロード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateEntityToThing](#)」の「」を参照してください。

create-flow-template

次の例は、create-flow-template を使用する方法を説明しています。

AWS CLI

フローを作成するには

次のcreate-flow-template例では、フロー (ワークフロー) を作成します。の値はMyFlowDefinition、フローをモデル化する GraphQL です。

```
aws iotthingsgraph create-flow-template \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```

出力:

```
{  
  "summary": {  
    "createdAt": 1559248067.545,  
    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",  
    "revisionNumber": 1  
  }  
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の「[フローの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFlowTemplate](#)」の「」を参照してください。

create-system-instance

次の例は、create-system-instance を使用する方法を説明しています。

AWS CLI

システムインスタンスを作成するには

次のcreate-system-instance例では、システムインスタンスを作成します。の値は、システムインスタンスをモデル化する GraphQL MySystemInstanceDefinitionです。

```
aws iotthingsgraph create-system-instance -\
  -definition language=GRAPHQL,text="MySystemInstanceDefinition" \
  --target CLLOUD \
  --flow-actions-role-arn myRoleARN
```

出力:

```
{
  "summary": {
    "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/Room218",
    "status": "NOT_DEPLOYED",
    "target": "CLLOUD",
    "createdAt": 1559249315.208,
    "updatedAt": 1559249315.208
  }
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の「[システムとフロー設定の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSystemInstance](#)」の「」を参照してください。

create-system-template

次のコード例は、create-system-template を使用する方法を示しています。

AWS CLI

システムを作成するには

次のcreate-system-template例では、システムを作成します。の値は MySystemDefinition、システムをモデル化する GraphQL です。

```
aws iotthingsgraph create-system-template \  
  --definition language=GraphQL,text="MySystemDefinition"
```

出力:

```
{  
  "summary": {  
    "createdAt": 1559249776.254,  
    "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",  
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/  
MySystem",  
    "revisionNumber": 1  
  }  
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の [「システムの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSystemTemplate](#)」の「」を参照してください。

delete-flow-template

次のコード例は、delete-flow-template を使用する方法を示しています。

AWS CLI

フローを削除するには

次のdelete-flow-template例では、フロー (ワークフロー) を削除します。

```
aws iotthingsgraph delete-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT Things Graph ユーザーガイド」の「IoT Things Graph エンティティ、フロー、システム、デプロイのライフサイクル管理」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteFlowTemplate](#)」の「」を参照してください。

delete-namespace

次の例は、delete-namespace を使用する方法を説明しています。

AWS CLI

名前空間を削除するには

次のdelete-namespace例では、名前空間を削除します。

```
aws iotthingsgraph delete-namespace
```

出力:

```
{  
  "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",  
  "namespaceName": "us-west-2/123456789012/default"  
}
```

詳細については、[AWS「IoT Things Graph ユーザーガイド」の「IoT Things Graph エンティティ、フロー、システム、デプロイのライフサイクル管理」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteNamespace](#)」の「」を参照してください。

delete-system-instance

次の例は、delete-system-instance を使用する方法を説明しています。

AWS CLI

システムインスタンスを削除するには

次のdelete-system-instance例では、システムインスタンスを削除します。

```
aws iotthingsgraph delete-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT Things Graph ユーザーガイド」の「IoT Things Graph エンティティ、フロー、システム、デプロイのライフサイクル管理」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSystemInstance](#)」の「」を参照してください。

delete-system-template

次の例は、delete-system-template を使用する方法を説明しています。

AWS CLI

システムを削除するには

次のdelete-system-template例では、システムを削除します。

```
aws iotthingsgraph delete-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT Things Graph ユーザーガイド」の「IoT Things Graph エンティティ、フロー、システム、デプロイのライフサイクル管理」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSystemTemplate](#)」の「」を参照してください。

deploy-system-instance

次の例は、deploy-system-instance を使用する方法を説明しています。

AWS CLI

システムインスタンスをデプロイするには

次のdelete-system-template例では、システムインスタンスをデプロイします。

```
aws iotthingsgraph deploy-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

出力:

```
{  
  "summary": {  
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment:Room218",  
    "createdAt": 1559249776.254,  
    "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",  
    "status": "DEPLOYED_IN_TARGET",  
    "target": "CLOUD",  
    "updatedAt": 1559249776.254  
  }  
}
```

詳細については、AWS「IoT Things Graph ユーザーガイド」の[「システムとフロー設定の操作」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeploySystemInstance](#)」の「」を参照してください。

deprecate-flow-template

次のコード例は、deprecate-flow-template を使用する方法を示しています。

AWS CLI

フローを非推奨にするには

次のdeprecate-flow-template例では、フロー (ワークフロー) を廃止しています。

```
aws iotthingsgraph deprecate-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT Things Graph ユーザーガイド」の「IoT Things Graph エンティティ、フロー、システム、デプロイのライフサイクル管理」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeprecateFlowTemplate](#)」の「」を参照してください。

deprecate-system-template

次の例は、deprecate-system-template を使用する方法を説明しています。

AWS CLI

システムを非推奨にするには

次のdeprecate-system-template例では、システムを非推奨にします。

```
aws iotthingsgraph deprecate-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT Things Graph ユーザーガイド」の「IoT Things Graph エンティティ、フロー、システム、デプロイのライフサイクル管理」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DeprecateSystemTemplate](#)」の「」を参照してください。

describe-namespace

次の例は、describe-namespace を使用する方法を説明しています。

AWS CLI

名前空間の説明を取得するには

次のdescribe-namespace例では、名前空間の説明を取得します。

```
aws iotthingsgraph describe-namespace
```

出力:

```
{
  "namespaceName": "us-west-2/123456789012/default",
  "trackingNamespaceName": "aws",
  "trackingNamespaceVersion": 1,
  "namespaceVersion": 5
}
```

詳細については、[「IoT Things Graph ユーザーガイド」の「名前空間」](#)を参照してください。

AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNamespace](#)」の「」を参照してください。

dissociate-entity-from-thing

次のコード例は、`dissociate-entity-from-thing` を使用する方法を示しています。

AWS CLI

モノをデバイスから関連付け解除するには

次の`dissociate-entity-from-thing`例では、デバイスからモノの関連付けを解除します。

```
aws iotthingsgraph dissociate-entity-from-thing \
  --thing-name "MotionSensorName" \
  --entity-type "DEVICE"
```

このコマンドでは何も出力されません。

詳細については、[「IoT Things Graph ユーザーガイド」の「モデルの作成とアップロード」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[DissociateEntityFromThing](#)」の「」を参照してください。

get-entities

次のコード例は、`get-entities` を使用する方法を示しています。

AWS CLI

エンティティの定義を取得するには

次のget-entities例では、デバイスモデルの定義を取得します。

```
aws iotthingsgraph get-entities \  
  --ids "urn:tdm:aws/examples:DeviceModel:MotionSensor"
```

出力:

```
{  
  "descriptions": [  
    {  
      "id": "urn:tdm:aws/examples:DeviceModel:MotionSensor",  
      "type": "DEVICE_MODEL",  
      "createdAt": 1559256190.599,  
      "definition": {  
        "language": "GRAPHQL",  
        "text": "##\n# Specification of motion sensor devices interface.\n##  
\n#type MotionSensor @deviceModel(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor  
\",\n      capability: \"urn:tdm:aws/examples:capability:MotionSensorCapability\")  
      {ignore:void}"  
      }  
    }  
  ]  
}
```

詳細については、AWS IoT Things Graph ユーザーガイドの「[モデルの作成とアップロード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEntities](#)」の「」を参照してください。

get-flow-template-revisions

次の例は、get-flow-template-revisions を使用する方法を説明しています。

AWS CLI

フローに関するリビジョン情報を取得するには

次のget-flow-template-revisions例では、フロー (ワークフロー) に関するリビジョン情報を取得します。

```
aws iotthingsgraph get-flow-template-revisions \  
  --id urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow
```

出力:

```
{  
  "summaries": [  
    {  
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",  
      "revisionNumber": 1,  
      "createdAt": 1559247540.292  
    }  
  ]  
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の「[フローの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFlowTemplateRevisions](#)」の「」を参照してください。

get-flow-template

次のコード例は、get-flow-template を使用する方法を示しています。

AWS CLI

フロー定義を取得するには

次のget-flow-template例では、フロー (ワークフロー) の定義を取得します。

```
aws iotthingsgraph get-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

出力:

```
{
```



```

    "description": {
      "summary": {
        "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
        "revisionNumber": 1,
        "createdAt": 1559247540.292
      },
      "definition": {
        "language": "GraphQL",
        "text": "{\nquery MyFlow($camera: string!, $screen: string!)
@workflowType(id: \"urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow\")
@annotation(type: \"tgc:FlowEvent\", id: \"sledged790c1b2bcd949e09da0c9bfc077f79d
\", x: 1586, y: 653) @triggers(definition: \"{MotionSensor(description:
\\\\\"\\\\\") @position(x: 1045, y: 635.6666564941406) {\n condition(expr:
\\\\\"devices[name == \\\\\\\\\\\\\\"motionSensor\\\\\\\\\\\\\\\\\\\\\"]}.events[name == \\\\
\\\\\\\\\"StateChanged\\\\\\\\\\\\\\\\\\\\\"].lastEvent\\\\\\\\\")\\\\\n action(expr: \\\\\"\\\\\")\\\\
\n}}\") {\n variables {\n cameraResult @property(id: \"urn:tdm:aws/
examples:property:CameraStateProperty\")\n }\n steps {\n step(name: \"Camera
\", outEvent: [\"sledged790c1b2bcd949e09da0c9bfc077f79d\"] @position(x: 1377,
y: 638.6666564941406) {\n DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Camera\", out: \"cameraResult\", deviceId: \"${camera}\")
{\n capture\n }\n }\n step(name: \"Screen\", inEvent:
[\"sledged790c1b2bcd949e09da0c9bfc077f79d\"] @position(x: 1675.6666870117188,
y: 637.9999847412109) {\n DeviceActivity(deviceModel: \"urn:tdm:aws/
examples:deviceModel:Screen\", deviceId: \"${screen}\") {\n display(imageUrl:
\"${cameraResult.lastClickedImage}\")\n }\n }\n }\n }\n }
\",
        "validatedNamespaceVersion": 5
      }
    }
  }
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の「[フローの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFlowTemplate](#)」の「」を参照してください。

get-namespace-deletion-status

次の例は、get-namespace-deletion-status を使用する方法を説明しています。

AWS CLI

名前空間削除タスクのステータスを取得するには

次のget-namespace-deletion-status例では、名前空間削除タスクのステータスを取得します。

```
aws iotthingsgraph get-namespace-deletion-status
```

出力:

```
{
  "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
  "namespaceName": "us-west-2/123456789012/default"
  "status": "SUCCEEDED "
}
```

詳細については、[「IoT Things Graph ユーザーガイド」の「名前空間」](#)を参照してください。

AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[GetNamespaceDeletionStatus](#)」の「」を参照してください。

get-system-instance

次のコード例は、get-system-instance を使用する方法を示しています。

AWS CLI

システムインスタンスを取得するには

次のget-system-instance例では、システムインスタンスの定義を取得します。

```
aws iotthingsgraph get-system-instance \
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

出力:

```
{
  "description": {
    "summary": {
      "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/Room218",

```

```

        "status": "NOT_DEPLOYED",
        "target": "CLOUD",
        "createdAt": 1559249315.208,
        "updatedAt": 1559249315.208
    },
    "definition": {
        "language": "GRAPHQL",
        "text": "{\r\nquery Room218 @deployment(id: \"urn:tdm:us-west-2/123456789012/default:Deployment:Room218\", systemId: \"urn:tdm:us-west-2/123456789012/default:System:SecurityFlow\") {\r\n    motionSensor(deviceId: \"MotionSensorName\")\r\n    screen(deviceId: \"ScreenName\")\r\n    camera(deviceId: \"CameraName\") \r\n    triggers {MotionEventTrigger(description: \"a trigger\") { \r\n        condition(expr: \"devices[name == 'motionSensor'].events[name == 'StateChanged'].lastEvent\") \r\n        action(expr: \"ThingsGraph.startFlow('SecurityFlow', bindings[name == 'camera'].deviceId, bindings[name == 'screen'].deviceId))\r\n    } \r\n    } \r\n    } \r\n    }"
    },
    "metricsConfiguration": {
        "cloudMetricEnabled": false
    },
    "validatedNamespaceVersion": 5,
    "flowActionsRoleArn": "arn:aws:iam::123456789012:role/ThingsGraphRole"
}
}

```

詳細については、AWS「IoT Things Graph ユーザーガイド」の「[システムおよびフロー設定の操作](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetSystemInstance](#)」の「」を参照してください。

get-system-template-revisions

次の例は、get-system-template-revisions を使用する方法を説明しています。

AWS CLI

システムに関するリビジョン情報を取得するには

次のget-system-template-revisions例では、システムに関するリビジョン情報を取得します。

```
aws iotthingsgraph get-system-template-revisions \
```

```
--id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

出力:

```
{
  "summaries": [
    {
      "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/MySystem",
      "revisionNumber": 1,
      "createdAt": 1559247540.656
    }
  ]
}
```

詳細については、AWS「IoT Things Graph ユーザーガイド」の[「システムとフロー設定の操作」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetSystemTemplateRevisions](#)」の「」を参照してください。

get-system-template

次のコード例は、get-system-template を使用する方法を示しています。

AWS CLI

システムを取得するには

次のget-system-template例では、システムの定義を取得します。

```
aws iotthingsgraph get-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

出力:

```
{
  "description": {
    "summary": {
      "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
```

```

    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/
MyFlow",
    "revisionNumber": 1,
    "createdAt": 1559247540.656
  },
  "definition": {
    "language": "GRAPHQL",
    "text": "{\n  type MySystem @systemType(id: \"urn:tdm:us-
west-2/123456789012/default:System:MySystem\", description: \"\") {\n    camera:
Camera @thing(id: \"urn:tdm:aws/examples:deviceModel:Camera\")\n    screen:
Screen @thing(id: \"urn:tdm:aws/examples:deviceModel:Screen\")\n    motionSensor:
MotionSensor @thing(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor
\")\n    MyFlow: MyFlow @workflow(id: \"urn:tdm:us-west-2/123456789012/
default:Workflow:MyFlow\")\n  }\n}"
  },
  "validatedNamespaceVersion": 5
}
}

```

詳細については、AWS「IoT Things Graph ユーザーガイド」の[「システムとフロー設定の操作」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetSystemTemplate](#)」の「」を参照してください。

get-upload-status

次の例は、get-upload-status を使用する方法を説明しています。

AWS CLI

エンティティアップロードのステータスを取得するには

次のget-upload-status例では、エンティティアップロードオペレーションのステータスを取得します。の値はMyUploadId、upload-entity-definitions オペレーションによって返される ID 値です。

```

aws iotthingsgraph get-upload-status \
  --upload-id "MyUploadId"

```

出力:

```
{
  "namespaceName": "us-west-2/123456789012/default",
  "namespaceVersion": 5,
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
  "uploadStatus": "SUCCEEDED"
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の「[エンティティのモデリング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUploadStatus](#)」の「」を参照してください。

list-flow-execution-messages

次の例は、list-flow-execution-messages を使用する方法を説明しています。

AWS CLI

フロー実行のイベントに関する情報を取得するには

次のlist-flow-execution-messages例では、フロー実行のイベントに関する情報を取得します。

```
aws iotthingsgraph list-flow-execution-messages \
  --flow-execution-id "urn:tdm:us-west-2/123456789012/
  default:Workflow:SecurityFlow_2019-05-11T19:39:55.317Z_MotionSensor_69b151ad-
  a611-42f5-ac21-fe537f9868ad"
```

出力:

```
{
  "messages": [
    {
      "eventType": "EXECUTION_STARTED",
      "messageId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
      "payload": "Flow execution started",
      "timestamp": 1559247540.656
    }
  ]
}
```

詳細については、AWS「IoT Things Graph ユーザーガイド」の「[フローの使用](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListFlowExecutionMessages](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのすべてのタグを一覧表示するには

次のlist-tags-for-resource例では、AWS IoT Things Graph リソースのすべてのタグを一覧表示します。

```
aws iotthingsgraph list-tags-for-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
  default/Room218"
```

出力:

```
{  
  "tags": [  
    {  
      "key": "Type",  
      "value": "Residential"  
    }  
  ]  
}
```

詳細については、[AWS IoT Things Graph ユーザーガイド](#)の「IoT Things Graph リソースのタグ付け」を参照してください。AWS IoT

- API 詳細については、AWS CLI「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

search-entities

次の例は、search-entities を使用する方法を説明しています。

AWS CLI

エンティティを検索するには

次のsearch-entities例では、タイプ のすべてのエンティティを検索しますEVENT。

```
aws iotthingsgraph search-entities \  
  --entity-types "EVENT"
```

出力:

```
{  
  "descriptions": [  
    {  
      "id": "urn:tdm:aws/examples:Event:MotionSensorEvent",  
      "type": "EVENT",  
      "definition": {  
        "language": "GRAPHQL",  
        "text": "##\n# Description of events emitted by motion  
sensor.\n##\n\ntype MotionSensorEvent @eventType(id: \"urn:tdm:aws/  
examples:event:MotionSensorEvent\", \n          payload: \"urn:tdm:aws/  
examples:property:MotionSensorStateProperty\") {ignore:void}"  
      }  
    },  
    {  
      "id": "urn:tdm:us-west-2/123456789012/  
default:Event:CameraClickedEventV2",  
      "type": "EVENT",  
      "definition": {  
        "language": "GRAPHQL",  
        "text": "type CameraClickedEventV2 @eventType(id: \"urn:tdm:us-  
west-2/123456789012/default:event:CameraClickedEventV2\", \r\npayload:  
\"urn:tdm:aws:Property:Boolean\") {ignore:void}"  
      }  
    },  
    {  
      "id": "urn:tdm:us-west-2/123456789012/  
default:Event:MotionSensorEventV2",  
      "type": "EVENT",  
      "definition": {  
        "language": "GRAPHQL",  
        "text": "# Event emitted by the motion sensor.\r\n\ntype  
MotionSensorEventV2 @eventType(id: \"urn:tdm:us-west-2/123456789012/"
```



```

default:event:MotionSensorEventV2\", \\r\\npayload: \\\"urn:tdm:us-west-2/123456789012/
default:property:MotionSensorStateProperty2\\\") {ignore:void}\"
    }
  }
],
  \"nextToken\": \"urn:tdm:us-west-2/123456789012/default:Event:MotionSensorEventV2\"
}

```

詳細については、[AWS 「IoT Things Graph ユーザーガイド」の「IoT Things Graph データモデルリファレンス」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchEntities](#)」の「」を参照してください。

search-flow-executions

次のコード例は、search-flow-executions を使用する方法を示しています。

AWS CLI

フロー実行を検索するには

次のsearch-flow-executions例では、指定されたシステムインスタンス内のフローのすべての実行を検索します。

```

aws iotthingsgraph search-flow-executions \\
  --system-instance-id \"urn:tdm:us-west-2/123456789012/default:Deployment:Room218\"

```

出力:

```

{
  \"summaries\": [
    {
      \"createdAt\": 1559247540.656,
      \"flowExecutionId\": \"f6294f1e-b109-4bbe-9073-f451a2dda2da\",
      \"flowTemplateId\": \"urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow\",
      \"status\": \"RUNNING \",
      \"systemInstanceId\": \"urn:tdm:us-west-2/123456789012/
default:System:MySystem\",
      \"updatedAt\": 1559247540.656
    }
  ]
}

```

```
}
```

詳細については、AWS「IoT Things Graph ユーザーガイド」の「[システムとフロー設定の操作](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[SearchFlowExecutions](#)」の「」を参照してください。

search-flow-templates

次のコード例は、search-flow-templates を使用する方法を示しています。

AWS CLI

フロー (またはワークフロー) を検索するには

次のsearch-flow-templates例では、カメラデバイスモデルを含むすべてのフロー (ワークフロー) を検索します。

```
aws iotthingsgraph search-flow-templates \  
  --filters name="DEVICE_MODEL_ID",value="urn:tdm:aws/examples:DeviceModel:Camera"
```

出力:

```
{  
  "summaries": [  
    {  
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",  
      "revisionNumber": 1,  
      "createdAt": 1559247540.292  
    },  
    {  
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:SecurityFlow",  
      "revisionNumber": 3,  
      "createdAt": 1548283099.27  
    }  
  ]  
}
```

詳細については、AWS IoT Things Graph ユーザーガイド」の「[フローの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SearchFlowTemplates](#)」の「」を参照してください。

search-system-instances

次のコード例は、search-system-instances を使用する方法を示しています。

AWS CLI

システムインスタンスを検索するには

次のsearch-system-instances例では、指定されたシステムを含むすべてのシステムインスタンスを検索します。

```
aws iotthingsgraph search-system-instances \  
  --filters name="SYSTEM_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/  
  default:System:SecurityFlow"
```

出力:

```
{  
  "summaries": [  
    {  
      "id": "urn:tdm:us-west-2/123456789012/  
default:Deployment:DeploymentForSample",  
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/DeploymentForSample",  
      "status": "NOT_DEPLOYED",  
      "target": "GREENGRASS",  
      "greengrassGroupName": "ThingsGraphGrnGr",  
      "createdAt": 1555716314.707,  
      "updatedAt": 1555716314.707  
    },  
    {  
      "id": "urn:tdm:us-west-2/123456789012/  
default:Deployment:MockDeployment",  
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/MockDeployment",  
      "status": "DELETED_IN_TARGET",  
      "target": "GREENGRASS",  
      "greengrassGroupName": "ThingsGraphGrnGr",  
      "createdAt": 1549416462.049,  
    }  
  ]  
}
```

```

      "updatedAt": 1549416722.361,
      "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
      "greengrassGroupVersionId": "7365aed7-2d3e-4d13-aad8-75443d45eb05"
    },
    {
      "id": "urn:tdm:us-west-2/123456789012/
default:Deployment:MockDeployment2",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/MockDeployment2",
      "status": "DEPLOYED_IN_TARGET",
      "target": "GREENGRASS",
      "greengrassGroupName": "ThingsGraphGrnGr",
      "createdAt": 1549572385.774,
      "updatedAt": 1549572418.408,
      "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",
      "greengrassGroupVersionId": "bfa70ab3-2bf7-409c-a4d4-bc8328ae5b86"
    },
    {
      "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room215",
      "status": "NOT_DEPLOYED",
      "target": "GREENGRASS",
      "greengrassGroupName": "ThingsGraphGG",
      "createdAt": 1547056918.413,
      "updatedAt": 1547056918.413
    },
    {
      "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room218",
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218",
      "status": "NOT_DEPLOYED",
      "target": "CLOUD",
      "createdAt": 1559249315.208,
      "updatedAt": 1559249315.208
    }
  ]
}

```

詳細については、AWS「IoT Things Graph ユーザーガイド」の「[システムとフロー設定の操作](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[SearchSystemInstances](#)」の「」を参照してください。

search-system-templates

次のコード例は、search-system-templates を使用する方法を示しています。

AWS CLI

システムを検索するには

次のsearch-system-templates例では、指定されたフローを含むすべてのシステムを検索します。

```
aws iotthingsgraph search-system-templates \  
  --filters name="FLOW_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/  
  default:Workflow:SecurityFlow"
```

出力:

```
{  
  "summaries": [  
    {  
      "id": "urn:tdm:us-west-2/123456789012/default:System:SecurityFlow",  
      "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/  
SecurityFlow",  
      "revisionNumber": 1,  
      "createdAt": 1548283099.433  
    }  
  ]  
}
```

詳細については、AWS「IoT Things Graph ユーザーガイド」の「[フローの使用](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[SearchSystemTemplates](#)」の「」を参照してください。

search-things

次の例は、search-things を使用する方法を説明しています。

AWS CLI

デバイスとデバイスモデルに関連付けられたモノを検索するには

次のsearch-things例では、HCSR501 MotionSensor デバイスに関連付けられているすべてのものを検索します。

```
aws iotthingsgraph search-things \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

出力:

```
{  
  "things": [  
    {  
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MotionSensor1",  
      "thingName": "MotionSensor1"  
    },  
    {  
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/TG_MS",  
      "thingName": "TG_MS"  
    }  
  ]  
}
```

詳細については、AWS IoT Things Graph ユーザーガイドの「[モデルの作成とアップロード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchThings](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを作成するには

次のtag-resource例では、指定されたリソースのタグを作成します。

```
aws iotthingsgraph tag-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218" \  
  --tags key="Type",value="Residential"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT Things Graph ユーザーガイド」の「IoT Things Graph リソースのタグ付け」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

undeploy-system-instance

次のコード例は、undeploy-system-instance を使用する方法を示しています。

AWS CLI

ターゲットからシステムインスタンスをデプロイ解除するには

次のundeploy-system-instance例では、ターゲットからシステムインスタンスを削除します。

```
aws iotthingsgraph undeploy-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room215"
```

出力:

```
{  
  "summary": {  
    "id": "urn:tdm:us-west-2/123456789012/default:Deployment:Room215",  
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/  
Room215",  
    "status": "PENDING_DELETE",  
    "target": "GREENGRASS",  
    "greengrassGroupName": "ThingsGraphGrnGr",  
    "createdAt": 1553189694.255,  
    "updatedAt": 1559344549.601,  
    "greengrassGroupId": "01d04b07-2a51-467f-9d03-0c90b3cdcaaf",  
    "greengrassGroupVersionId": "731b371d-d644-4b67-ac64-3934e99b75d7"  
  }  
}
```

詳細については、[AWS 「IoT Things Graph ユーザーガイド」の「IoT Things Graph エンティティ、フロー、システム、デプロイのライフサイクル管理」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UndeploySystemInstance](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを削除するには

次のuntag-resource例では、指定されたリソースのタグを削除します。

```
aws iotthingsgraph untag-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218" \  
  --tag-keys "Type"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT Things Graph ユーザーガイド](#)の「IoT Things Graph リソースのタグ付け」を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-flow-template

次の例は、update-flow-template を使用する方法を説明しています。

AWS CLI

フローを更新するには

次のupdate-flow-template例では、フロー (ワークフロー) を更新します。の値はMyFlowDefinition、フローをモデル化する GraphQL です。

```
aws iotthingsgraph update-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow" \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```

出力:


```
{
  "summary": {
    "createdAt": 1559248067.545,
    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
    "revisionNumber": 2
  }
}
```

詳細については、AWS「IoT Things Graph ユーザーガイド」の「[フローの使用](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateFlowTemplate](#)」の「」を参照してください。

update-system-template

次の例は、update-system-template を使用する方法を説明しています。

AWS CLI

システムを更新するには

次のupdate-system-template例では、システムを更新します。の値はMySystemDefinition、システムをモデル化する GraphQL です。

```
aws iotthingsgraph update-system-template \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem" \
  --definition language=GraphQL,text="MySystemDefinition"
```

出力:

```
{
  "summary": {
    "createdAt": 1559249776.254,
    "id": "urn:tdm:us-west-2/123456789012/default:System:MySystem",
    "arn": "arn:aws:iotthingsgraph:us-west-2:123456789012:System/default/MySystem",
    "revisionNumber": 2
  }
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の [「システムの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateSystemTemplate](#)」の「」を参照してください。

upload-entity-definitions

次のコード例は、upload-entity-definitions を使用する方法を示しています。

AWS CLI

エンティティ定義をアップロードするには

次のupload-entity-definitions例では、エンティティ定義を名前空間にアップロードします。の値はMyEntityDefinitions、エンティティをモデル化する GraphQL です。

```
aws iotthingsgraph upload-entity-definitions \  
  --document language=GRAPHQL,text="MyEntityDefinitions"
```

出力:

```
{  
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"  
}
```

詳細については、AWS 「IoT Things Graph ユーザーガイド」の [「エンティティのモデリング」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UploadEntityDefinitions](#)」の「」を参照してください。

AWS IoT Wireless を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS IoT Wireless。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-aws-account-with-partner-account

次の例は、associate-aws-account-with-partner-account を使用する方法を説明しています。

AWS CLI

パートナーアカウントを AWS アカウントに関連付けるには

次のassociate-aws-account-with-partner-account例では、次の Sidewalk アカウントの認証情報を AWS アカウントに関連付けます。

```
aws iotwireless associate-aws-account-with-partner-account \  
  --sidewalk  
  AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78
```

出力:

```
{  
  "Sidewalk": {  
    "AmazonId": "12345678901234",  
    "AppServerPrivateKey":  
    "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"  
  }  
}
```

詳細については、[AWS IoT デベロッパーガイド](#)の「[Amazon Sidewalk Integration for IoT Core](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス AssociateAwsAccountWithPartnerAccount](#)」の「」を参照してください。

associate-wireless-device-with-thing

次のコード例は、`associate-wireless-device-with-thing` を使用する方法を示しています。

AWS CLI

モノをワイヤレスデバイスに関連付けるには

次の`associate-wireless-device-with-thing`例では、指定された ID を持つワイヤレスデバイスにモノを関連付けます。

```
aws iotwireless associate-wireless-device-with-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core にゲートウェイとワイヤレスデバイスを追加する LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateWirelessDeviceWithThing](#)」の「」を参照してください。

associate-wireless-gateway-with-certificate

次の例は、`associate-wireless-gateway-with-certificate` を使用する方法を説明しています。

AWS CLI

証明書をワイヤレスゲートウェイに関連付けるには

以下は、ワイヤレスゲートウェイを証明書に`associate-wireless-gateway-with-certificate`関連付けます。

```
aws iotwireless associate-wireless-gateway-with-certificate \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --iot-certificate-  
  id "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

出力:

```
{
  "IotCertificateId":
  "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
}
```

詳細については、[AWS IoT デベロッパーガイド](#)の「[の IoT Core にゲートウェイとワイヤレス デバイスを追加する LoRaWAN AWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス AssociateWirelessGatewayWithCertificate](#)」の「」を参照してください。

associate-wireless-gateway-with-thing

次のコード例は、`associate-wireless-gateway-with-thing` を使用する方法を示しています。

AWS CLI

モノをワイヤレスゲートウェイに関連付けるには

次の`associate-wireless-gateway-with-thing`例では、モノをワイヤレスゲートウェイに関連付けます。

```
aws iotwireless associate-wireless-gateway-with-thing \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
  --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「[の IoT Core にゲートウェイとワイヤレス デバイスを追加する LoRaWAN AWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス AssociateWirelessGatewayWithThing](#)」の「」を参照してください。

create-destination

次のコード例は、`create-destination` を使用する方法を示しています。

AWS CLI

IoT ワイヤレス送信先を作成するには

次のcreate-destination例では、デバイスメッセージを AWS IoT ルールにマッピングするための送信先を作成します。このコマンドを実行する前に、AWS IoT ルールにデータを送信するために必要なアクセス許可を LoRaWAN AWS IoT Core に付与するIAMロールを作成しておく必要があります。

```
aws iotwireless create-destination \  
  --name IoTWirelessDestination \  
  --expression-type RuleName \  
  --expression IoTWirelessRule \  
  --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

出力:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoTWirelessDestination",  
  "Name": "IoTWirelessDestination"  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT Core への送信先の追加 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDestination](#)」の「」を参照してください。

create-device-profile

次のコード例は、create-device-profile を使用する方法を示しています。

AWS CLI

新しいデバイスプロフィールを作成するには

次のcreate-device-profile例では、新しい IoT ワイヤレスデバイスプロフィールを作成します。

```
aws iotwireless create-device-profile
```

出力:

```
{
```

```
"Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

詳細については、[AWS「IoT デベロッパーガイド」の「の IoT Core にプロフィールを追加する LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [CreateDeviceProfile](#)」の「」を参照してください。

create-service-profile

次のコード例は、create-service-profile を使用する方法を示しています。

AWS CLI

新しいサービスプロフィールを作成するには

次のcreate-service-profile例では、新しい IoT ワイヤレスサービスプロフィールを作成します。

```
aws iotwireless create-service-profile
```

出力:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

詳細については、[AWS「IoT デベロッパーガイド」の「の IoT Core にプロフィールを追加する LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [CreateServiceProfile](#)」の「」を参照してください。

create-wireless-device

次のコード例は、create-wireless-device を使用する方法を示しています。

AWS CLI

IoT ワイヤレスデバイスを作成するには

次のcreate-wireless-device例では、タイプのワイヤレスデバイスリソースを作成します LoRaWAN。

```
aws iotwireless create-wireless-device \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "Description": "My LoRaWAN wireless device"  
  "DestinationName": "IoTWirelessDestination"  
  "LoRaWAN": {  
    "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",  
    "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",  
    "OtaaV1_1": {  
      "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",  
      "JoinEui": "b4c231a359bc2e3d",  
      "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"  
    },  
    "DevEui": "ac12efc654d23fc2"  
  },  
  "Name": "SampleIoTWirelessThing"  
  "Type": LoRaWAN  
}
```

出力:

```
{  
  "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",  
  "Id": "1ffd32c8-8130-4194-96df-622f072a315f"  
}
```

詳細については、[AWS 「IoT デベロッパガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateWirelessDevice](#)」の「」を参照してください。

create-wireless-gateway-task-definition

次のコード例は、create-wireless-gateway-task-definition を使用方法を示しています。

AWS CLI

ワイヤレスゲートウェイタスク定義を作成するには

以下では、指定した最新バージョンのすべてのゲートウェイに対して、このタスク定義を使用してタスクcreate-wireless-gateway-task-definitionを自動的に作成します。

```
aws iotwireless create-wireless-gateway-task-definition \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "AutoCreateTasks": true,  
  "Name": "TestAutoUpdate",  
  "Update": {  
    "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",  
    "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",  
    "LoRaWAN" : {  
      "CurrentVersion" : {  
        "PackageVersion" : "1.0.0",  
        "Station" : "2.0.5",  
        "Model" : "linux"  
      },  
      "UpdateVersion" : {  
        "PackageVersion" : "1.0.1",  
        "Station" : "2.0.5",  
        "Model" : "minihub"  
      }  
    }  
  }  
}
```

出力:

```
{
```

```
"Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateWirelessGatewayTaskDefinition](#)」の「」を参照してください。

create-wireless-gateway-task

次のコード例は、create-wireless-gateway-task を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイのタスクを作成するには

次のcreate-wireless-gateway-task例では、ワイヤレスゲートウェイのタスクを作成します。

```
aws iotwireless create-wireless-gateway-task \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
  --wireless-gateway-task-definition-id "aa000102-0304-b0cd-ef56-a1b23cde456a"
```

出力:

```
{
  "WirelessGatewayTaskDefinitionId": "aa204003-0604-30fb-ac82-a4f95aaf450a",
  "Status": "Success"
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateWirelessGatewayTask](#)」の「」を参照してください。

create-wireless-gateway

次のコード例は、create-wireless-gateway を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイを作成するには

次のcreate-wireless-gateway例では、ワイヤレス LoRaWANデバイスゲートウェイを作成します。

```
aws iotwireless create-wireless-gateway \  
  --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \  
  --name "myFirstLoRaWANGateway" \  
  --description "Using my first LoRaWAN gateway"
```

出力:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/12345678-  
a1b2-3c45-67d8-e90fa1b2c34d",  
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateWirelessGateway](#)」の「」を参照してください。

delete-destination

次の例は、delete-destination を使用する方法を説明しています。

AWS CLI

IoT ワイヤレス送信先を削除するには

次のdelete-destination例では、IoTWirelessDestination作成した名前のワイヤレス送信先リソースを削除します。

```
aws iotwireless delete-destination \  
  --name "IoTWirelessDestination"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core への送信先の追加 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDestination](#)」の「」を参照してください。

delete-device-profile

次のコード例は、delete-device-profile を使用する方法を示しています。

AWS CLI

デバイスプロフィールを削除するには

次のdelete-device-profile例では、作成した指定された ID を持つデバイスプロフィールを削除します。

```
aws iotwireless delete-device-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core にプロフィールを追加する LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDeviceProfile](#)」の「」を参照してください。

delete-service-profile

次のコード例は、delete-service-profile を使用する方法を示しています。

AWS CLI

サービスプロフィールを削除するには

次のdelete-service-profile例では、作成した指定された ID のサービスプロフィールを削除します。

```
aws iotwireless delete-service-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core にプロファイルを追加する LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteServiceProfile](#)」の「」を参照してください。

delete-wireless-device

次の例は、delete-wireless-device を使用する方法を説明しています。

AWS CLI

ワイヤレスデバイスを削除するには

次のdelete-wireless-device例では、指定された ID を持つワイヤレスデバイスを削除します。

```
aws iotwireless delete-wireless-device \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteWirelessDevice](#)」の「」を参照してください。

delete-wireless-gateway-task-definition

次のコード例は、delete-wireless-gateway-task-definition を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイタスク定義を削除するには

次のdelete-wireless-gateway-task-definition例では、次の ID で作成したワイヤレスゲートウェイタスク定義を削除します。

```
aws iotwireless delete-wireless-gateway-task-definition \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DeleteWirelessGatewayTaskDefinition](#)」の「」を参照してください。

delete-wireless-gateway-task

次の例は、delete-wireless-gateway-task を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイタスクを削除するには

次のdelete-wireless-gateway-task例では、指定された ID を持つワイヤレスゲートウェイタスクを削除します。

```
aws iotwireless delete-wireless-gateway-task \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DeleteWirelessGatewayTask](#)」の「」を参照してください。

delete-wireless-gateway

次の例は、delete-wireless-gateway を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイを削除するには

次のdelete-wireless-gateway例では、指定された ID を持つワイヤレスゲートウェイを削除します。

```
aws iotwireless delete-wireless-gateway \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteWirelessGateway](#)」の「」を参照してください。

disassociate-aws-account-from-partner-account

次のコード例は、disassociate-aws-account-from-partner-account を使用する方法を示しています。

AWS CLI

パートナーアカウントの AWS アカウントとの関連付けを解除するには

次のdisassociate-aws-account-from-partner-account例では、パートナーアカウントと現在関連付けられている AWS アカウントとの関連付けを解除します。

```
aws iotwireless disassociate-aws-account-from-partner-account \  
  --partner-account-id "12345678901234" \  
  --partner-type "Sidewalk"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド」の「の IoT Core にゲートウェイとワイヤレスデバイスを追加する LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DisassociateAwsAccountFromPartnerAccount](#)」の「」を参照してください。

disassociate-wireless-device-from-thing

次のコード例は、disassociate-wireless-device-from-thing を使用する方法を示しています。

AWS CLI

ワイヤレスデバイスからモノの関連付けを解除するには

次のdisassociate-wireless-device-from-thing例では、現在関連付けられているモノからワイヤレスデバイスの関連付けを解除します。

```
aws iotwireless disassociate-wireless-device-from-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「[の IoT Core にゲートウェイとワイヤレスデバイスを追加する LoRaWANAWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDisassociateWirelessDeviceFromThing](#)」の「」を参照してください。

disassociate-wireless-gateway-from-certificate

次の例は、disassociate-wireless-gateway-from-certificate を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイから証明書の関連付けを解除するには

以下は、ワイヤレスゲートウェイと現在関連付けられている証明書の関連付けdisassociate-wireless-gateway-from-certificateを解除します。

```
aws iotwireless disassociate-wireless-gateway-from-certificate \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「[の IoT Core にゲートウェイとワイヤレスデバイスを追加する LoRaWANAWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDisassociateWirelessGatewayFromCertificate](#)」の「」を参照してください。

disassociate-wireless-gateway-from-thing

次の例は、disassociate-wireless-gateway-from-thing を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイからモノの関連付けを解除するには

次のdisassociate-wireless-gateway-from-thing例では、ワイヤレスゲートウェイを現在関連付けられているモノから関連付け解除します。

```
aws iotwireless disassociate-wireless-gateway-from-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド](#)の「[の IoT Core にゲートウェイとワイヤレスデバイスを追加する LoRaWANAWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDisassociateWirelessGatewayFromThing](#)」の「」を参照してください。

get-destination

次の例は、get-destination を使用する方法を説明しています。

AWS CLI

IoT ワイヤレス送信先に関する情報を取得するには

次のget-destination例では、IoTWirelessDestination作成した名前で送信先リソースに関する情報を取得します。

```
aws iotwireless get-destination \  
  --name "IoTWirelessDestination"
```

出力:

```
{
```

```
"Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
  "Name": "IoTWirelessDestination",
  "Expression": "IoTWirelessRule",
  "ExpressionType": "RuleName",
  "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT Core への送信先の追加 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDestination](#)」の「」を参照してください。

get-device-profile

次のコード例は、get-device-profile を使用する方法を示しています。

AWS CLI

デバイスプロフィールに関する情報を取得するには

次のget-device-profile例では、作成した指定された ID を持つデバイスプロフィールに関する情報を取得します。

```
aws iotwireless get-device-profile \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

出力:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
  "LoRaWAN": {
    "MacVersion": "1.0.3",
    "MaxDutyCycle": 10,
    "Supports32BitFCnt": false,
    "RegParamsRevision": "RP002-1.0.1",
    "SupportsJoin": true,
    "RfRegion": "US915",
```

```
"MaxEirp": 13,  
"SupportsClassB": false,  
"SupportsClassC": false  
}  
}
```

詳細については、[AWS「IoT デベロッパーガイド」の「の IoT Core にプロファイルを追加する LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [GetDeviceProfile](#)」の「」を参照してください。

get-partner-account

次のコード例は、get-partner-account を使用する方法を示しています。

AWS CLI

パートナーアカウント情報を取得するには

次のget-partner-account例では、次の ID を持つ Sidewalk アカウントに関する情報を取得します。

```
aws iotwireless get-partner-account \  
  --partner-account-id "12345678901234" \  
  --partner-type "Sidewalk"
```

出力:

```
{  
  "Sidewalk": {  
    "AmazonId": "12345678901234",  
    "Fingerprint":  
    "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"  
  },  
  "AccountLinked": false  
}
```

詳細については、[AWS IoT デベロッパーガイド」の「Amazon Sidewalk Integration for IoT Core AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPartnerAccount](#)」の「」を参照してください。

get-service-endpoint

次の例は、get-service-endpoint を使用する方法を説明しています。

AWS CLI

サービスエンドポイントを取得するには

次のget-service-endpoint例では、CUPSプロトコルのアカウント固有のエンドポイントを取得します。

```
aws iotwireless get-service-endpoint
```

出力:

```
{
  "ServiceType": "CUPS",
  "ServiceEndpoint": "https://A1RMKZ37ACAG0T.cups.lorawan.us-east-1.amazonaws.com:443",
  "ServerTrust": "-----BEGIN CERTIFICATE-----\n
MIIESTCCAzGgAwIBAgITBn+UV4WH6Kx33rJTMlu8mYtWDTANBgkqhkiG9w0BAQsF\n
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1UEChMGQW1hem9uMRkwFwYDVQQDEExBBbWF6\n
b24gUm9vdCBDQSAxMB4XDTE1MTAyMjAwMDAwMFoXDTE1MTAxOTAwMDAwMFowRjEL\n
MAKGA1UEBhMCMVVMxZANBgNVBAoTBkFtYXNpYXNpYXNpYXNpYXNpYXNpYXNpYXNp\n
IDFCMQ8wDQYDVQQDEwZBbWF6b24wgwgEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK\n
AoIBAQCDCThZn3c68asg3Wuw6MLAd5tES6BIOsMzoKcG5b1PVo+sDORrMd4f2AbnZ\n
cMzPa43j4wNxhplty6aUKk4T1qe9B0wKFjwK6zmxxLVYo7bHViXsP1J6q0MpFge5\n
b1DP+18x+B26A0piiQ0uPkfyDyeR4xQghfj66Yo19V+emU3nazfvpFA+R0z6WoVm\n
B5x+F2pV8xeKNR7u6azDdU5YVX1Tawp1mxRC1+WsAYmz6qP+z8ArDITC2FMVy2fw\n
0IjK0tEXc/VfmtTFch5+AfGYMGmqqvJ6LcXiAhqG5TI+Dr0RtM88k+8XUBCeQ8IG\n
KuANaL7TiItKZYxK1MMuTJtV9Ib1AgMBAAGjggE7MIIBNzASBgNVHRMBAf8ECDAG\n
AQH/AgEAMA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUWaRmBlKge5WSPK0UByew\n
dFv5PdAwHwYDVR0jBBgwFoAUhBjMhTTsvAyU1C4IWZzHshB0CggwewYIKwYBBQUH\n
AQEEbzBtMC8GCCsGAQUFBzABhiNodHRwOi8vb2NzcC5yb290Y2ExLmFtYXNpYXNp\n
dXN0LmNvbTA6BgggrBgEFBQcwAoYuaHR0cDovL2NydC5yb290Y2ExLmFtYXNpYXNp\n
dXN0LmNvbS9yb290Y2ExLmN1c3QuY29tL3Jvb3RjYTEuY3J5Sj5MBMGA1UdIAQMMAow\n
LnJvb3RjYTEuY29tL3Jvb3RjYTEuY3J5Sj5MBMGA1UdIAQMMAowLnJvb3RjYTEuY3J5Sj5M\n
CAYGZ4EMAQIBMA0GCSqGSIb3DQEBCwUAA4IBAQCfkr41u3nPo4FCH0TjY3NT0VI1\n
59Gt/a6ZiqyJEi+752+a1U5y6iAwYfmXss21JwJFqMp2PphKg5625kXg8kP2CN5t\n
-----"
```

```
6G7bMQcT8C8xDZNtYTd7WPD8UZiRKAJPBXa30/AbwuZe0GaFEQ8ugcYQgSn+IGBI\n8/LwhBNTZTUVEWuCUUBVV18YtbAiPq3yXqMB480z+ctBWuZSkbvkNodPLamkB2g1\nupRyzQ7qDn1X8nn8N8V7YJ6y68AtkHcNSRAnpTitxBKjtKPISLMVCx7i4hncxHZS\nyLyKQXhw2W2Xs0qLeC1etA+jTGDK4UfLeC0SF7FSi8o5LL21L8IzApar2pR/\n-----END CERTIFICATE-----\n"}
```

詳細については、[AWS「IoT デベロッパーガイド」の「IoT Core へのデバイスとゲートウェイの接続 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [GetServiceEndpoint](#)」の「」を参照してください。

get-service-profile

次の例は、get-service-profile を使用する方法を説明しています。

AWS CLI

サービスプロファイルに関する情報を取得するには

次のget-service-profile例では、作成した指定された ID を持つサービスプロファイルに関する情報を取得します。

```
aws iotwireless get-service-profile \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

出力:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:651419225604:ServiceProfile/538185bb-
d7e7-4b95-96a0-c51aa4a5b9a0",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
  "LoRaWAN": {
    "HrAllowed": false,
    "NwkGeoLoc": false,
    "DrMax": 15,
    "UlBucketSize": 4096,
    "PrAllowed": false,
    "ReportDevStatusBattery": false,
    "DrMin": 0,
  }
}
```

```
"DlRate": 60,  
"AddGwMetadata": false,  
"ReportDevStatusMargin": false,  
"MinGwDiversity": 1,  
"RaAllowed": false,  
"DlBucketSize": 4096,  
"DevStatusReqFreq": 24,  
"TargetPer": 5,  
"UlRate": 60  
}  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core にプロファイルを追加する LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceProfile](#)」の「」を参照してください。

get-wireless-device-statistics

次の例は、get-wireless-device-statistics を使用する方法を説明しています。

AWS CLI

ワイヤレスデバイスに関する運用情報を取得するには

次のget-wireless-device-statistics例では、ワイヤレスデバイスに関する操作情報を取得します。

```
aws iotwireless get-wireless-device-statistics \  
  --wireless-device-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

出力:

```
{  
  "WirelessDeviceId": "1ffd32c8-8130-4194-96df-622f072a315f"  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWirelessDeviceStatistics](#)」の「」を参照してください。

get-wireless-device

次のコード例は、get-wireless-device を使用する方法を示しています。

AWS CLI

ワイヤレスデバイスに関する情報を取得するには

次のget-wireless-device例では、AWS アカウントで使用可能なウィジェットを一覧表示します。

```
aws iotwireless get-wireless-device \  
  --identifier "1ffd32c8-8130-4194-96df-622f072a315f" \  
  --identifier-type WirelessDeviceID
```

出力:

```
{  
  "Name": "myLoRaWANDevice",  
  "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/44b87eb4-9bce-423d-b5fc-973f5ecc358b",  
  "DestinationName": "IoTWirelessDestination",  
  "Id": "1ffd32c8-8130-4194-96df-622f072a315f",  
  "ThingName": "44b87eb4-9bce-423d-b5fc-973f5ecc358b",  
  "Type": "LoRaWAN",  
  "LoRaWAN": {  
    "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",  
    "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",  
    "OtaaV1_1": {  
      "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",  
      "JoinEui": "b4c231a359bc2e3d",  
      "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"  
    },  
    "DevEui": "ac12efc654d23fc2"  
  },  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",  
  "Description": "My LoRaWAN wireless device"  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWirelessDevice](#)」の「」を参照してください。

get-wireless-gateway-certificate

次の例は、get-wireless-gateway-certificate を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイに関連付けられた証明書の ID を取得するには

次のget-wireless-gateway-certificate例では、指定された ID を持つワイヤレスゲートウェイに関連付けられた証明書 ID を取得します。

```
aws iotwireless get-wireless-gateway-certificate \  
  --id "6c44ab31-8b4d-407a-bed3-19b6c7cda551"
```

出力:

```
{  
  "IotCertificateId":  
  "8ea4aeae3db34c78cce75d9abd830356869ead6972997e0603e5fd032c804b6f"  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWirelessGatewayCertificate](#)」の「」を参照してください。

get-wireless-gateway-firmware-information

次の例は、get-wireless-gateway-firmware-information を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイに関するファームウェア情報を取得するには

次のget-wireless-gateway-firmware-information例では、ファームウェアバージョンとワイヤレスゲートウェイに関するその他の情報を取得します。

```
aws iotwireless get-wireless-gateway-firmware-information \  
--id "3039b406-5cc9-4307-925b-9948c63da25b"
```

出力:

```
{  
  "LoRaWAN" :{  
    "CurrentVersion" :{  
      "PackageVersion" : "1.0.0",  
      "Station" : "2.0.5",  
      "Model" : "linux"  
    }  
  }  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetWirelessGatewayFirmwareInformation](#)」の「」を参照してください。

get-wireless-gateway-statistics

次の例は、get-wireless-gateway-statistics を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイに関する運用情報を取得するには

次のget-wireless-gateway-statistics例では、ワイヤレスゲートウェイに関する操作情報を取得します。

```
aws iotwireless get-wireless-gateway-statistics \  
--wireless-gateway-id "3039b406-5cc9-4307-925b-9948c63da25b"
```

出力:

```
{
```

```
"WirelessGatewayId": "3039b406-5cc9-4307-925b-9948c63da25b"
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetWirelessGatewayStatistics](#)」の「」を参照してください。

get-wireless-gateway-task-definition

次のコード例は、get-wireless-gateway-task-definition を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイタスク定義に関する情報を取得するには

次のget-wireless-gateway-task-definition例では、指定された ID を持つワイヤレスタスク定義に関する情報を取得します。

```
aws iotwireless get-wireless-gateway-task-definition \
  --id "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
```

出力:

```
{
  "AutoCreateTasks": true,
  "Name": "TestAutoUpdate",
  "Update": {
    "UpdateDataSource" : "s3://cupsalphagafirmwarebin/station",
    "UpdateDataRole" : "arn:aws:iam::001234567890:role/SDK_Test_Role",
    "LoRaWAN" : {
      "CurrentVersion" : {
        "PackageVersion" : "1.0.0",
        "Station" : "2.0.5",
        "Model" : "linux"
      },
      "UpdateVersion" : {
        "PackageVersion" : "1.0.1",
        "Station" : "2.0.5",
        "Model" : "minihub"
      }
    }
  }
}
```

```
    }  
  }  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWirelessGatewayTaskDefinition](#)」の「」を参照してください。

get-wireless-gateway-task

次のコード例は、get-wireless-gateway-task を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイタスクに関する情報を取得するには

次のget-wireless-gateway-task例では、指定された ID のワイヤレスゲートウェイタスクに関する情報を取得します。

```
aws iotwireless get-wireless-gateway-task \  
  --id "11693a46-6866-47c3-a031-c9a616e7644b"
```

出力:

```
{  
  "WirelessGatewayId": "6c44ab31-8b4d-407a-bed3-19b6c7cda551",  
  "WirelessGatewayTaskDefinitionId": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",  
  "Status": "Success"  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWirelessGatewayTask](#)」の「」を参照してください。

get-wireless-gateway

次のコード例は、get-wireless-gateway を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイに関する情報を取得するには

次のget-wireless-gateway例では、ワイヤレスゲートウェイに関する情報を取得しますmyFirstLoRaWANGateway。

```
aws iotwireless get-wireless-gateway \  
  --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --identifier-type WirelessGatewayId
```

出力:

```
{  
  "Description": "My first LoRaWAN gateway",  
  "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/a1b2c3d4-5678-90ab-cdef-12ab345c67de",  
  "LoRaWAN": {  
    "RfRegion": "US915",  
    "GatewayEui": "a1b2c3d4567890ab"  
  },  
  "ThingName": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",  
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/6c44ab31-8b4d-407a-bed3-19b6c7cda551",  
  "Name": "myFirstLoRaWANGateway"  
}
```

詳細については、[AWS「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetWirelessGateway](#)」の「」を参照してください。

list-destinations

次のコード例は、list-destinations を使用する方法を示しています。

AWS CLI

ワイヤレス送信先を一覧表示するには

次のlist-destinations例では、AWS アカウントに登録されている利用可能な送信先を一覧表示します。

```
aws iotwireless list-destinations
```

出力:

```
{
  "DestinationList": [
    {
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
      "Name": "IoTWirelessDestination",
      "Expression": "IoTWirelessRule",
      "Description": "Destination for messages processed using
IoTWirelessRule",
      "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
    },
    {
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination2",
      "Name": "IoTWirelessDestination2",
      "Expression": "IoTWirelessRule2",
      "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
    }
  ]
}
```

詳細については、[AWS IoT デベロッパーガイド](#)の「[の IoT Core への送信先の追加](#)
[LoRaWAN AWS IoT](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDestinations](#)」の「」を参照してください。

list-device-profiles

次の例は、list-device-profiles を使用する方法を説明しています。

AWS CLI

デバイスプロフィールを一覧表示するには

次のlist-device-profiles例は、AWS アカウントに登録されている利用可能なデバイスプロフィールを一覧表示します。

```
aws iotwireless list-device-profiles
```

出力:

```
{
  "DeviceProfileList": [
    {
      "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d"
    },
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/a1b2c3d4-5678-90ab-cdef-12ab345c67de"
    }
  ]
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core にプロフィールを追加する LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeviceProfiles](#)」の「」を参照してください。

list-partner-accounts

次の例は、list-partner-accounts を使用する方法を説明しています。

AWS CLI

パートナーアカウントを一覧表示するには

次のlist-partner-accounts例では、アカウントに関連付けられている利用可能なパートナー AWS アカウントを一覧表示します。

```
aws iotwireless list-partner-accounts
```

出力:

```
{
  "Sidewalk": [
    {
      "AmazonId": "78965678771228",
      "Fingerprint":
"bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
    },
    {
      "AmazonId": "89656787651228",
      "Fingerprint":
"bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
    }
  ]
}
```

詳細については、[AWS IoT デベロッパーガイド](#)の「[Amazon Sidewalk Integration for IoT Core](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPartnerAccounts](#)」の「」を参照してください。

list-service-profiles

次のコード例は、list-service-profiles を使用する方法を示しています。

AWS CLI

サービスプロファイルを一覧表示するには

次のlist-service-profiles例では、AWS アカウントに登録されている利用可能なサービスプロファイルを一覧表示します。

```
aws iotwireless list-service-profiles
```

出力:

```
{
  "ServiceProfileList": [
    {
```

```
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:ServiceProfile/538185bb-d7e7-4b95-96a0-c51aa4a5b9a0"
  },
  {
    "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/
ea8bc823-5d13-472e-8d26-9550737d8100"
  }
]
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT Core にプロファイルを追加する LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListServiceProfiles](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースに割り当てられたタグを一覧表示するには

次のlist-tags-for-resource例では、ワイヤレス送信先リソースに割り当てられたタグを一覧表示します。

```
aws iotwireless list-tags-for-resource \
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination"
```

出力:

```
{
  "Tags": [
    {
      "Value": "MyValue",
      "Key": "MyTag"
    }
  ]
}
```



```
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「リソースの LoRaWAN IoT Core の説明」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-wireless-devices

次のコード例は、list-wireless-devices を使用する方法を示しています。

AWS CLI

使用可能なワイヤレスデバイスを一覧表示するには

次のlist-wireless-devices例では、AWS アカウントに登録されている使用可能なワイヤレスデバイスを一覧表示します。

```
aws iotwireless list-wireless-devices
```

出力:

```
{
  "WirelessDeviceList": [
    {
      "Name": "myLoRaWANDevice",
      "DestinationName": "IoTWirelessDestination",
      "Id": "1fffd32c8-8130-4194-96df-622f072a315f",
      "Type": "LoRaWAN",
      "LoRaWAN": {
        "DevEui": "ac12efc654d23fc2"
      },
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/1fffd32c8-8130-4194-96df-622f072a315f"
    }
  ]
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListWirelessDevices](#)」の「」を参照してください。

list-wireless-gateway-task-definitions

次のコード例は、list-wireless-gateway-task-definitions を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイタスク定義を一覧表示するには

次のlist-wireless-gateway-task-definitions例は、AWS アカウントに登録されている利用可能なワイヤレスゲートウェイタスク定義を一覧表示します。

```
aws iotwireless list-wireless-gateway-task-definitions
```

出力:

```
{
  "TaskDefinitions": [
    {
      "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",
      "LoRaWAN" :
      {
        "CurrentVersion" :{
          "PackageVersion" : "1.0.0",
          "Station" : "2.0.5",
          "Model" : "linux"
        },
        "UpdateVersion" :{
          "PackageVersion" : "1.0.1",
          "Station" : "2.0.5",
          "Model" : "minihub"
        }
      }
    }
  ]
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListWirelessGatewayTaskDefinitions](#)」の「」を参照してください。

list-wireless-gateways

次のコード例は、list-wireless-gateways を使用する方法を示しています。

AWS CLI

ワイヤレスゲートウェイを一覧表示するには

次のlist-wireless-gateways例では、AWS アカウントで使用可能なワイヤレスゲートウェイを一覧表示します。

```
aws iotwireless list-wireless-gateways
```

出力:

```
{
  "WirelessGatewayList": [
    {
      "Description": "My first LoRaWAN gateway",
      "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "dac632ebc01d23e4"
      },
      "Id": "3039b406-5cc9-4307-925b-9948c63da25b",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/3039b406-5cc9-4307-925b-9948c63da25b",
      "Name": "myFirstLoRaWANGateway"
    },
    {
      "Description": "My second LoRaWAN gateway",
      "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "cda123fffe92ecd2"
      },
      "Id": "3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Name": "mySecondLoRaWANGateway"
    }
  ]
}
```

```
]
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListWirelessGateways](#)」の「」を参照してください。

send-data-to-wireless-device

次の例は、send-data-to-wireless-device を使用する方法を説明しています。

AWS CLI

ワイヤレスデバイスにデータを送信するには

次のsend-data-to-wireless-device例では、復号されたアプリケーションデータフレームをワイヤレスデバイスに送信します。

```
aws iotwireless send-data-to-wireless-device \
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
  --transmit-mode "1" \
  --payload-data "SGVsbG8gVG8gRGV2c2lt" \
  --wireless-metadata LoRaWAN={FPort=1}
```

出力:

```
{
  MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendDataToWirelessDevice](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースのタグキーと値を指定するには

次のtag-resource例では、ワイヤレス送信先にキーMyTagと値IoWirelessDestinationでタグ付けしますMyValue。

```
aws iotwireless tag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:651419225604:Destination/  
IoWirelessDestination" \  
  --tags Key="MyTag",Value="MyValue"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「リソースの LoRaWAN IoT Core の説明」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

test-wireless-device

次のコード例は、test-wireless-device を使用する方法を示しています。

AWS CLI

ワイヤレスデバイスをテストするには

次のtest-wireless-device例ではHello、 のアップリンクデータを、指定された ID を持つデバイスに送信します。

```
aws iotwireless test-wireless-device \  
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49"
```

出力:

```
{  
  Result: "Test succeeded. one message is sent with payload: hello"  
}
```

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core へのデバイスとゲートウェイの接続 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TestWirelessDevice](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースから 1 つ以上のタグを削除するには

次のuntag-resource例では、ワイヤレス送信先 からタグMyTagとその値を削除しますIoWirelessDestination。

```
aws iotwireless untag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoWirelessDestination" \  
  --tag-keys "MyTag"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「リソースの LoRaWAN IoT Core の説明」](#)を参照してください。AWS IoT

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-destination

次の例は、update-destination を使用する方法を説明しています。

AWS CLI

送信先のプロパティを更新するには

次のupdate-destination例では、ワイヤレス送信先の説明プロパティを更新します。

```
aws iotwireless update-destination \  
  --name "IoWirelessDestination" \  
  --description "Destination for messages processed using IoWirelessRule"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「IoT デベロッパーガイド」の「の IoT Core への送信先の追加 LoRaWAN AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDestination](#)」の「」を参照してください。

update-partner-account

次のコード例は、update-partner-account を使用する方法を示しています。

AWS CLI

パートナーアカウントのプロパティを更新するには

以下に、指定された ID を持つ AppServerPrivateKey アカウントの update-partner-account を更新します。

```
aws iotwireless update-partner-account \  
  --partner-account-id "78965678771228" \  
  --partner-type "Sidewalk" \  
  --sidewalk  
  AppServerPrivateKey="f798ab4899346a88599180fee9e14fa1ada7b6df989425b7c6d2146dd6c815bb"
```

このコマンドでは何も出力されません。

詳細については、[AWS IoT デベロッパーガイド」の「Amazon Sidewalk Integration for IoT Core AWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePartnerAccount](#)」の「」を参照してください。

update-wireless-device

次の例は、update-wireless-device を使用する方法を説明しています。

AWS CLI

ワイヤレスデバイスのプロパティを更新するには

次の update-wireless-device 例では、AWS アカウントに登録されているワイヤレスデバイスのプロパティを更新します。

```
aws iotwireless update-wireless-device \  
  --id "1fffd32c8-8130-4194-96df-622f072a315f" \  
  --destination-name IoTWirelessDestination2 \  
  --description "Using my first LoRaWAN device"
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT デベロッパーガイド」の「IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateWirelessDevice](#)」の「」を参照してください。

update-wireless-gateway

次の例は、update-wireless-gateway を使用する方法を説明しています。

AWS CLI

ワイヤレスゲートウェイを更新するには

次のupdate-wireless-gateway例では、ワイヤレスゲートウェイの説明を更新します。

```
aws iotwireless update-wireless-gateway \  
  --id "3285bdc7-5a12-4991-84ed-dadca65e342e" \  
  --description "Using my LoRaWAN gateway"
```

このコマンドでは何も出力されません。

詳細については、[AWS「IoT デベロッパーガイド」の「IoT Core へのデバイスとゲートウェイの接続 LoRaWANAWS IoT」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateWirelessGateway](#)」の「」を参照してください。

を使用した Amazon IVS の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますIVS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-get-channel

次の例は、batch-get-channel を使用する方法を説明しています。

AWS CLI

複数のチャンネルに関するチャンネル設定情報を取得するには

次のbatch-get-channel例は、指定されたチャンネルに関する情報を示しています。

```
aws ivs batch-get-channel \
  --arns arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
  arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl
```

出力:

```
{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "authorized": false,
      "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
      "insecureIngest": false,
      "latencyMode": "LOW",
      "name": "channel-1",
      "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel-1.abcdEFGH.m3u8",
      "preset": "",
      "playbackRestrictionPolicyArn": ""
    }
  ]
}
```

```

    "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
    "srt": {
        "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
        "passphrase":
"AB1C2defGHijklMN03PqQRstUvwxyzABCDEFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "tags": {},
    "type": "STANDARD"
},
{
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
    "authorized": false,
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": true,
    "latencyMode": "LOW",
    "name": "channel-2",
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/
api/video/v1/us-west-2.123456789012.channel-2.abcdEFGH.m3u8",
    "preset": "",
    "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
    "recordingConfigurationArn": "",
    "srt": {
        "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
        "passphrase":
"BA1C2defGHijklMN03PqQRstUvwxyzABCDEFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "tags": {},
    "type": "STANDARD"
}
]
}

```

詳細については、IVS「低レイテンシーユーザーガイド」の「[チャンネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[BatchGetChannel](#)」の「」を参照してください。

batch-get-stream-key

次の例は、batch-get-stream-key を使用する方法を説明しています。

AWS CLI

複数のストリームキーに関する情報を取得するには

次のbatch-get-stream-key例では、指定されたストリームキーに関する情報を取得します。

```
aws ivs batch-get-stream-key \  
  --arns arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh \  
         arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop
```

出力:

```
{  
  "streamKeys": [  
    {  
      "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",  
      "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",  
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
      "tags": {}  
    },  
    {  
      "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKIJKLmnop",  
      "value": "sk_us-west-2_abcdABCDefgh_567890ghijkl",  
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
      "tags": {}  
    }  
  ]  
}
```

詳細については、[「低レイテンシーユーザーガイド」の「チャネルの作成」](#)を参照してください。IVS

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetStreamKey](#)」の「」を参照してください。

batch-start-viewer-session-revocation

次のコード例は、batch-start-viewer-session-revocation を使用する方法を示しています。

AWS CLI

複数のチャンネル ID ARNペアとビューワー ID ペアのビューワーセッションを取り消すには

次のbatch-start-viewer-session-revocation例では、複数のチャンネル ID ARNペアとビューワー ID ペアで同時にセッションの失効を実行します。リクエストは正常に完了できませんが、発信者に指定されたセッションを取り消すアクセス許可がない場合、エラーフィールドに値を返します。

```
aws ivs batch-start-viewer-session-revocation \
  --viewer-sessions '[{"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1","viewerId":"abcdefg1","viewerSessionVersionsLessThanOrEqualTo":1234567890},
\
  {"channelArn":"arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2","viewerId":"abcdefg2","viewerSessionVersionsLessThanOrEqualTo":1234567890}]'
```

出力:

```
{
  "errors": [
    {
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh1",
      "viewerId": "abcdefg1",
      "code": "403",
      "message": "not authorized",
    },
    {
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/
abcdABCDefgh2",
      "viewerId": "abcdefg2",
      "code": "403",
      "message": "not authorized",
    }
  ]
}
```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[プライベートチャンネルの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchStartViewerSessionRevocation](#)」の「」を参照してください。

create-channel

次のコード例は、create-channel を使用する方法を示しています。

AWS CLI

例 1: 録画のないチャンネルを作成するには

次のcreate-channel例では、ストリーミングを開始するための新しいチャンネルと関連付けられたストリームキーを作成します。

```
aws ivs create-channel \  
  --name "test-channel" \  
  --no-insecure-ingest
```

出力:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "authorized": false,  
    "name": "test-channel",  
    "latencyMode": "LOW",  
    "playbackRestrictionPolicyArn": "",  
    "recordingConfigurationArn": "",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "tags": {},  
    "type": "STANDARD"  
  },  
  "streamKey": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6",  
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",  
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "tags": {}  
  }  
}
```

```
}
}
```

詳細については、[「低レイテンシーユーザーガイド」の「チャンネルの作成」](#)を参照してください。 IVS

例 2: 録画が有効になっているチャンネルを作成するには、そのチャンネルで指定された RecordingConfiguration リソースを使用します。 ARN

次の create-channel 例では、ストリーミングを開始するための新しいチャンネルと関連付けられたストリームキーを作成し、チャンネルの録画を設定します。

```
aws ivs create-channel \
  --name test-channel-with-recording \
  --insecure-ingest \
  --recording-configuration-arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh"
```

出力:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "test-channel-with-recording",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "",
    "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": true,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
    "authorized": false,
    "tags": {},
    "type": "STANDARD"
  },
}
```

```

    "streamKey": {
      "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
      "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "tags": {}
    }
  }
}

```

詳細については、IVS「低レイテンシーユーザーガイド」の[Amazon S3への録音](#)を参照してください。

例 3: で指定された再生制限ポリシーを使用してチャンネルを作成するには ARN

次のcreate-channel例では、ストリーミングを開始するための新しいチャンネルと関連付けられたストリームキーを作成し、チャンネルの再生制限ポリシーを設定します。

```

aws ivs create-channel \
  --name test-channel-with-playback-restriction-policy \
  --insecure-ingest \
  --playback-restriction-policy-arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"

```

出力:

```

{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "test-channel-with-playback-restriction-policy",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
    "recordingConfigurationArn": "",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"AB1C2edfGHijklMN03PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": true,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
  }
}

```

```

    "authorized": false,
    "tags": {},
    "type": "STANDARD"
  },
  "streamKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "tags": {}
  }
}

```

詳細については、IVS低レイテンシーユーザーガイドの[「望ましくないコンテンツとビューワー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateChannel](#)」の「」を参照してください。

create-playback-restriction-policy

次のコード例は、create-playback-restriction-policy を使用する方法を示しています。

AWS CLI

再生制限ポリシーを作成するには

次のcreate-playback-restriction-policy例では、新しい再生リストラクションポリシーを作成します。

```

aws ivs create-playback-restriction-policy \
  --name "test-playback-restriction-policy" \
  --enable-strict-origin-enforcement \
  --tags "key1=value1, key2=value2" \
  --allowed-countries US MX \
  --allowed-origins https://www.website1.com https://www.website2.com

```

出力:

```

{
  "playbackRestrictionPolicy": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABCdef34ghIJ",
    "allowedCountries": [

```



```

        "US",
        "MX"
    ],
    "allowedOrigins": [
        "https://www.website1.com",
        "https://www.website2.com"
    ],
    "enableStrictOriginEnforcement": true,
    "name": "test-playback-restriction-policy",
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
}
}

```

詳細については、IVS「[低レイテンシーユーザーガイド](#)」の「[望ましくないコンテンツとビューワー](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスCreatePlaybackRestrictionPolicy](#)」の「」を参照してください。

create-recording-configuration

次のコード例は、create-recording-configuration を使用する方法を示しています。

AWS CLI

RecordingConfiguration リソースを作成するには

次のcreate-recording-configuration例では、Amazon S3 への記録を有効にする RecordingConfiguration リソースを作成します。

```

aws ivs create-recording-configuration \
  --name "test-recording-config" \
  --recording-reconnect-window-seconds 60 \
  --tags "key1=value1, key2=value2" \
  --rendition-configuration renditionSelection="CUSTOM",renditions="HD" \
  --thumbnail-configuration
recordingMode="INTERVAL",targetIntervalSeconds=1,storage="LATEST",resolution="LOWEST_RESOLUTION" \
  --destination-configuration s3={bucketName=demo-recording-bucket}

```

出力:

```
{
  "recordingConfiguration": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
    "name": "test-recording-config",
    "destinationConfiguration": {
      "s3": {
        "bucketName": "demo-recording-bucket"
      }
    },
    "state": "CREATING",
    "tags": {
      "key1": "value1",
      "key2": "value2"
    },
    "thumbnailConfiguration": {
      "recordingMode": "INTERVAL",
      "targetIntervalSeconds": 1,
      "resolution": "LOWEST_RESOLUTION",
      "storage": [
        "LATEST"
      ]
    },
    "recordingReconnectWindowSeconds": 60,
    "renditionConfiguration": {
      "renditionSelection": "CUSTOM",
      "renditions": [
        "HD"
      ]
    }
  }
}
```

詳細については、[Amazon S3 への録音](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRecordingConfiguration](#)」の「」を参照してください。

create-stream-key

次のコード例は、create-stream-key を使用する方法を示しています。

AWS CLI

ストリームキーを作成するには

次のcreate-stream-key例では、指定された ARN (Amazon リソースネーム) のストリームキーを作成します。

```
aws ivs create-stream-key \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

出力:

```
{  
  "streamKey": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",  
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",  
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "tags": {}  
  }  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の「[チャンネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateStreamKey](#)」の「」を参照してください。

delete-channel

次の例は、delete-channel を使用する方法を説明しています。

AWS CLI

チャンネルとそれに関連付けられたストリームキーを削除するには

次のdelete-channel例では、指定された ARN (Amazon リソースネーム) を持つチャンネルを削除します。

```
aws ivs delete-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

このコマンドでは何も出力されません。

詳細については、IVS「低レイテンシーユーザーガイド」の[「チャンネルの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteChannel](#)」の「」を参照してください。

delete-playback-key-pair

次のコード例は、delete-playback-key-pair を使用する方法を示しています。

AWS CLI

指定された再生キーペアを削除するには

次のdelete-playback-key-pair例では、指定されたキーペアのフィンガープリントを返しません。

```
aws ivs delete-playback-key-pair \  
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

このコマンドでは何も出力されません。

詳細については、「Amazon Interactive Video Service ユーザーガイド」の[「プライベートチャンネルの設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeletePlaybackKeyPair](#)」の「」を参照してください。

delete-playback-restriction-policy

次の例は、delete-playback-restriction-policy を使用する方法を説明しています。

AWS CLI

再生制限ポリシーを削除するには

次のdelete-playback-restriction-policy例では、指定されたポリシー ARN (Amazon リソースネーム) を使用して再生リスクシヨンポリシーを削除します。

```
aws ivs delete-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABCdef34ghIJ"
```

このコマンドでは何も出力されません。

詳細については、IVS「低レイテンシーユーザーガイド」の[「望ましくないコンテンツとビューワー」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeletePlaybackRestrictionPolicy](#)」の「」を参照してください。

delete-recording-configuration

次の例は、delete-recording-configuration を使用する方法を説明しています。

AWS CLI

で指定された RecordingConfiguration リソースを削除するには ARN

次のdelete-recording-configuration例では、指定された を持つ RecordingConfiguration リソースを削除しますARN。

```
aws ivs delete-recording-configuration \  
  --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

このコマンドでは何も出力されません。

詳細については、[Amazon S3 への録音](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteRecordingConfiguration](#)」の「」を参照してください。

delete-stream-key

次の例は、delete-stream-key を使用する方法を説明しています。

AWS CLI

ストリームキーを削除するには

次のdelete-stream-key例では、指定された ARN (Amazon リソースネーム) のストリームキーを削除するため、ストリーミングに使用することはできません。

```
aws ivs delete-stream-key \  
  --arn "arn:aws:ivs:us-west-2:123456789012:stream-key/ABcdef34ghIJ"
```

```
--arn arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6
```

このコマンドでは何も出力されません。

詳細については、IVS「低レイテンシーユーザーガイド」の「[チャンネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDeleteStreamKey](#)」の「」を参照してください。

get-channel

次の例は、get-channel を使用する方法を説明しています。

AWS CLI

チャンネルの設定情報を取得するには

次のget-channel例では、指定されたチャンネル ARN (Amazon リソースネーム) のチャンネル設定を取得します。

```
aws ivs get-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

出力:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "channel-1",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "",  
    "preset": "",  
    "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-  
configuration/ABCD12cdEFgh",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
  }  
}
```

```
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "tags": {}
  }
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[「チャンネルの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetChannel](#)」の「」を参照してください。

get-playback-key-pair

次のコード例は、get-playback-key-pair を使用する方法を示しています。

AWS CLI

指定された再生キーペアを取得するには

次のget-playback-key-pair例では、指定されたキーペアのフィンガープリントを返します。

```
aws ivs get-playback-key-pair \
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

出力:

```
{
  "keyPair": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
    "name": "my-playback-key",
    "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",
    "tags": {}
  }
}
```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の[「プライベートチャンネルの設定」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetPlaybackKeyPair](#)」の「」を参照してください。

get-playback-restriction-policy

次のコード例は、get-playback-restriction-policy を使用する方法を示しています。

AWS CLI

再生制限ポリシーの設定情報を取得するには

次のget-playback-restriction-policy例では、指定されたポリシー ARN (Amazon リソースネーム) を使用して再生のレストリシトポリシー設定を取得します。

```
aws ivs get-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABcdef34ghIJ"
```

出力:

```
{  
  "playbackRestrictionPolicy": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABcdef34ghIJ",  
    "allowedCountries": [  
      "US",  
      "MX"  
    ],  
    "allowedOrigins": [  
      "https://www.website1.com",  
      "https://www.website2.com"  
    ],  
    "enableStrictOriginEnforcement": true,  
    "name": "test-playback-restriction-policy",  
    "tags": {  
      "key1": "value1",  
      "key2": "value2"  
    }  
  }  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[「望ましくないコンテンツとビューワー」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetPlaybackRestrictionPolicy](#)」の「」を参照してください。

get-recording-configuration

次のコード例は、get-recording-configuration を使用する方法を示しています。

AWS CLI

RecordingConfiguration リソースに関する情報を取得するには

次のget-recording-configuration例では、指定された の RecordingConfiguration リソースに関する情報を取得しますARN。

```
aws ivs get-recording-configuration \  
--arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

出力:

```
{  
  "recordingConfiguration": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/  
ABcdef34ghIJ",  
    "destinationConfiguration": {  
      "s3": {  
        "bucketName": "demo-recording-bucket"  
      }  
    },  
    "name": "test-recording-config",  
    "recordingReconnectWindowSeconds": 60,  
    "state": "ACTIVE",  
    "tags": {  
      "key1" : "value1",  
      "key2" : "value2"  
    },  
    "thumbnailConfiguration": {  
      "recordingMode": "INTERVAL",  
      "targetIntervalSeconds": 1,  
      "resolution": "LOWEST_RESOLUTION",  
      "storage": [  
        "LATEST"  
      ]  
    },  
    "renditionConfiguration": {  
      "renditionSelection": "CUSTOM",  
      "renditions": [  

```

```
        "HD"  
      ]  
    }  
  }  
}
```

詳細については、[Amazon S3 への録音](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRecordingConfiguration](#)」の「」を参照してください。

get-stream-key

次のコード例は、get-stream-key を使用する方法を示しています。

AWS CLI

ストリームに関する情報を取得するには

次のget-stream-key例では、指定されたストリームキーに関する情報を取得します。

```
aws ivs get-stream-key \  
  --arn arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh --region=us-  
west-2
```

出力:

```
{  
  "streamKey": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh",  
    "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",  
    "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "tags": {}  
  }  
}
```

詳細については、IVS 「低レイテンシーユーザーガイド」の[「チャネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStreamKey](#)」の「」を参照してください。

get-stream-session

次の例は、get-stream-session を使用する方法を説明しています。

AWS CLI

指定されたストリームのメタデータを取得するには

次のget-stream-session例では、指定されたチャンネル ARN (Amazon リソースネーム) と指定されたストリームのメタデータ設定を取得します。が指定され streamId ていない場合、チャンネルの最新のストリームが選択されます。

```
aws ivs get-stream-session \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --stream-id "mystream"
```

出力:

```
{  
  "streamSession": {  
    "streamId": "mystream1",  
    "startTime": "2023-06-26T19:09:28+00:00",  
    "channel": {  
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
      "name": "mychannel",  
      "latencyMode": "LOW",  
      "type": "STANDARD",  
      "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCdef34ghIJ",  
      "srt": {  
        "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
        "passphrase":  
        "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
      },  
      "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
      "playbackUrl": "url-string",  
      "authorized": false,  
      "insecureIngest": false,  
      "preset": ""  
    },  
    "ingestConfiguration": {  
      "video": {  
        "avcProfile": "Baseline",
```

```
        "avcLevel": "4.2",
        "codec": "avc1.42C02A",
        "encoder": "Lavf58.45.100",
        "targetBitrate": 8789062,
        "targetFramerate": 60,
        "videoHeight": 1080,
        "videoWidth": 1920
    },
    "audio": {
        "codec": "mp4a.40.2",
        "targetBitrate": 46875,
        "sampleRate": 8000,
        "channels": 2
    }
},
"recordingConfiguration": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABcdef34ghIJ",
    "name": "test-recording-config",
    "destinationConfiguration": {
        "s3": {
            "bucketName": "demo-recording-bucket"
        }
    },
    "state": "ACTIVE",
    "tags": {
        "key1": "value1",
        "key2": "value2"
    },
    "thumbnailConfiguration": {
        "recordingMode": "INTERVAL",
        "targetIntervalSeconds": 1,
        "resolution": "LOWEST_RESOLUTION",
        "storage": [
            "LATEST"
        ]
    },
    "recordingReconnectWindowSeconds": 60,
    "renditionConfiguration": {
        "renditionSelection": "CUSTOM",
        "renditions": [
            "HD"
        ]
    }
}
```

```
    },
    "truncatedEvents": [
      {
        "name": "Recording Start",
        "type": "IVS Recording State Change",
        "eventTime": "2023-06-26T19:09:35+00:00"
      },
      {
        "name": "Stream Start",
        "type": "IVS Stream State Change",
        "eventTime": "2023-06-26T19:09:34+00:00"
      },
      {
        "name": "Session Created",
        "type": "IVS Stream State Change",
        "eventTime": "2023-06-26T19:09:28+00:00"
      }
    ]
  }
}
```

詳細については、[「低レイテンシーユーザーガイド」の「チャンネルの作成」](#)を参照してください。IVS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStreamSession](#)」の「」を参照してください。

get-stream

次の例は、get-stream を使用する方法を説明しています。

AWS CLI

ストリームに関する情報を取得するには

次のget-stream例では、指定されたチャンネルのストリームに関する情報を取得します。

```
aws ivs get-stream \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

出力:

```
{
```

```

    "stream": {
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
      "startTime": "2020-05-05T21:55:38Z",
      "state": "LIVE",
      "health": "HEALTHY",
      "streamId": "st-ABCDEFghij01234KLMN5678",
      "viewerCount": 1
    }
  }
}

```

詳細については、IVS「低レイテンシーユーザーガイド」の「[チャンネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetStream](#)」の「」を参照してください。

import-playback-key-pair

次の例は、import-playback-key-pair を使用する方法を説明しています。

AWS CLI

新しいキーペアのパブリック部分をインポートするには

次のimport-playback-key-pair例では、指定されたパブリックキー (PEM形式の文字列として指定) をインポートし、新しいキーペアの arn とフィンガープリントを返します。

```

aws ivs import-playback-key-pair \
  --name "my-playback-key" \
  --public-key-material "G1lbnQx0TA3BgNVBAMMFdoeSBhcmUgeW91IGR1..."

```

出力:

```

{
  "keyPair": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
    "name": "my-playback-key",
    "fingerprint": "0a:1b:2c:ab:cd:ef:34:56:70:b1:b2:71:01:2a:a3:72",
    "tags": {}
  }
}

```

```
}
```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[プライベートチャンネルの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportPlaybackKeyPair](#)」の「」を参照してください。

list-channels

次のコード例は、list-channels を使用する方法を示しています。

AWS CLI

例 1: すべてのチャンネルの概要情報を取得するには

次のlist-channels例では、AWS アカウントのすべてのチャンネルを一覧表示します。

```
aws ivs list-channels
```

出力:

```
{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "name": "channel-1",
      "latencyMode": "LOW",
      "authorized": false,
      "insecureIngest": false,
      "preset": "",
      "playbackRestrictionPolicyArn": "",
      "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",
      "tags": {},
      "type": "STANDARD"
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
      "name": "channel-2",
      "latencyMode": "LOW",
      "authorized": false,
```

```

        "preset": "",
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABCdef34ghIJ",
        "recordingConfigurationArn": "",
        "tags": {},
        "type": "STANDARD"
    }
]
}

```

詳細については、IVS「低レイテンシーユーザーガイド」の[「チャンネルの作成」](#)を参照してください。

例 2: 指定された でフィルタリングされたすべてのチャンネルの概要情報を取得するには
RecordingConfiguration ARN

次のlist-channels例では、指定された に関連付けられている AWS アカウントのすべてのチャンネルを一覧表示します RecordingConfiguration ARN。

```

aws ivs list-channels \
  --filter-by-recording-configuration-arn "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh"

```

出力:

```

{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "name": "channel-1",
      "latencyMode": "LOW",
      "authorized": false,
      "insecureIngest": false,
      "preset": "",
      "playbackRestrictionPolicyArn": "",
      "recordingConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh",
      "tags": {},
      "type": "STANDARD"
    }
  ]
}

```


詳細については、IVS「低レイテンシーユーザーガイド」の[Amazon S3への録音](#)を参照してください。

例 3: 指定された でフィルタリングされたすべてのチャンネルに関する概要情報を取得するには PlaybackRestrictionPolicy ARN

次のlist-channels例では、指定された に関連付けられている AWS アカウントのすべてのチャンネルを一覧表示します PlaybackRestrictionPolicy ARN。

```
aws ivs list-channels \
  --filter-by-playback-restriction-policy-arn "arn:aws:ivs:us-
  west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"
```

出力:

```
{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
      "name": "channel-2",
      "latencyMode": "LOW",
      "authorized": false,
      "preset": "",
      "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
      west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
      "recordingConfigurationArn": "",
      "tags": {},
      "type": "STANDARD"
    }
  ]
}
```

詳細については、IVS低レイテンシーユーザーガイドの[「望ましくないコンテンツとビューワー」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListChannels](#)」の「」を参照してください。

list-playback-key-pairs

次の例は、list-playback-key-pairs を使用する方法を説明しています。

AWS CLI

すべての再生キーペアに関する概要情報を取得するには

次のlist-playback-key-pairs例では、すべてのキーペアに関する情報を返します。

```
aws ivs list-playback-key-pairs
```

出力:

```
{
  "keyPairs": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh",
      "name": "test-key-0",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:playback-key/ijkl15678mnop",
      "name": "test-key-1",
      "tags": {}
    }
  ]
}
```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[プライベートチャネルの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPlaybackKeyPairs](#)」の「」を参照してください。

list-playback-restriction-policies

次のコード例は、list-playback-restriction-policies を使用する方法を示しています。

AWS CLI

すべての再生制限ポリシーの概要情報を取得するには

次のlist-playback-restriction-policies例では、AWS アカウントのすべての再生制限ポリシーを一覧表示します。

aws ivs list-playback-restriction-policies

出力:

```
{
  "playbackRestrictionPolicies": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABCdef34ghIJ",
      "allowedCountries": [
        "US",
        "MX"
      ],
      "allowedOrigins": [
        "https://www.website1.com",
        "https://www.website2.com"
      ],
      "enableStrictOriginEnforcement": true,
      "name": "test-playback-restriction-policy",
      "tags": {
        "key1": "value1",
        "key2": "value2"
      }
    }
  ]
}
```

詳細については、IVS低レイテンシーユーザーガイドの[「望ましくないコンテンツとビューワー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPlaybackRestrictionPolicies](#)」の「」を参照してください。

list-recording-configurations

次のコード例は、list-recording-configurations を使用する方法を示しています。

AWS CLI

このアカウントで作成されたすべての RecordingConfiguration リソースを一覧表示するには

次のlist-recording-configurations例では、アカウント内のすべてのRecordingConfiguration リソースに関する情報を取得します。

```
aws ivs list-recording-configurations
```

出力:

```
{
  "recordingConfigurations": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
ABCdef34ghIJ",
      "name": "test-recording-config-1",
      "destinationConfiguration": {
        "s3": {
          "bucketName": "demo-recording-bucket-1"
        }
      },
      "state": "ACTIVE",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/
CD12abcdGHIJ",
      "name": "test-recording-config-2",
      "destinationConfiguration": {
        "s3": {
          "bucketName": "demo-recording-bucket-2"
        }
      },
      "state": "ACTIVE",
      "tags": {}
    }
  ]
}
```

詳細については、[Amazon S3 への録音](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRecordingConfigurations](#)」の「」を参照してください。

list-stream-keys

次の例は、list-stream-keys を使用する方法を説明しています。

AWS CLI

ストリームキーのリストを取得するには

次のlist-stream-keys例では、指定された ARN (Amazon リソースネーム) のすべてのストリームキーを一覧表示します。

```
aws ivs list-stream-keys \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

出力:

```
{  
  "streamKeys": [  
    {  
      "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",  
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
      "tags": {}  
    }  
  ]  
}
```

For 詳細については、IVS「低レイテンシーユーザーガイド」の[「チャネルの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListStreamKeys](#)」の「」を参照してください。

list-stream-sessions

次のコード例は、list-stream-sessions を使用する方法を示しています。

AWS CLI

現在の AWS リージョンで指定されたチャネルの現在および以前のストリームの概要を取得するには

次のlist-stream-sessions例では、指定されたチャンネル ARN (Amazon リソースネーム) のストリームの概要情報をレポートします。

```
aws ivs list-stream-sessions \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --max-results 25 \  
  --next-token ""
```

出力:

```
{  
  "nextToken": "set-2",  
  "streamSessions": [  
    {  
      "startTime": 1641578182,  
      "endTime": 1641579982,  
      "hasErrorEvent": false,  
      "streamId": "mystream"  
    }  
    ...  
  ]  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[「チャンネルの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListStreamSessions](#)」の「」を参照してください。

list-streams

次の例は、list-streams を使用する方法を説明しています。

AWS CLI

ライブストリームとその状態のリストを取得するには

次のlist-streams例では、AWS アカウントのすべてのライブストリームを一覧表示します。

```
aws ivs list-streams
```

出力:

```
{
  "streams": [
    {
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "state": "LIVE",
      "health": "HEALTHY",
      "streamId": "st-ABCDEFghij01234KLMN5678",
      "viewerCount": 1
    }
  ]
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[「チャンネルの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListStreams](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

AWS リソースのすべてのタグを一覧表示するには (チャンネル、ストリームキーなど)

次のlist-tags-for-resource例では、指定されたリソース ARN (Amazon リソースネーム) のすべてのタグを一覧表示します。

```
aws ivs list-tags-for-resource \
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

出力:

```
{
  "tags":
  {
    "key1": "value1",
    "key2": "value2"
  }
}
```

```
}  
}
```

詳細については、「Amazon Interactive Video Service APIリファレンス」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-metadata

次のコード例は、put-metadata を使用する方法を示しています。

AWS CLI

指定されたチャンネルのアクティブなストリームにメタデータを挿入するには

次のput-metadata例では、指定されたメタデータを、指定されたチャンネルのストリームに挿入します。

```
aws ivs put-metadata \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --metadata '{"my": "metadata"}'
```

このコマンドでは何も出力されません。

詳細については、IVS「低レイテンシーユーザーガイド」の「[チャンネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutMetadata](#)」の「」を参照してください。

start-viewer-session-revocation

次のコード例は、start-viewer-session-revocation を使用する方法を示しています。

AWS CLI

特定の複数のチャンネル ID ARNペアとビューワー ID ペアのビューワーセッションを取り消すには

次のstart-viewer-session-revocation例では、指定したセッションバージョン番号まで、指定したチャンネルARNとビューワー ID に関連付けられたビューワーセッションを取り消すプロセスを開始します。バージョンが指定されていない場合、デフォルトは 0 です。

```
aws ivs batch-start-viewer-session-revocation \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --viewer-id abcdefg \  
  --viewer-session-versions-less-than-or-equal-to 1234567890
```

このコマンドでは何も出力されません。

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[プライベートチャンネルの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartViewerSessionRevocation](#)」の「」を参照してください。

stop-stream

次の例は、stop-stream を使用する方法を説明しています。

AWS CLI

指定されたストリームを停止するには

次のstop-stream例では、指定されたチャンネルでストリームを停止します。

```
aws ivs stop-stream \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

このコマンドでは何も出力されません。

詳細については、IVS 「低レイテンシーユーザーガイド」の「[チャンネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopStream](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

AWS リソースのタグを追加または更新するには (チャンネル、ストリームキーなど)

次のtag-resource例では、指定されたリソース ARN (Amazon リソースネーム) のタグを追加または更新します。

```
aws ivs tag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tags "tagkey1=tagvalue1, tagkey2=tagvalue2"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Interactive Video Service APIリファレンス」の「タグ付け<https://docs.aws.amazon.com/ivs/latest/APIReference/Welcome.html>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

AWS リソースのタグを削除するには (チャンネル、ストリームキーなど)

次のuntag-resource例では、指定されたリソース ARN (Amazon リソースネーム) の指定されたタグを削除します。

```
aws ivs untag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tag-keys "tagkey1, tagkey2"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Interactive Video Service APIリファレンス」の「タグ付け<https://docs.aws.amazon.com/ivs/latest/APIReference/Welcome.html>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-channel

次のコード例は、update-channel を使用する方法を示しています。

AWS CLI

例 1: チャネルの設定情報を更新するには

次のupdate-channel例では、指定されたチャネルのチャネル設定を更新ARNして、チャネル名を変更します。これは、このチャネルの進行中のストリームには影響しません。変更を有効にするには、ストリームを停止して再起動する必要があります。

```
aws ivs update-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --name "channel-1" \  
  --insecure-ingest
```

出力:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "channel-1",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "",  
    "recordingConfigurationArn": "",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijk1MN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": true,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "authorized": false,  
    "tags": {}  
  }  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[「チャネルの作成」](#)を参照してください。

例 2: チャンネルの設定を更新して記録を有効にするには

次のupdate-channel例では、指定したチャンネルのチャンネル設定を更新ARNして、録画を有効にします。これは、このチャンネルの進行中のストリームには影響しません。変更を有効にするには、ストリームを停止して再起動する必要があります。

```
aws ivs update-channel \  
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
  --no-insecure-ingest \  
  --recording-configuration-arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh"
```

出力:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "test-channel-with-recording",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "",  
    "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"BA1C2defGHijkLMNo3PqQRstUvwxyzABCDefghh4ijk1MN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "authorized": false,  
    "tags": {}  
  }  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[Amazon S3への録音](#)を参照してください。

例 3: チャンネルの設定を更新して記録を無効にするには

次のupdate-channel例では、指定されたチャンネルのチャンネル設定を更新ARNして、録画を無効にします。これは、このチャンネルの進行中のストリームには影響しません。変更を有効にするには、ストリームを停止して再起動する必要があります。

```
aws ivs update-channel \  
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
  --recording-configuration-arn ""
```

出力:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "test-channel-with-recording",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "",  
    "recordingConfigurationArn": "",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijk1MN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "authorized": false,  
    "tags": {}  
  }  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[Amazon S3への録音](#)を参照してください。

例 4: チャンネルの設定を更新して再生制限を有効にするには

次のupdate-channel例では、指定されたチャンネルのチャンネル設定を更新ARNして、再生制限ポリシーを適用します。これは、このチャンネルの進行中のストリームには影響しません。変更を有効にするには、ストリームを停止して再起動する必要があります。

```
aws ivs update-channel \  
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
  --no-insecure-ingest \  
  --playback-restriction-policy-arn "arn:aws:ivs:us-west-2:123456789012:playback-  
restriction-policy/ABcdef34ghIJ"
```

出力:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "test-channel-with-playback-restriction-policy",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "arn:aws:ivs:us-  
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",  
    "recordingConfigurationArn": "",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "authorized": false,  
    "tags": {}  
  }  
}
```

詳細については、IVS低レイテンシーユーザーガイドの[「望ましくないコンテンツとビューワー」](#)を参照してください。

例 5: チャンネルの設定を更新して再生制限を無効にするには

次のupdate-channel例では、指定されたチャンネルのチャンネル設定を更新ARNして、再生制限を無効にします。これは、このチャンネルの進行中のストリームには影響しません。変更を有効にするには、ストリームを停止して再起動する必要があります。

```
aws ivs update-channel \  
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
  --no-insecure-ingest \  
  --playback-restriction-policy-arn "arn:aws:ivs:us-west-2:123456789012:playback-  
restriction-policy/ABcdef34ghIJ"
```

```
--arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
--playback-restriction-policy-arn ""
```

出力:

```
{  
  "channel": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
    "name": "test-channel-with-playback-restriction-policy",  
    "latencyMode": "LOW",  
    "type": "STANDARD",  
    "playbackRestrictionPolicyArn": "",  
    "recordingConfigurationArn": "",  
    "srt": {  
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",  
      "passphrase":  
"AB1C2defGHijklMNop3PqQRstUvwxyzABCDeFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
    },  
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",  
    "insecureIngest": false,  
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/  
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",  
    "preset": "",  
    "authorized": false,  
    "tags": {}  
  }  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[「望ましくないコンテンツとビューワー」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateChannel](#)」の「」を参照してください。

update-playback-restriction-policy

次の例は、update-playback-restriction-policy を使用する方法を説明しています。

AWS CLI

再生制限ポリシーを更新するには

次のupdate-playback-restriction-policy例では、指定されたポリシーで再生制限ポリシーを更新ARNして、厳密なオリジン適用を無効にします。これは、関連付けられたチャンネルの進行中のストリームには影響しません。変更を有効にするには、ストリームを停止して再起動する必要があります。

```
aws ivs update-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
ABcdef34ghIJ" \  
  --no-enable-strict-origin-enforcement
```

出力:

```
{  
  "playbackRestrictionPolicy": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
ABcdef34ghIJ",  
    "allowedCountries": [  
      "US",  
      "MX"  
    ],  
    "allowedOrigins": [  
      "https://www.website1.com",  
      "https://www.website2.com"  
    ],  
    "enableStrictOriginEnforcement": false,  
    "name": "test-playback-restriction-policy",  
    "tags": {  
      "key1": "value1",  
      "key2": "value2"  
    }  
  }  
}
```

詳細については、IVS「低レイテンシーユーザーガイド」の[「望ましくないコンテンツとビューワー」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdatePlaybackRestrictionPolicy](#)」の「」を参照してください。

を使用した Amazon IVS Chat の例 AWS CLI

次のコード例は、Amazon IVS Chat AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-chat-token

次のコード例は、create-chat-token を使用する方法を示しています。

AWS CLI

チャットトークンを作成するには

次のcreate-chat-token例では、部屋への個々の WebSocket 接続を確立するために使用される暗号化されたチャットトークンを作成します。トークンは 1 分間有効であり、トークンで確立された接続 (セッション) は指定された期間有効です。

```
aws ivschat create-chat-token \
  --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6", \
  --userId "11231234" \
  --capabilities "SEND_MESSAGE", \
  --sessionDurationInMinutes 30
```

出力:

```
{
  "token": "ACEGmnoq#1rstu2...BDFH3vxwy!4hlm!#5",
  "sessionExpirationTime": "2022-03-16T04:44:09+00:00"
  "state": "CREATING",
```

```
"tokenExpirationTime": "2022-03-16T03:45:09+00:00"
}
```

詳細については、「Amazon Interactive Video Service [ユーザーガイド](#)」の「[ステップ 3: チャットクライアントの認証と承認](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス CreateChatToken](#)」の「」を参照してください。

create-logging-configuration

次の例は、create-logging-configuration を使用する方法を説明しています。

AWS CLI

チャット LoggingConfiguration リソースを作成するには

次のcreate-logging-configuration例では、クライアントが送信されたメッセージを保存および記録できるようにする LoggingConfiguration リソースを作成します。

```
aws ivschat create-logging-configuration \
  --destination-configuration s3={bucketName=demo-logging-bucket} \
  --name "test-logging-config" \
  --tags "key1=value1, key2=value2"
```

出力:

```
{
  "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/ABcdef34ghIJ",
  "createTime": "2022-09-14T17:48:00.653000+00:00",
  "destinationConfiguration": {
    "s3": {
      "bucketName": "demo-logging-bucket"
    }
  },
  "id": "ABcdef34ghIJ",
  "name": "test-logging-config",
  "state": "ACTIVE",
  "tags": { "key1" : "value1", "key2" : "value2" },
  "updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

詳細については、「[Amazon Interactive Video Service ユーザーガイド](#)」の「[Amazon IVS Chat の開始方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoggingConfiguration](#)」の「」を参照してください。

create-room

次のコード例は、create-room を使用する方法を示しています。

AWS CLI

ルームを作成するには

次のcreate-room例では、新しいルームを作成します。

```
aws ivschat create-room \  
  --name "test-room-1" \  
  --logging-configuration-identifiers "arn:aws:ivschat:us-  
west-2:123456789012:logging-configuration/ABCdef34ghIJ" \  
  --maximum-message-length 256 \  
  --maximum-message-rate-per-second 5
```

出力:

```
{  
  "arn": "arn:aws:ivschat:us-west-2:123456789012:room/g1H2I3j4k5L6",  
  "id": "g1H2I3j4k5L6",  
  "createTime": "2022-03-16T04:44:09+00:00",  
  "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-  
west-2:123456789012:logging-configuration/ABCdef34ghIJ"],  
  "maximumMessageLength": 256,  
  "maximumMessageRatePerSecond": 5,  
  "name": "test-room-1",  
  "tags": {}  
  "updateTime": "2022-03-16T07:22:09+00:00"  
}
```

詳細については、「[Amazon Interactive Video Service ユーザーガイド](#)」の「[ステップ 2: チャットルームを作成する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRoom](#)」の「」を参照してください。

delete-logging-configuration

次の例は、delete-logging-configuration を使用する方法を説明しています。

AWS CLI

チャット LoggingConfiguration リソースを削除するには

次のdelete-logging-configuration例では、指定された の LoggingConfiguration リソースを削除しますARN。

```
aws ivschat delete-logging-configuration \  
  --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
  ABcdef34ghIJ"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLoggingConfiguration](#)」の「」を参照してください。

delete-message

次の例は、delete-message を使用する方法を説明しています。

AWS CLI

指定されたルームからメッセージを削除するには

次のdelete-message例では、を指定されたルームに送信します。このルームは、クライアントに指定されたメッセージを削除するように指示します。つまり、そのメッセージをビューからレンダリング解除し、クライアントのチャット履歴から削除します。

```
aws ivschat delete-message \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --id "ABC123def456" \  
  --reason "Message contains profanity"
```

出力:

```
{
  "id": "12345689012"
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMessage](#)」の「」を参照してください。

delete-room

次のコード例は、delete-room を使用する方法を示しています。

AWS CLI

ルームを削除するには

次のdelete-room例では、指定されたルームを削除します。接続されたクライアントは切断されます。成功すると、空のレスポンス本文を持つ HTTP 204 が返されます。

```
aws ivschat delete-room \
  --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRoom](#)」の「」を参照してください。

disconnect-user

次の例は、disconnect-user を使用する方法を説明しています。

AWS CLI

ルームからユーザーを切断するには

次のdisconnect-user例では、指定されたユーザーのすべての接続を指定されたルームから切断します。成功すると、空のレスポンス本文を持つ HTTP 200 が返されます。

```
aws ivschat disconnect-user \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --userId "ABC123def456" \  
  --reason "Violated terms of service"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」](#)の「[Amazon IVS Chat の開始方法](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDisconnectUser](#)」の「」を参照してください。

get-logging-configuration

次のコード例は、get-logging-configuration を使用する方法を示しています。

AWS CLI

LoggingConfiguration リソースに関する情報を取得するには

次のget-logging-configuration例では、指定された の LoggingConfiguration リソースに関する情報を取得しますARN。

```
aws ivschat get-logging-configuration \  
  --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
  ABcdef34ghIJ"
```

出力:

```
{  
  "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
  ABcdef34ghIJ",  
  "createTime": "2022-09-14T17:48:00.653000+00:00",  
  "destinationConfiguration": {  
    "s3": {  
      "bucketName": "demo-logging-bucket"  
    }  
  },  
  "id": "ABcdef34ghIJ",  
  "name": "test-logging-config",
```

```
"state": "ACTIVE",
"tags": { "key1" : "value1", "key2" : "value2" },
"updateTime": "2022-09-14T17:48:01.104000+00:00"
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLoggingConfiguration](#)」の「」を参照してください。

get-room

次のコード例は、get-room を使用する方法を示しています。

AWS CLI

指定されたルームを取得するには

次のget-room例では、指定されたルームに関する情報を取得します。

```
aws ivschat get-room \
  --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

出力:

```
{
  "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
  "createTime": "2022-03-16T04:44:09+00:00",
  "id": "g1H2I3j4k5L6",
  "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABCdef34ghIJ"],
  "maximumMessageLength": 256,
  "maximumMessageRatePerSecond": 5,
  "name": "test-room-1",
  "tags": {},
  "updateTime": "2022-03-16T07:22:09+00:00"
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRoom](#)」の「」を参照してください。

list-logging-configurations

次のコード例は、list-logging-configurations を使用する方法を示しています。

AWS CLI

APIリクエストが処理される AWS リージョン内のユーザーのすべてのログ記録設定に関する概要情報を取得するには

次のlist-logging-configurations例では、APIリクエストが処理される AWS リージョン内のユーザーのすべての LoggingConfiguration リソースに関する情報を示しています。

```
aws ivschat list-logging-configurations \  
  --max-results 2 \  
  --next-token ""
```

出力:

```
{  
  "nextToken": "set-2",  
  "loggingConfigurations": [  
    {  
      "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
ABcdef34ghIJ",  
      "createTime": "2022-09-14T17:48:00.653000+00:00",  
      "destinationConfiguration": {  
        "s3": {  
          "bucketName": "demo-logging-bucket"  
        }  
      },  
      "id": "ABcdef34ghIJ",  
      "name": "test-logging-config",  
      "state": "ACTIVE",  
      "tags": { "key1" : "value1", "key2" : "value2" },  
      "updateTime": "2022-09-14T17:48:01.104000+00:00"  
    }  
    ...  
  ]  
}
```



```
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」](#)の[「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスListLoggingConfigurations」](#)の「」を参照してください。

list-rooms

次の例は、list-rooms を使用する方法を説明しています。

AWS CLI

現在のリージョンのすべてのルームの概要情報を取得するには

次のlist-rooms例では、リクエストが処理される AWS リージョン内のすべてのルームの概要情報を取得します。結果は の降順でソートされますupdateTime。

```
aws ivschat list-rooms \
  --logging-configuration-identifier "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
  --max-results 10 \
  --next-token ""
```

出力:

```
{
  "nextToken": "page3",
  "rooms": [
    {
      "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
      "createTime": "2022-03-16T04:44:09+00:00",
      "id": "g1H2I3j4k5L6",
      "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],
      "name": "test-room-1",
      "tags": {},
      "updateTime": "2022-03-16T07:22:09+00:00"
    }
  ]
}
```

```
}
```

詳細については、「[Amazon Interactive Video Service ユーザーガイド](#)」の「[Amazon IVS Chat の開始方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRooms](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

AWS リソースのすべてのタグを一覧表示するには (例: Room)

次のlist-tags-for-resource例では、指定されたリソース ARN (Amazon リソースネーム) のすべてのタグを一覧表示します。

```
aws ivschat list-tags-for-resource \  
--resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6
```

出力:

```
{  
  "tags":  
  {  
    "key1": "value1",  
    "key2": "value2"  
  }  
}
```

詳細については、「[Amazon Interactive Video Service APIリファレンス](#)」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

send-event

次のコード例は、send-event を使用する方法を示しています。

AWS CLI

イベントをルームに送信するには

次のsend-event例では、指定されたイベントを指定されたルームに送信します。

```
aws ivschat send-event \  
  --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --eventName "SystemMessage" \  
  --attributes \  
    "msgType"="user-notification", \  
    "msgText"="This chat room will close in 15 minutes."
```

出力:

```
{  
  "id": "12345689012"  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendEvent](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

AWS リソースのタグを追加または更新するには (例: Room)

次のtag-resource例では、指定されたリソース ARN (Amazon リソース名前) のタグを追加または更新します。成功すると、空のレスポンス本文を持つ HTTP 200 が返されます。

```
aws ivschat tag-resource \  
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
  --tags "tagkey1=tagkeyvalue1, tagkey2=tagkeyvalue2"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Interactive Video Service APIリファレンス」の「[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

AWS リソースのタグを削除するには (例: Room)

次のuntag-resource例では、指定されたリソース ARN (Amazon リソースネーム) の指定されたタグを削除します。成功すると、空のレスポンス本文を持つ HTTP 200 が返されます。

```
aws ivschat untag-resource \  
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
  --tag-keys "tagkey1, tagkey2"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Interactive Video Service APIリファレンス」の「タグ付け<https://docs.aws.amazon.com/ivs/latest/APIReference/Welcome.html>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-logging-configuration

次の例は、update-logging-configuration を使用する方法を説明しています。

AWS CLI

ルームのログ記録設定を更新するには

次のupdate-logging-configuration例では、指定されたデータを使用して LoggingConfiguration リソースを更新します。

```
aws ivschat update-logging-configuration \  
  --logging-configuration ...
```

```
--destination-configuration s3={bucketName=demo-logging-bucket} \  
--identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
ABcdef34ghIJ" \  
--name "test-logging-config"
```

出力:

```
{  
  "arn": "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
ABcdef34ghIJ",  
  "createTime": "2022-09-14T17:48:00.653000+00:00",  
  "destinationConfiguration": {  
    "s3": {  
      "bucketName": "demo-logging-bucket"  
    }  
  },  
  "id": "ABcdef34ghIJ",  
  "name": "test-logging-config",  
  "state": "ACTIVE",  
  "tags": { "key1" : "value1", "key2" : "value2" },  
  "updateTime": "2022-09-14T17:48:01.104000+00:00"  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon IVS Chat の開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLoggingConfiguration](#)」の「」を参照してください。

update-room

次の例は、update-room を使用する方法を説明しています。

AWS CLI

ルームの設定を更新するには

次のupdate-room例では、指定されたデータを使用して、指定されたルームの設定を更新します。

```
aws ivschat update-room \  
--identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
--name "test-room"
```

```
--logging-configuration-identifiers "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/ABcdef34ghIJ" \  
--name "chat-room-a" \  
--maximum-message-length 256 \  
--maximum-message-rate-per-second 5
```

出力:

```
{  
  "arn": "arn:aws:ivschat:us-west-2:123456789012:room/g1H2I3j4k5L6",  
  "createTime": "2022-03-16T04:44:09+00:00",  
  "id": "g1H2I3j4k5L6",  
  "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-west-2:123456789012:logging-configuration/ABcdef34ghIJ"],  
  "maximumMessageLength": 256,  
  "maximumMessageRatePerSecond": 5,  
  "name": "chat-room-a",  
  "tags": {},  
  "updateTime": "2022-03-16T07:22:09+00:00"  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」](#)の「[Amazon IVS Chat の開始方法](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateRoom](#)」の「」を参照してください。

を使用した Amazon IVS リアルタイムストリーミングの例 AWS CLI

次のコード例は、Amazon IVS Real-Time Streaming AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-encoder-configuration

次の例は、create-encoder-configuration を使用する方法を説明しています。

AWS CLI

コンポジションエンコーダー設定を作成するには

次のcreate-encoder-configuration例では、指定されたプロパティを持つコンポジションエンコーダー設定を作成します。

```
aws ivs-realtime create-encoder-configuration \  
  --name test-ec --video bitrate=3500000,framerate=30.0,height=1080,width=1920
```

出力:

```
{  
  "encoderConfiguration": {  
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/  
ABabCDcdEFef",  
    "name": "test-ec",  
    "tags": {},  
    "video": {  
      "bitrate": 3500000,  
      "framerate": 30,  
      "height": 1080,  
      "width": 1920  
    }  
  }  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEncoderConfiguration](#)」の「」を参照してください。

create-participant-token

次のコード例は、create-participant-token を使用する方法を示しています。

AWS CLI

ステージ参加者トークンを作成するには

次のcreate-participant-token例では、指定されたステージの参加者トークンを作成します。

```
aws ivs-realtime create-participant-token \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --user-id bob
```

出力:

```
{  
  "participantToken": {  
    "expirationTime": "2023-03-07T09:47:43+00:00",  
    "participantId": "ABCDEFghij01234KLMN6789",  
    "token": "abcd1234defg5678"  
  }  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateParticipantToken](#)」の「」を参照してください。

create-stage

次のコード例は、create-stage を使用する方法を示しています。

AWS CLI

例 1: ステージを作成するには

次のcreate-stage例では、指定されたユーザーのステージとステージの参加者トークンを作成します。

```
aws ivs-realtime create-stage \  
  --name stage1 \  
  --participant-token-configurations userId=alice
```

出力:


```
{
  "participantTokens": [
    {
      "participantId": "ABCDEFghij01234KLMN5678",
      "token": "a1b2c3d4567890ab",
      "userId": "alice"
    }
  ],
  "stage": {
    "activeSessionId": "st-a1b2c3d4e5f6g",
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
    "endpoints": {
      "events": "wss://global.events.live-video.net",
      "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
    },
    "name": "stage1",
    "tags": {}
  }
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

例 2: ステージを作成し、個々の参加者の録音を設定するには

次のcreate-stage例では、ステージを作成し、個々の参加者の録音を設定します。

```
aws ivs-realtime create-stage \
  --name stage1 \
  --auto-participant-recording-configuration '{"mediaTypes":  
["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-  
west-2:123456789012:storage-configuration/abcdABCDefgh}"'
```

出力:

```
{
  "stage": {
    "activeSessionId": "st-a1b2c3d4e5f6g",
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
    "autoParticipantRecordingConfiguration": {
      "mediaTypes": [
        "AUDIO_VIDEO"
      ]
    }
  }
}
```

```
    ],
    "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-configuration/abcdABCDefgh",
  },
  "endpoints": {
    "events": "wss://global.events.live-video.net",
    "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
  },
  "name": "stage1",
  "tags": {}
}
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStage](#)」の「」を参照してください。

create-storage-configuration

次のコード例は、create-storage-configuration を使用する方法を示しています。

AWS CLI

コンポジションストレージ設定を作成するには

次のcreate-storage-configuration例では、指定されたプロパティを持つコンポジションストレージ設定を作成します。

```
aws ivs-realtime create-storage-configuration \
  --name "test-sc" --s3 "bucketName=test-bucket-name"
```

出力:

```
{
  "storageConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/ABabCDcdEFef",
    "name": "test-sc",
    "s3": {
      "bucketName": "test-bucket-name"
    }
  }
}
```

```
    },  
    "tags": {}  
  }  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStorageConfiguration](#)」の「」を参照してください。

delete-encoder-configuration

次のコード例は、delete-encoder-configuration を使用する方法を示しています。

AWS CLI

コンポジションエンコーダー設定を削除するには

以下は、指定された ARN (Amazon リソースネーム) で指定されたコンポジションエンコーダー設定delete-encoder-configurationを削除します。

```
aws ivs-realtime delete-encoder-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/  
  ABabCDcdEFef"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEncoderConfiguration](#)」の「」を参照してください。

delete-public-key

次のコード例は、delete-public-key を使用する方法を示しています。

AWS CLI

パブリックキーを削除するには

以下では、指定されたパブリックキーdelete-public-keyを削除します。

```
aws ivs-realtime delete-public-key \  
  --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Real-Time Streaming ユーザーガイド」の「参加者トークンの配布」](#)を参照してください。 IVS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePublicKey](#)」の「」を参照してください。

delete-stage

次の例は、delete-stage を使用する方法を説明しています。

AWS CLI

ステージを削除するには

次のdelete-stage例では、指定されたステージを削除します。

```
aws ivs-realtime delete-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStage](#)」の「」を参照してください。

delete-storage-configuration

次の例は、delete-storage-configuration を使用する方法を説明しています。

AWS CLI

コンポジションストレージ設定を削除するには

以下は、指定された ARN (Amazon リソースネーム) で指定されたコンポジションストレージ設定delete-storage-configurationを削除します。

```
aws ivs-realtime delete-storage-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/  
  ABabCDcdEFef"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStorageConfiguration](#)」の「」を参照してください。

disconnect-participant

次の例は、disconnect-participant を使用する方法を説明しています。

AWS CLI

ステージ参加者を切断するには

次のdisconnect-participant例では、指定されたステージから指定された参加者を切断します。

```
aws ivs-realtime disconnect-participant \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --participant-id ABCDEfghij01234KLMN5678
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisconnectParticipant](#)」の「」を参照してください。

get-composition

次の例は、get-composition を使用する方法を説明しています。

AWS CLI

例 1: デフォルトのレイアウト設定でコンポジションを取得するには

次のget-composition例では、指定された ARN (Amazon リソースネーム) の構成を取得します。

```
aws ivs-realtime get-composition \  
--arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

出力:

```
{  
  "composition": {  
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",  
    "destinations": [  
      {  
        "configuration": {  
          "channel": {  
            "channelArn": "arn:aws:ivs:ap-northeast-1:123456789012:channel/abcABCdefDEg",  
            "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"  
          },  
          "name": ""  
        },  
        "id": "AabBCcdDEefF",  
        "startTime": "2023-10-16T23:26:00+00:00",  
        "state": "ACTIVE"  
      },  
      {  
        "configuration": {  
          "name": "",  
          "s3": {  
            "encoderConfigurationArns": [  
              "arn:aws:ivs:arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"  
            ],  
            "recordingConfiguration": {  
              "format": "HLS"  
            },  
            "storageConfigurationArn": "arn:arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/FefABabCDcdE"  
          }  
        },  
        "detail": {  
          "s3": {
```

```

        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
        }
    },
    "id": "GHFabcgefABC",
    "startTime": "2023-10-16T23:26:00+00:00",
    "state": "STARTING"
}
],
"layout": {
    "grid": {
        "featuredParticipantAttribute": ""
        "gridGap": 2,
        "omitStoppedVideo": false,
        "videoAspectRatio": "VIDEO",
        "videoFillMode": ""
    }
},
"stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
"startTime": "2023-10-16T23:24:00+00:00",
"state": "ACTIVE",
"tags": {}
}
}

```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[複合録画 \(リアルタイムストリーミング\)](#)」を参照してください。

例 2: PiP レイアウトでコンポジションを取得するには

次のget-composition例では、PiP レイアウトを使用する、指定された ARN (Amazon リソースネーム) の構成を取得します。

```

aws ivs-realtime get-composition \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs"

```

出力:

```

{
  "composition": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
    "destinations": [
      {
        "configuration": {

```

```

        "channel": {
            "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
            "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
        },
        "name": ""
    },
    "id": "AabBCcdDEefF",
    "startTime": "2023-10-16T23:26:00+00:00",
    "state": "ACTIVE"
},
{
    "configuration": {
        "name": "",
        "s3": {
            "encoderConfigurationArns": [
                "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
            ],
            "recordingConfiguration": {
                "format": "HLS"
            },
            "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
        }
    },
    "detail": {
        "s3": {
            "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
        }
    },
    "id": "GHFabcgefABC",
    "startTime": "2023-10-16T23:26:00+00:00",
    "state": "STARTING"
}
],
"layout": {
    "pip": {
        "featuredParticipantAttribute": "abcdefg",
        "gridGap": 0,
        "omitStoppedVideo": false,
        "pipBehavior": "STATIC",

```



```
        "pipOffset": 0,
        "pipParticipantAttribute": "",
        "pipPosition": "BOTTOM_RIGHT",
        "videoFillMode": "COVER"
    }
},
"stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
"startTime": "2023-10-16T23:24:00+00:00",
"state": "ACTIVE",
"tags": {}
}
```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[複合録画 \(リアルタイムストリーミング\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetComposition](#)」の「」を参照してください。

get-encoder-configuration

次の例は、get-encoder-configuration を使用する方法を説明しています。

AWS CLI

コンポジションエンコーダー設定を取得するには

次のget-encoder-configuration例では、指定された ARN (Amazon リソースネーム) で指定されたコンポジションエンコーダー設定を取得します。

```
aws ivs-realtime get-encoder-configuration \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh"
```

出力:

```
{
  "encoderConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh",
    "name": "test-ec",
```

```
    "tags": {},
    "video": {
      "bitrate": 3500000,
      "framerate": 30,
      "height": 1080,
      "width": 1920
    }
  }
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEncoderConfiguration](#)」の「」を参照してください。

get-participant

次の例は、get-participant を使用する方法を説明しています。

AWS CLI

ステージ参加者を取得するには

次のget-participant例では、指定されたステージ ARN (Amazon リソースネーム) で、指定された参加者 ID とセッション ID のステージ参加者を取得します。

```
aws ivs-realtime get-participant \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g \
  --participant-id abCDEf12GHIj
```

出力:

```
{
  "participant": {
    "browserName", "Google Chrome",
    "browserVersion", "116",
    "firstJoinTime": "2023-04-26T20:30:34+00:00",
    "ispName", "Comcast",
    "osName", "Microsoft Windows 10 Pro",
```

```

    "osVersion", "10.0.19044"
    "participantId": "abCDEf12GHIj",
    "published": true,
    "recordingS3BucketName": "bucket-name",
    "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
    "recordingState": "ACTIVE",
    "sdkVersion", "",
    "state": "CONNECTED",
    "userId": "",
  }
}

```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParticipant](#)」の「」を参照してください。

get-public-key

次のコード例は、get-public-key を使用する方法を示しています。

AWS CLI

ステージ参加者トークンの署名に使用される既存のパブリックキーを取得するには

次のget-public-key例では、ステージ参加者トークンをシグニングするためにARN、指定されたで指定されたパブリックキーを取得します。

```

aws ivs-realtime get-public-key \
  --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2

```

出力:

```

{
  "publicKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
    "name": "",
    "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHkoZIZj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpiWu/

```

```
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcF00AzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
  "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
  "tags": {}
}
}
```

詳細については、[「Amazon Real-Time Streaming ユーザーガイド」](#)の「[参加者トークンの配布](#)」を参照してください。 IVS

- API 詳細については、AWS CLI 「[コマンドリファレンスGetPublicKey](#)」の「」を参照してください。

get-stage-session

次のコード例は、get-stage-session を使用する方法を示しています。

AWS CLI

ステージセッションを取得するには

次のget-stage-session例では、指定されたステージ ARN (Amazon リソースネーム) の指定されたセッション ID のステージセッションを取得します。

```
aws ivs-realtime get-stage-session \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g
```

出力:

```
{
  "stageSession": {
    "endTime": "2023-04-26T20:36:29+00:00",
    "sessionId": "st-a1b2c3d4e5f6g",
    "startTime": "2023-04-26T20:30:29.602000+00:00"
  }
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」](#)の「[Amazon Stream IVS で複数のホストを有効にする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStageSession](#)」の「」を参照してください。

get-stage

次の例は、get-stage を使用する方法を説明しています。

AWS CLI

ステージの設定情報を取得するには

次のget-stage例では、指定されたステージ ARN (Amazon リソースネーム) のステージ設定を取得します。

```
aws ivs-realtime get-stage \  
--arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

出力:

```
{  
  "stage": {  
    "activeSessionId": "st-a1b2c3d4e5f6g",  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",  
    "autoParticipantRecordingConfiguration": {  
      "mediaTypes": [  
        "AUDIO_VIDEO"  
      ],  
      "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-configuration/abcdABCDefgh",  
    },  
    "endpoints": {  
      "events": "wss://global.events.live-video.net",  
      "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"  
    },  
    "name": "test",  
    "tags": {}  
  }  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」](#)の「[Amazon Stream IVS で複数のホストを有効にする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStage](#)」の「」を参照してください。

get-storage-configuration

次のコード例は、get-storage-configuration を使用する方法を示しています。

AWS CLI

コンポジションストレージ設定を取得するには

次のget-storage-configuration例では、指定された ARN (Amazon リソースネーム) で指定されたコンポジションストレージ設定を取得します。

```
aws ivs-realtime get-storage-configuration \  
  --name arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/  
abcdABCDefgh"
```

出力:

```
{  
  "storageConfiguration": {  
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/  
abcdABCDefgh",  
    "name": "test-sc",  
    "s3": {  
      "bucketName": "test-bucket-name"  
    },  
    "tags": {}  
  }  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStorageConfiguration](#)」の「」を参照してください。

import-public-key

次の例は、import-public-key を使用する方法を説明しています。

AWS CLI

ステージ参加者トークンの署名に使用する既存のパブリックキーをインポートするには

次のimport-public-key例では、ステージ参加者トークンのシグニングに使用される公開キーをマテリアルファイルからインポートします。

```
aws ivs-realtime import-public-key \
  --public-key-material="`cat public.pem`"
```

出力:

```
{
  "publicKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
    "name": "",
    "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHkoZIZj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpWu/
uhNStG1teJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcF00AzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
    "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
    "tags": {}
  }
}
```

詳細については、[「Amazon リアルタイムストリーミングユーザーガイド」の「参加者トークンの配布IVS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportPublicKey](#)」の「」を参照してください。

list-compositions

次の例は、list-compositions を使用する方法を説明しています。

AWS CLI

コンポジションのリストを取得するには

以下は、APIリクエストが処理される AWS リージョンにおける AWS アカウントのすべての構成をlist-compositions一覧表示します。

aws ivs-realtime list-compositions

出力:

```
{
  "compositions": [
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
abcdABCDefgh",
      "destinations": [
        {
          "id": "AabBCcdDEeff",
          "startTime": "2023-10-16T23:25:23+00:00",
          "state": "ACTIVE"
        }
      ],
      "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
defgABCDabcd",
      "startTime": "2023-10-16T23:25:21+00:00",
      "state": "ACTIVE",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/
ABcdabCDefgh",
      "destinations": [
        {
          "endTime": "2023-10-16T23:25:00.786512+00:00",
          "id": "aABbcCDdeEFf",
          "startTime": "2023-10-16T23:24:01+00:00",
          "state": "STOPPED"
        },
        {
          "endTime": "2023-10-16T23:25:00.786512+00:00",
          "id": "deEFfaABbcCD",
          "startTime": "2023-10-16T23:24:01+00:00",
          "state": "STOPPED"
        }
      ],
      "endTime": "2023-10-16T23:25:00+00:00",
      "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/
efghabcdABCD",
      "startTime": "2023-10-16T23:24:00+00:00",
    }
  ]
}
```



```
        "state": "STOPPED",
        "tags": {}
    }
]
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCompositions](#)」の「」を参照してください。

list-encoder-configurations

次の例は、list-encoder-configurations を使用する方法を説明しています。

AWS CLI

コンポジションエンコーダ設定を一覧表示するには

以下は、APIリクエストが処理される AWS リージョンにおける AWS、アカウントのすべてのコンポジションエンコーダ設定をlist-encoder-configurations一覧表示します。

```
aws ivs-realtime list-encoder-configurations
```

出力:

```
{
  "encoderConfigurations": [
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/abcdABCDefgh",
      "name": "test-ec-1",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/ABCefgEFGabc",
      "name": "test-ec-2",
      "tags": {}
    }
  ]
}
```

```
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEncoderConfigurations](#)」の「」を参照してください。

list-participant-events

次の例は、list-participant-events を使用する方法を説明しています。

AWS CLI

ステージ参加者イベントのリストを取得するには

次のlist-participant-events例では、指定されたステージ (ARNAmazon リソースネーム) の指定された参加者 ID とセッション ID のすべての参加者イベントを一覧表示します。

```
aws ivs-realtime list-participant-events \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --session-id st-a1b2c3d4e5f6g \  
  --participant-id abCDEf12GHIj
```

出力:

```
{  
  "events": [  
    {  
      "eventTime": "2023-04-26T20:36:28+00:00",  
      "name": "LEFT",  
      "participantId": "abCDEf12GHIj"  
    },  
    {  
      "eventTime": "2023-04-26T20:36:28+00:00",  
      "name": "PUBLISH_STOPPED",  
      "participantId": "abCDEf12GHIj"  
    },  
    {  
      "eventTime": "2023-04-26T20:30:34+00:00",  
      "name": "JOINED",  
      "participantId": "abCDEf12GHIj"  
    }  
  ]  
}
```

```
    },
    {
      "eventTime": "2023-04-26T20:30:34+00:00",
      "name": "PUBLISH_STARTED",
      "participantId": "abCDEf12GHIj"
    }
  ]
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListParticipantEvents](#)」の「」を参照してください。

list-participants

次のコード例は、list-participants を使用する方法を示しています。

AWS CLI

ステージ参加者のリストを取得するには

次のlist-participants例では、指定されたステージ ARN (Amazon リソースネーム) の指定されたセッション ID のすべての参加者を一覧表示します。

```
aws ivs-realtime list-participants \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g
```

出力:

```
{
  "participants": [
    {
      "firstJoinTime": "2023-04-26T20:30:34+00:00",
      "participantId": "abCDEf12GHIj"
      "published": true,
      "recordingState": "STOPPED",
      "state": "DISCONNECTED",
      "userId": ""
    }
  ]
}
```

```
]
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVS で複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListParticipants](#)」の「」を参照してください。

list-public-keys

次のコード例は、list-public-keys を使用する方法を示しています。

AWS CLI

ステージ参加者トークンの署名に使用できる既存のパブリックキーを一覧表示するには

次のlist-public-keys例では、APIリクエストが処理される AWS リージョンで、ステージ参加者トークンのシグニングに使用できるすべてのパブリックキーを一覧表示します。

```
aws ivs-realtime list-public-keys
```

出力:

```
{
  "publicKeys": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
      "name": "",
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/3bcdABCDefg4",
      "name": "",
      "tags": {}
    }
  ]
}
```

詳細については、[「Amazon リアルタイムストリーミングユーザーガイド」の「参加者トークンの配布IVS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPublicKeys](#)」の「」を参照してください。

list-stage-sessions

次の例は、list-stage-sessions を使用する方法を説明しています。

AWS CLI

ステージセッションのリストを取得するには

次のlist-stage-sessions例では、指定されたステージ ARN (Amazon リソースネーム) のすべてのセッションを一覧表示します。

```
aws ivs-realtime list-stage-sessions \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

出力:

```
{  
  "stageSessions": [  
    {  
      "endTime": "2023-04-26T20:36:29+00:00",  
      "sessionId": "st-a1b2c3d4e5f6g",  
      "startTime": "2023-04-26T20:30:29.602000+00:00"  
    }  
  ]  
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」](#)の「[Amazon Stream IVS で複数のホストを有効にする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStageSessions](#)」の「」を参照してください。

list-stages

次のコード例は、list-stages を使用する方法を示しています。

AWS CLI

すべてのステージの概要情報を取得するには

次のlist-stages例では、APIリクエストが処理される AWS リージョンの AWS アカウントのすべてのステージを一覧表示します。

```
aws ivs-realtime list-stages
```

出力:

```
{
  "stages": [
    {
      "activeSessionId": "st-a1b2c3d4e5f6g",
      "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
      "name": "stage1",
      "tags": {}
    },
    {
      "activeSessionId": "st-a123bcd456efg",
      "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcd1234ABCD",
      "name": "stage2",
      "tags": {}
    },
    {
      "activeSessionId": "st-abcDEF1234ghi",
      "arn": "arn:aws:ivs:us-west-2:123456789012:stage/ABCD1234efgh",
      "name": "stage3",
      "tags": {}
    }
  ]
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStages](#)」の「」を参照してください。

list-storage-configurations

次の例は、list-storage-configurations を使用する方法を説明しています。

AWS CLI

コンポジションストレージ設定を一覧表示するには

以下は、APIリクエストが処理される AWS リージョンにおける AWS アカウントのすべてのコンポジションストレージ設定を `list-storage-configurations` 一覧表示します。

```
aws ivs-realtime list-storage-configurations
```

出力:

```
{
  "storageConfigurations": [
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/abcdABCDefgh",
      "name": "test-sc-1",
      "s3": {
        "bucketName": "test-bucket-1-name"
      },
      "tags": {}
    },
    {
      "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/ABCefgEFGabc",
      "name": "test-sc-2",
      "s3": {
        "bucketName": "test-bucket-2-name"
      },
      "tags": {}
    }
  ]
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStorageConfigurations](#)」の「」を参照してください。

start-composition

次のコード例は、start-composition を使用する方法を示しています。

AWS CLI

例 1: デフォルトのレイアウト設定でコンポジションを開始するには

次のstart-composition例では、指定されたステージの構成を開始して、指定された場所にストリーミングします。

```
aws ivs-realtime start-composition \  
  --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCdabcd \  
  --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-  
northeast-1:123456789012:channel/abcABCdefDEg", \  
  "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-  
configuration/ABabCDcdEFef"}}, \  
  {"s3": {"encoderConfigurationArns": ["arn:aws:ivs:ap-  
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \  
  "storageConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-  
configuration/FefABabCDcdE"}}]'
```

出力:

```
{  
  "composition": {  
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",  
    "destinations": [  
      {  
        "configuration": {  
          "channel": {  
            "channelArn": "arn:aws:ivs:ap-  
northeast-1:123456789012:channel/abcABCdefDEg",  
            "encoderConfigurationArn": "arn:aws:ivs:ap-  
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"  
          },  
          "name": ""  
        },  
        "id": "AabBCcdDEefF",  
        "state": "STARTING"  
      },  
      {  
        "configuration": {
```



```

        "name": "",
        "s3": {
            "encoderConfigurationArns": [
                "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
            ],
            "recordingConfiguration": {
                "format": "HLS"
            },
            "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
        }
    },
    "detail": {
        "s3": {
            "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
        }
    },
    "id": "GHFabcgefABC",
    "state": "STARTING"
}
],
"layout": {
    "grid": {
        "featuredParticipantAttribute": "",
        "gridGap": 2,
        "omitStoppedVideo": false,
        "videoAspectRatio": "VIDEO",
        "videoFillMode": ""
    }
},
"stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
"startTime": "2023-10-16T23:24:00+00:00",
"state": "STARTING",
"tags": {}
}
}

```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[複合録画 \(リアルタイムストリーミング\)](#)」を参照してください。

例 2: PiP レイアウトでコンポジションを開始するには

次のstart-composition例では、PiP レイアウトを使用して、指定されたステージを指定された場所にストリーミングする構成を開始します。

```
aws ivs-realtime start-composition \
  --stage-arn arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCdabcd \
  --destinations '[{"channel": {"channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg"}, \
  "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-
configuration/ABabCDcdEFef"}], \
  {"s3":{"encoderConfigurationArns":["arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"], \
  "storageConfigurationArn":"arn:aws:ivs:ap-northeast-1:123456789012:storage-
configuration/FefABabCDcdE"}}]' \
  --layout pip='{featuredParticipantAttribute="abcdefg}"'
```

出力:

```
{
  "composition": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs",
    "destinations": [
      {
        "configuration": {
          "channel": {
            "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
            "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
          },
          "name": ""
        },
        "id": "AabBCcdDEefF",
        "state": "STARTING"
      },
      {
        "configuration": {
          "name": "",
          "s3": {
            "encoderConfigurationArns": [
              "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
            ],
            "recordingConfiguration": {
```

```
        "format": "HLS"
      },
      "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
    }
  },
  "detail": {
    "s3": {
      "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
    }
  },
  "id": "GHFabcgefABC",
  "state": "STARTING"
}
],
"layout": {
  "pip": {
    "featuredParticipantAttribute": "abcdefg",
    "gridGap": 0,
    "omitStoppedVideo": false,
    "pipBehavior": "STATIC",
    "pipOffset": 0,
    "pipParticipantAttribute": "",
    "pipPosition": "BOTTOM_RIGHT",
    "videoFillMode": "COVER"
  }
},
"stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
"startTime": "2023-10-16T23:24:00+00:00",
"state": "STARTING",
"tags": {}
}
}
```

詳細については、「Amazon Interactive Video Service ユーザーガイド」の「[複合録画 \(リアルタイムストリーミング\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartComposition](#)」の「」を参照してください。

stop-composition

次の例は、stop-composition を使用する方法を説明しています。

AWS CLI

コンポジションを停止するには

以下は、指定された ARN (Amazon リソースネーム) で指定された構成stop-compositionを停止します。

```
aws ivs-realtime stop-composition \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopComposition](#)」の「」を参照してください。

update-stage

次の例は、update-stage を使用する方法を説明しています。

AWS CLI

ステージの設定を更新するには

次のupdate-stage例では、指定されたステージのステージを更新ARNしてステージ名を更新し、個々の参加者の録音を設定します。

```
aws ivs-realtime update-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --auto-participant-recording-configuration '{"mediaTypes":  
  ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-  
west-2:123456789012:storage-configuration/abcdABCDefgh"}' \  
  --name stage1a
```

出力:

```
{
  "stage": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
    "autoParticipantRecordingConfiguration": {
      "mediaTypes": [
        "AUDIO_VIDEO"
      ],
      "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-configuration/abcdABCDefgh",
    },
    "endpoints": {
      "events": "wss://global.events.live-video.net",
      "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
    },
    "name": "stage1a",
    "tags": {}
  }
}
```

詳細については、[「Amazon Interactive Video Service ユーザーガイド」の「Amazon Stream IVSで複数のホストを有効にする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateStage](#)」の「」を参照してください。

を使用した Amazon Kendra の例 AWS CLI

次のコード例は、Amazon Kendra AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-data-source

次のコード例は、create-data-source を使用する方法を示しています。

AWS CLI

Amazon Kendra データソースコネクタを作成するには

以下は、Amazon Kendra データソースコネクタ create-data-source を作成して設定します。describe-data-source を使用して、データソースコネクタのステータスを表示し、ステータスにデータソースコネクタ FAILED 「」が表示されている場合はエラーメッセージをすべて読み取って完全に作成できません。

```
aws kendra create-data-source \  
  --name "example data source 1" \  
  --description "Example data source 1 for example index 1 contains the first set  
of example documents" \  
  --tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",  
"Value": "aws"}' \  
  --role-arn "arn:aws:iam::my-account-id:role/  
KendraRoleForS3TemplateConfigDataSource" \  
  --index-id exampleindex1 \  
  --language-code "es" \  
  --schedule "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *" \  
  --configuration '{"TemplateConfiguration": {"Template": file://  
s3schemaconfig.json}}' \  
  --type "TEMPLATE" \  
  --custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":  
{"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",  
"S3Bucket": "s3://my-s3-bucket/scanned-image-text-example-docs"}, "RoleArn":  
"arn:aws:iam:my-account-id:role/KendraRoleForCDE"}' \  
  --vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":  
["subnet-1c234", "subnet-2b134"]}'
```

出力:

```
{  
  "Id": "exampledatasource1"  
}
```

詳細については、[「Amazon Kendra デベロッパーガイド」の「Amazon Kendra インデックスとデータソースコネクタの使用開始」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDataSource](#)」の「」を参照してください。

create-index

次のコード例は、create-index を使用する方法を示しています。

AWS CLI

Amazon Kendra インデックスを作成するには

以下は、Amazon Kendra インデックスcreate-indexを作成して設定します。describe-index を使用してインデックスのステータスを表示し、ステータスが完全に作成するインデックスFAILED「」が表示されている場合は、エラーメッセージを読み取ることができます。

```
aws kendra create-index \  
  --name "example index 1" \  
  --description "Example index 1 contains the first set of example documents" \  
  --tags '{"Key": "test resources", "Value": "kendra"}, {"Key": "test resources",  
  "Value": "aws"}' \  
  --role-arn "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex" \  
  --edition "DEVELOPER_EDITION" \  
  --server-side-encryption-configuration '{"KmsKeyId": "my-kms-key-id"}' \  
  --user-context-policy "USER_TOKEN" \  
  --user-token-configurations '{"JsonTokenTypeConfiguration":  
  {"GroupAttributeField": "groupNameField", "UserNameAttributeField":  
  "userNameField"}}'
```

出力:

```
{  
  "Id": index1  
}
```

詳細については、[「Amazon Kendra デベロッパーガイド」の「Amazon Kendra インデックスとデータソースコネクタの開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateIndex](#)」の「」を参照してください。

describe-data-source

次のコード例は、describe-data-source を使用する方法を示しています。

AWS CLI

Amazon Kendra データソースコネクタに関する情報を取得するには

以下は、Amazon Kendra データソースコネクタに関する情報describe-data-sourceを取得します。ステータスにデータソースコネクタFAILED「」が表示されている場合は、データソースコネクタの設定を表示し、エラーメッセージをすべて読み取ることができます。

```
aws kendra describe-data-source \  
  --id exampledatasource1 \  
  --index-id exampleindex1
```

出力:

```
{  
  "Configuration": {  
    "TemplateConfiguration": {  
      "Template": {  
        "connectionConfiguration": {  
          "repositoryEndpointMetadata": {  
            "BucketName": "my-bucket"  
          }  
        },  
        "repositoryConfigurations": {  
          "document": {  
            "fieldMappings": [  
              {  
                "indexFieldName": "_document_title",  
                "indexFieldType": "STRING",  
                "dataSourceFieldName": "title"  
              },  
              {  
                "indexFieldName": "_last_updated_at",  
                "indexFieldType": "DATE",  
                "dataSourceFieldName": "modified_date"  
              }  
            ]  
          }  
        }  
      }  
    }  
  },  
}
```



```

        "additionalProperties": {
          "inclusionPatterns": [
            "*.txt",
            "*.doc",
            "*.docx"
          ],
          "exclusionPatterns": [
            "*.json"
          ],
          "inclusionPrefixes": [
            "PublicExampleDocsFolder"
          ],
          "exclusionPrefixes": [
            "PrivateDocsFolder/private"
          ],
          "aclConfigurationFilePath": "ExampleDocsFolder/AclConfig.json",
          "metadataFilesPrefix": "metadata"
        },
        "syncMode": "FULL_CRAWL",
        "type": "S3",
        "version": "1.0.0"
      }
    },
    "CreatedAt": "2024-02-25T13:30:10+00:00",
    "CustomDocumentEnrichmentConfiguration": {
      "PostExtractionHookConfiguration": {
        "LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",
        "S3Bucket": "s3://my-s3-bucket/scanned-image-text-example-docs/function"
      },
      "RoleArn": "arn:aws:iam:my-account-id:role/KendraRoleForCDE"
    },
    "Description": "Example data source 1 for example index 1 contains the first set
of example documents",
    "Id": "exampledatasource1",
    "IndexId": "exampleindex1",
    "LanguageCode": "en",
    "Name": "example data source 1",
    "RoleArn": "arn:aws:iam::my-account-id:role/
KendraRoleForS3TemplateConfigDataSource",
    "Schedule": "0 0 18 ? * TUE,MON,WED,THU,FRI,SAT *",
    "Status": "ACTIVE",
    "Type": "TEMPLATE",
    "UpdatedAt": "1709163615",

```

```
"VpcConfiguration": {
  "SecurityGroupIds": ["sg-1234567890abcdef0"],
  "SubnetIds": ["subnet-1c234", "subnet-2b134"]
}
```

詳細については、[「Amazon Kendra デベロッパーガイド」の「Amazon Kendra インデックスとデータソースコネクタの使用開始」](#)を参照してください。

- API 詳細については、「[コマンドリファレンスDescribeDataSource](#)」の「」を参照してください。AWS CLI

describe-index

次のコード例は、describe-index を使用する方法を示しています。

AWS CLI

Amazon Kendra インデックスに関する情報を取得するには

以下は、Amazon Kendra インデックスに関する情報describe-indexを取得します。インデックスの設定を表示し、ステータスにインデックスFAILED「」が表示されている場合はエラーメッセージをすべて読み取って完全に作成できます。

```
aws kendra describe-index \
  --id exampleindex1
```

出力:

```
{
  "CapacityUnits": {
    "QueryCapacityUnits": 0,
    "StorageCapacityUnits": 0
  },
  "CreatedAt": 2024-02-25T12:30:10+00:00,
  "Description": "Example index 1 contains the first set of example documents",
  "DocumentMetadataConfigurations": [
    {
      "Name": "_document_title",
      "Relevance": {
        "Importance": 8
      }
    }
  ],
}
```

```
    "Search": {
      "Displayable": true,
      "Facetable": false,
      "Searchable": true,
      "Sortable": false
    },
    "Type": "STRING_VALUE"
  },
  {
    "Name": "_document_body",
    "Relevance": {
      "Importance": 5
    },
    "Search": {
      "Displayable": true,
      "Facetable": false,
      "Searchable": true,
      "Sortable": false
    },
    "Type": "STRING_VALUE"
  },
  {
    "Name": "_last_updated_at",
    "Relevance": {
      "Importance": 6,
      "Duration": "2628000s",
      "Freshness": true
    },
    "Search": {
      "Displayable": true,
      "Facetable": false,
      "Searchable": true,
      "Sortable": true
    },
    "Type": "DATE_VALUE"
  },
  {
    "Name": "department_custom_field",
    "Relevance": {
      "Importance": 7,
      "ValueImportanceMap": {
        "Human Resources" : 4,
        "Marketing and Sales" : 2,
        "Research and innvoation" : 3,

```

```
        "Admin" : 1
      }
    },
    "Search": {
      "Displayable": true,
      "Facetable": true,
      "Searchable": true,
      "Sortable": true
    },
    "Type": "STRING_VALUE"
  }
],
"Edition": "DEVELOPER_EDITION",
"Id": "index1",
"IndexStatistics": {
  "FaqStatistics": {
    "IndexedQuestionAnswersCount": 10
  },
  "TextDocumentStatistics": {
    "IndexedTextBytes": 1073741824,
    "IndexedTextDocumentsCount": 1200
  }
},
"Name": "example index 1",
"RoleArn": "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex",
"ServerSideEncryptionConfiguration": {
  "KmsKeyId": "my-kms-key-id"
},
"Status": "ACTIVE",
"UpdatedAt": 1709163615,
"UserContextPolicy": "USER_TOKEN",
"UserTokenConfigurations": [
  {
    "JsonTokenTypeConfiguration": {
      "GroupAttributeField": "groupNameField",
      "UserNameAttributeField": "userNameField"
    }
  }
]
}
```

詳細については、[「Amazon Kendra デベロッパーガイド」の「Amazon Kendra インデックスとデータソースコネクタの使用開始」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeIndex](#)」の「」を参照してください。

update-data-source

次のコード例は、update-data-source を使用する方法を示しています。

AWS CLI

Amazon Kendra データソースコネクタを更新するには

以下は、Amazon Kendra データソースコネクタの設定update-data-sourceを更新します。アクションが成功すると、サービスは出力なし、HTTPステータスコード 200、または AWS CLI リターンコード 0 を返します。を使用してdescribe-data-source、データソースコネクタの設定とステータスを表示できます。

```
aws kendra update-data-source \  
  --id exampledatasource1 \  
  --index-id exampleindex1 \  
  --name "new name for example data source 1" \  
  --description "new description for example data source 1" \  
  --role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForExampleDataSource \  
  --configuration '{"TemplateConfiguration": {"Template": file://  
s3schemanewconfig.json}}' \  
  --custom-document-enrichment-configuration '{"PostExtractionHookConfiguration":  
{"LambdaArn": "arn:aws:iam::my-account-id:function/my-function-ocr-docs",  
"S3Bucket": "s3://my-s3-bucket/scanned-image-text-example-docs"}, "RoleArn":  
"arn:aws:iam:my-account-id:role/KendraNewRoleForCDE"}' \  
  --language-code "es" \  
  --schedule "0 0 18 ? * MON,WED,FRI *" \  
  --vpc-configuration '{"SecurityGroupIds": ["sg-1234567890abcdef0"], "SubnetIds":  
["subnet-1c234", "subnet-2b134"]}'
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Kendra デベロッパーガイド」の「Amazon Kendra インデックスとデータソースコネクタの開始方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDataSource](#)」の「」を参照してください。

update-index

次の例は、update-index を使用する方法を説明しています。

AWS CLI

Amazon Kendra インデックスを更新するには

以下は、Amazon Kendra インデックスの設定update-indexを更新します。アクションが成功すると、サービスは出力なし、HTTPステータスコード 200、または AWS CLIリターンコード 0 を返します。describe-index を使用して、インデックスの設定とステータスを表示できます。

```
aws kendra update-index \  
  --id enterpriseindex1 \  
  --name "new name for Enterprise Edition index 1" \  
  --description "new description for Enterprise Edition index 1" \  
  --role-arn arn:aws:iam::my-account-id:role/KendraNewRoleForEnterpriseIndex \  
  --capacity-units '{"QueryCapacityUnits": 2, "StorageCapacityUnits": 1}' \  
  --document-metadata-configuration-updates '{"Name": "_document_title",  
  "Relevance": {"Importance": 6}}, {"Name": "_last_updated_at", "Relevance":  
  {"Importance": 8}}' \  
  --user-context-policy "USER_TOKEN" \  
  --user-token-configurations '{"JsonTokenTypeConfiguration":  
  {"GroupAttributeField": "groupNameField", "UserNameAttributeField":  
  "userNameField"}}'
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Kendra デベロッパーガイド」の「Amazon Kendra インデックスとデータソースコネクタの使用開始」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIndex](#)」の「」を参照してください。

を使用した Kinesis の例 AWS CLI

次のコード例は、Kinesis AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags-to-stream

次の例は、`add-tags-to-stream` を使用する方法を説明しています。

AWS CLI

データストリームにタグを追加するには

次の`add-tags-to-stream`例では、キー`samplekey`と値を持つタグを指定されたストリーム`example`に割り当てます。

```
aws kinesis add-tags-to-stream \  
  --stream-name samplestream \  
  --tags samplekey=example
```

このコマンドでは何も出力されません。

詳細については、「Amazon Kinesis Data [Streams デベロッパーガイド](#)」の「[ストリームのタグ付け](#)」を参照してください。Amazon Kinesis

- API 詳細については、AWS CLI 「[コマンドリファレンスAddTagsToStream](#)」の「」を参照してください。

create-stream

次のコード例は、`create-stream` を使用する方法を示しています。

AWS CLI

データストリームを作成するには

次の `create-stream` の例は、3 つのシャードを含む `samplestream` という名前のデータストリームを作成します。

```
aws kinesis create-stream \  
  --stream-name samplestream \  
  --shard-count 3
```

このコマンドでは何も出力されません。

詳細については、「Amazon Kinesis Data Streams デベロッパーガイド」の「[API を使用したストリームの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStream](#)」の「」を参照してください。

decrease-stream-retention-period

次の例は、`decrease-stream-retention-period` を使用する方法を説明しています。

AWS CLI

データストリームの保持期間を短縮するには

次の `decrease-stream-retention-period` 例では、サンプルストリームという名前のストリームの保持期間 (データレコードがストリームに追加されてからアクセス可能になる期間) を 48 時間に短縮します。

```
aws kinesis decrease-stream-retention-period \  
  --stream-name samplestream \  
  --retention-period-hours 48
```

このコマンドでは何も出力されません。

詳細については、「Amazon Kinesis Data Streams デベロッパーガイド」の「データ[保持期間の変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DecreaseStreamRetentionPeriod](#)」の「」を参照してください。

delete-stream

次の例は、`delete-stream` を使用する方法を説明しています。

AWS CLI

データストリームを削除するには

次の `delete-stream` の例は、指定されたデータストリームを削除します。

```
aws kinesis delete-stream \  
  --stream-name samplestream
```

このコマンドでは何も出力されません。

詳細については、「Amazon Kinesis Data Streams デイベロッパーガイド」の「[ストリームを削除する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStream](#)」の「」を参照してください。

deregister-stream-consumer

次のコード例は、`deregister-stream-consumer` を使用する方法を示しています。

AWS CLI

データストリームコンシューマーの登録を解除するには

次の `deregister-stream-consumer` 例では、指定されたコンシューマーを指定されたデータストリームから登録解除します。

```
aws kinesis deregister-stream-consumer \  
  --stream-arn arn:aws:kinesis:us-west-2:123456789012:stream/samplestream \  
  --consumer-name KinesisConsumerApplication
```

このコマンドでは何も出力されません。

詳細については、「Amazon [Kinesis Data Streams デベロッパーガイド](#)」の「[Kinesis Data Streams を使用した拡張ファンアウトによるコンシューマーAPIの開発](#)」を参照してください。

Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterStreamConsumer](#)」の「」を参照してください。

describe-limits

次の例は、describe-limits を使用する方法を説明しています。

AWS CLI

シャードの制限を記述するには

次のdescribe-limits例は、現在の AWS アカウントのシャード制限と使用状況を示しています。

```
aws kinesis describe-limits
```

出力:

```
{
  "ShardLimit": 500,
  "OpenShardCount": 29
}
```

詳細については、[「Amazon Kinesis Data Streams デベロッパーガイド」の「ストリームの再シャードイング」](#)を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLimits](#)」の「」を参照してください。

describe-stream-consumer

次のコード例は、describe-stream-consumer を使用する方法を示しています。

AWS CLI

データストリームコンシューマーを記述するには

次のdescribe-stream-consumer例では、指定されたデータストリームに登録された、指定されたコンシューマーの説明を返します。

```
aws kinesis describe-stream-consumer \
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
  --consumer-name KinesisConsumerApplication
```

出力:

```
{
  "ConsumerDescription": {
    "ConsumerName": "KinesisConsumerApplication",
    "ConsumerARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream/
consumer/KinesisConsumerApplication:1572383852",
    "ConsumerStatus": "ACTIVE",
    "ConsumerCreationTimestamp": 1572383852.0,
    "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream"
  }
}
```

詳細については、[Amazon Kinesis Data Streams デベロッパーガイド](#)の「[Amazon Kinesis Data Streams からのデータの読み取り](#)」を参照してください。Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStreamConsumer](#)」の「」を参照してください。

describe-stream-summary

次の例は、describe-stream-summary を使用する方法を説明しています。

AWS CLI

データストリームの概要を記述するには

次のdescribe-stream-summary例では、指定されたデータストリームの概要説明 (シャードリストなし) を示します。

```
aws kinesis describe-stream-summary \
  --stream-name samplestream
```

出力:

```
{
  "StreamDescriptionSummary": {
    "StreamName": "samplestream",
    "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
    "StreamStatus": "ACTIVE",
    "RetentionPeriodHours": 48,
    "StreamCreationTimestamp": 1572297168.0,
    "EnhancedMonitoring": [
      {
```

```
        "ShardLevelMetrics": []
      }
    ],
    "EncryptionType": "NONE",
    "OpenShardCount": 3,
    "ConsumerCount": 0
  }
}
```

詳細については、「Amazon Kinesis Data Streams デイベロッパーガイド」の「[ストリームの作成と管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStreamSummary](#)」の「」を参照してください。

describe-stream

次のコード例は、describe-stream を使用する方法を示しています。

AWS CLI

データストリームを記述するには

次の describe-stream の例は、指定されたデータストリームの詳細を返します。

```
aws kinesis describe-stream \  
  --stream-name samplestream
```

出力:

```
{  
  "StreamDescription": {  
    "Shards": [  
      {  
        "ShardId": "shardId-000000000000",  
        "HashKeyRange": {  
          "StartingHashKey": "0",  
          "EndingHashKey": "113427455640312821154458202477256070484"  
        },  
        "SequenceNumberRange": {  
          "StartingSequenceNumber":  
            "49600871682957036442365024926191073437251060580128653314"  
        }  
      }  
    ]  
  }  
}
```

```

    },
    {
      "ShardId": "shardId-000000000001",
      "HashKeyRange": {
        "StartingHashKey": "113427455640312821154458202477256070485",
        "EndingHashKey": "226854911280625642308916404954512140969"
      },
      "SequenceNumberRange": {
        "StartingSequenceNumber":
"4960087168297933718756355549332609155523708941634633746"
      }
    },
    {
      "ShardId": "shardId-000000000002",
      "HashKeyRange": {
        "StartingHashKey": "226854911280625642308916404954512140970",
        "EndingHashKey": "340282366920938463463374607431768211455"
      },
      "SequenceNumberRange": {
        "StartingSequenceNumber":
"49600871683001637932762086172474144873796357303140614178"
      }
    }
  ],
  "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
  "StreamName": "samplestream",
  "StreamStatus": "ACTIVE",
  "RetentionPeriodHours": 24,
  "EnhancedMonitoring": [
    {
      "ShardLevelMetrics": []
    }
  ],
  "EncryptionType": "NONE",
  "KeyId": null,
  "StreamCreationTimestamp": 1572297168.0
}
}

```

詳細については、「Amazon Kinesis Data Streams デイベロッパーガイド」の「[ストリームの作成と管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStream](#)」の「」を参照してください。

disable-enhanced-monitoring

次のコード例は、`disable-enhanced-monitoring` を使用する方法を示しています。

AWS CLI

シャードレベルのメトリクスの拡張モニタリングを無効にするには

次の`disable-enhanced-monitoring`例では、シャードレベルのメトリクスの拡張 Kinesis データストリームモニタリングを無効にします。

```
aws kinesis disable-enhanced-monitoring \  
  --stream-name samplestream --shard-level-metrics ALL
```

出力:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardLevelMetrics": [  
    "IncomingBytes",  
    "OutgoingRecords",  
    "IteratorAgeMilliseconds",  
    "IncomingRecords",  
    "ReadProvisionedThroughputExceeded",  
    "WriteProvisionedThroughputExceeded",  
    "OutgoingBytes"  
  ],  
  "DesiredShardLevelMetrics": []  
}
```

詳細については、[Amazon Kinesis Data Streams デベロッパーガイド](#)の「[Amazon Kinesis Data Streams でのストリームのモニタリング](#)」を参照してください。Amazon Kinesis

- API 詳細については、AWS CLI 「[コマンドリファレンスDisableEnhancedMonitoring](#)」の「」を参照してください。

enable-enhanced-monitoring

次のコード例は、`enable-enhanced-monitoring` を使用する方法を示しています。

AWS CLI

シャードレベルのメトリクスの拡張モニタリングを有効にするには

次のenable-enhanced-monitoring例では、シャードレベルのメトリクスの拡張 Kinesis データストリームモニタリングを有効にします。

```
aws kinesis enable-enhanced-monitoring \  
  --stream-name samplestream \  
  --shard-level-metrics ALL
```

出力:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardLevelMetrics": [],  
  "DesiredShardLevelMetrics": [  
    "IncomingBytes",  
    "OutgoingRecords",  
    "IteratorAgeMilliseconds",  
    "IncomingRecords",  
    "ReadProvisionedThroughputExceeded",  
    "WriteProvisionedThroughputExceeded",  
    "OutgoingBytes"  
  ]  
}
```

詳細については、[Amazon Kinesis Data Streams デベロッパーガイド](#)の「[Amazon Kinesis Data Streams でのストリームのモニタリング](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableEnhancedMonitoring](#)」の「」を参照してください。

get-records

次の例は、get-records を使用する方法を説明しています。

AWS CLI

シャードからレコードを取得するには

次の get-records の例は、指定されたシャードイテレーターを使用して Kinesis データストリームのシャードからデータレコードを取得します。

```
aws kinesis get-records \  
  --shard-iterator-type AT_TIMESTAMP \  
  --shard-id shard-1 \  
  --max-records 10
```

```
--shard-iterator AAAAAAAAAAF7/0mWD7IuHj1yGv/
TKuNgx2ukD5xipCY4cy4gU96orWwZwcSXh3K9tAmGYe0ZyLZrvzze0FVf9iN99hUPw/w/
b0YWYeefNvnf1DYt5XpDJghLKr3DzgzknkTmMymDP3R+3wRKeuEw6/kdxY2yKJH0veaiekaVc4N2VwK/
GvaGP2Hh9Fg7N++q0Adg6fIDQPt4p8RpavDbk+A4sL9SWGE1
```

出力:

```
{
  "Records": [],
  "MillisBehindLatest": 80742000
}
```

詳細については、「Amazon [Kinesis Data Streams デベロッパーガイド](#)」の「[for Java APIでの AWS SDK Kinesis Data Streams を使用したコンシューマーの開発](#)」を参照してください。

Amazon Kinesis

- API 詳細については、AWS CLI 「[コマンドリファレンスGetRecords](#)」の「」を参照してください。

get-shard-iterator

次のコード例は、get-shard-iterator を使用する方法を示しています。

AWS CLI

シャードイテレータを取得するには

次のget-shard-iterator例ではAT_SEQUENCE_NUMBER、シャードイテレータタイプを使用してシャードイテレータを生成し、指定されたシーケンス番号で示される位置からデータレコードの読み取りを正確に開始します。

```
aws kinesis get-shard-iterator \
  --stream-name samplestream \
  --shard-id shardId-00000000001 \
  --shard-iterator-type LATEST
```

出力:

```
{
  "ShardIterator": "AAAAAAAAAAFEvJjIYI+3jw/4aqqH9FifJ+n48XWTh/
IFIsbILP6o5eDueD39NXNBfpZ10WL5K6ADXk8w+5H+Qhd9cFA9k268CPXCz/kebq1TGYI7Vy
```



```
+lUkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn9408SqeEqwhigWzXNWxID3Ej7YYYcxQi8Q/fIrCjGAy/  
n2r5Z9G864YpWdfN9upNNQAR/ii0WKs"  
}
```

詳細については、「Amazon [Kinesis Data Streams デベロッパーガイド](#)」の「[for Java APIでの AWS SDK Kinesis Data Streams を使用したコンシューマーの開発](#)」を参照してください。

Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[GetShardIterator](#)」の「」を参照してください。

increase-stream-retention-period

次の例は、increase-stream-retention-period を使用する方法を説明しています。

AWS CLI

データストリームの保持期間を延長するには

次のincrease-stream-retention-period例では、指定されたストリームの保持期間 (データレコードがストリームに追加されてからアクセス可能になる時間) を 168 時間に延長します。

```
aws kinesis increase-stream-retention-period \  
  --stream-name samplestream \  
  --retention-period-hours 168
```

このコマンドでは何も出力されません。

詳細については、Amazon Kinesis Data Streams デベロッパーガイド」の「データ[保持期間の変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[IncreaseStreamRetentionPeriod](#)」の「」を参照してください。

list-shards

次のコード例は、list-shards を使用する方法を示しています。

AWS CLI

データストリーム内のシャードを一覧表示するには

次のlist-shards例では、指定されたストリーム内のすべてのシャードを一覧表示します。これは、ID が exclusive-start-shard-idの指定された直後のシャードから始まりまずshardId-000000000000。

```
aws kinesis list-shards \  
  --stream-name samplestream \  
  --exclusive-start-shard-id shardId-000000000000
```

出力:

```
{  
  "Shards": [  
    {  
      "ShardId": "shardId-000000000001",  
      "HashKeyRange": {  
        "StartingHashKey": "113427455640312821154458202477256070485",  
        "EndingHashKey": "226854911280625642308916404954512140969"  
      },  
      "SequenceNumberRange": {  
        "StartingSequenceNumber":  
"49600871682979337187563555549332609155523708941634633746"  
      }  
    },  
    {  
      "ShardId": "shardId-000000000002",  
      "HashKeyRange": {  
        "StartingHashKey": "226854911280625642308916404954512140970",  
        "EndingHashKey": "340282366920938463463374607431768211455"  
      },  
      "SequenceNumberRange": {  
        "StartingSequenceNumber":  
"49600871683001637932762086172474144873796357303140614178"  
      }  
    }  
  ]  
}
```

詳細については、[「Amazon Kinesis Data Streams デベロッパーガイド」の「シャードの一覧表示」](#)を参照してください。Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[ListShards](#)」の「」を参照してください。

list-streams

次のコード例は、list-streams を使用する方法を示しています。

AWS CLI

データストリームを一覧表示するには

次の list-streams の例は、現在のアカウントとリージョンのすべてのアクティブなデータストリームを一覧表示します。

```
aws kinesisis list-streams
```

出力:

```
{
  "StreamNames": [
    "samplestream",
    "samplestream1"
  ]
}
```

詳細については、「Amazon Kinesis Data Streams デイベロッパーガイド」の「[ストリームのリスト](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStreams](#)」の「」を参照してください。

list-tags-for-stream

次の例は、list-tags-for-stream を使用する方法を説明しています。

AWS CLI

データストリームのタグを一覧表示するには

次のlist-tags-for-stream例では、指定されたデータストリームにアタッチされたタグを一覧表示します。

```
aws kinesisis list-tags-for-stream \
```

```
--stream-name samplestream
```

出力:

```
{
  "Tags": [
    {
      "Key": "samplekey",
      "Value": "example"
    }
  ],
  "HasMoreTags": false
}
```

詳細については、「Amazon Kinesis Data [Streams デベロッパーガイド](#)」の「[ストリームのタグ付け](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForStream](#)」の「」を参照してください。

merge-shards

次のコード例は、merge-shards を使用する方法を示しています。

AWS CLI

シャードをマージするには

次のmerge-shards例では、指定されたデータストリームで2つの隣接するシャードを IDs shardId-000000000000 と shardId-000000000001 にマージし、それらを1つのシャードに結合します。

```
aws kinesis merge-shards \  
  --stream-name samplestream \  
  --shard-to-merge shardId-000000000000 \  
  --adjacent-shard-to-merge shardId-000000000001
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Kinesis Data Streams デベロッパーガイド](#)」の「[2つのシャードのマージ](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[MergeShards](#)」の「」を参照してください。

put-record

次の例は、put-record を使用する方法を説明しています。

AWS CLI

データストリームにデータレコードを書き込むには

次の put-record の例は、指定されたパーティションキーを使用して、指定されたデータストリームに単一のデータレコードを書き込みます。

```
aws kinesis put-record \  
  --stream-name samplestream \  
  --data sampledatarecord \  
  --partition-key samplepartitionkey
```

出力:

```
{  
  "ShardId": "shardId-0000000000009",  
  "SequenceNumber": "49600902273357540915989931256901506243878407835297513618",  
  "EncryptionType": "KMS"  
}
```

詳細については、[Amazon Kinesis Data Streams デベロッパーガイド](#)の「for Java APIでの [AWS SDK Amazon Kinesis Data Streams を使用したプロデューサーの開発](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRecord](#)」の「」を参照してください。

put-records

次のコード例は、put-records を使用する方法を示しています。

AWS CLI

データストリームに複数のレコードを書き込むには

次のput-records例では、指定されたパーティションキーを使用してデータレコードを書き込み、1回の呼び出しで別のパーティションキーを使用して別のデータレコードを書き込みます。

```
aws kinesis put-records \  
  --stream-name samplestream \  
  --  
records Data=blob1,PartitionKey=partitionkey1 Data=blob2,PartitionKey=partitionkey2
```

出力:

```
{  
  "FailedRecordCount": 0,  
  "Records": [  
    {  
      "SequenceNumber":  
"49600883331171471519674795588238531498465399900093808706",  
      "ShardId": "shardId-000000000004"  
    },  
    {  
      "SequenceNumber":  
"49600902273357540915989931256902715169698037101720764562",  
      "ShardId": "shardId-000000000009"  
    }  
  ],  
  "EncryptionType": "KMS"  
}
```

詳細については、[Amazon Kinesis Data Streams デベロッパーガイド](#)の「for Java APIでのAWS SDK Amazon Kinesis Data Streamsを使用したプロデューサーの開発」を参照してください。Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[PutRecords](#)」の「」を参照してください。

register-stream-consumer

次の例は、register-stream-consumer を使用する方法を説明しています。

AWS CLI

データストリームコンシューマーを登録するには

次のregister-stream-consumer例では、というコンシューマーKinesisConsumerApplicationを指定されたデータストリームに登録します。

```
aws kinesis register-stream-consumer \  
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \  
  --consumer-name KinesisConsumerApplication
```

出力:

```
{  
  "Consumer": {  
    "ConsumerName": "KinesisConsumerApplication",  
    "ConsumerARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream/  
consumer/KinesisConsumerApplication:1572383852",  
    "ConsumerStatus": "CREATING",  
    "ConsumerCreationTimestamp": 1572383852.0  
  }  
}
```

詳細については、「[Amazon Kinesis Data Streams デベロッパーガイド](#)」の「[Kinesis Data Streams を使用した拡張ファンアウトによるコンシューマーAPIの開発](#)」を参照してください。

Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterStreamConsumer](#)」の「」を参照してください。

remove-tags-from-stream

次のコード例は、remove-tags-from-stream を使用する方法を示しています。

AWS CLI

データストリームからタグを削除するには

次のremove-tags-from-stream例では、指定されたキーを持つ タグを指定されたデータストリームから削除します。

```
aws kinesis remove-tags-from-stream \  
  --stream-name samplestream \  
  --tag-keys samplekey
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Kinesis Data Streams デベロッパーガイド](#)」の「[ストリームのタグ付け](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「[コマンドリファレンスRemoveTagsFromStream](#)」の「」を参照してください。

split-shard

次の例は、split-shard を使用する方法を説明しています。

AWS CLI

シャードを分割するには

次のsplit-shard例では、新しい開始ハッシュキー 10 を使用して、指定されたシャードを 2 つの新しいシャードに分割します。

```
aws kinesis split-shard \  
  --stream-name samplestream \  
  --shard-to-split shardId-000000000000 \  
  --new-starting-hash-key 10
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Kinesis Data Streams デベロッパーガイド](#)」の「[シャードの分割](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「[コマンドリファレンスSplitShard](#)」の「」を参照してください。

start-stream-encryption

次のコード例は、start-stream-encryption を使用する方法を示しています。

AWS CLI

データストリーム暗号化を有効にするには

次のstart-stream-encryption例では、指定されたキーを使用して、指定された AWS KMS ストリームのサーバー側の暗号化を有効にします。


```
aws kinesis start-stream-encryption \  
  --encryption-type KMS \  
  --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-  
b334-4d3eb496e452 \  
  --stream-name samplestream
```

このコマンドでは何も出力されません。

詳細については、[Amazon Kinesis Data Streams デベロッパーガイド](#)の「[Amazon Kinesis Data Streams でのデータ保護](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[StartStreamEncryption](#)」の「」を参照してください。

stop-stream-encryption

次のコード例は、stop-stream-encryption を使用する方法を示しています。

AWS CLI

データストリームの暗号化を無効にするには

次のstop-stream-encryption例では、指定されたキーを使用して、指定された AWS KMS ストリームのサーバー側の暗号化を無効にします。

```
aws kinesis start-stream-encryption \  
  --encryption-type KMS \  
  --key-id arn:aws:kms:us-west-2:012345678912:key/a3c4a7cd-728b-45dd-  
b334-4d3eb496e452 \  
  --stream-name samplestream
```

このコマンドでは何も出力されません。

詳細については、[Amazon Kinesis Data Streams デベロッパーガイド](#)の「[Amazon Kinesis Data Streams でのデータ保護](#)」を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[StopStreamEncryption](#)」の「」を参照してください。

update-shard-count

次の例は、update-shard-count を使用する方法を説明しています。

AWS CLI

データストリームのシャード数を更新するには

次のupdate-shard-count例では、指定されたデータストリームのシャード数を 6 に更新します。この例では、均一なスケーリングを使用して、同じサイズのシャードを作成します。

```
aws kinesis update-shard-count \  
  --stream-name samplestream \  
  --scaling-type UNIFORM_SCALING \  
  --target-shard-count 6
```

出力:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardCount": 3,  
  "TargetShardCount": 6  
}
```

詳細については、[「Amazon Kinesis Data Streams デベロッパーガイド」の「ストリームの再シャードイング」](#)を参照してください。 Amazon Kinesis

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateShardCount](#)」の「」を参照してください。

AWS KMS を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS KMS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

cancel-key-deletion

次のコード例は、cancel-key-deletion を使用する方法を示しています。

AWS CLI

カスタマーマネージドKMSキーのスケジュールされた削除をキャンセルするには

次のcancel-key-deletion例では、カスタマーマネージドKMSキーのスケジュールされた削除をキャンセルします。

```
aws kms cancel-key-deletion \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

cancel-key-deletion コマンドが成功すると、スケジュールされた削除はキャンセルされます。ただし、KMSキーのキー状態は `Disabled` であるため、暗号化オペレーションでKMSキーを使用することはできません。機能を復元するには、enable-key コマンド を使用します。

詳細については、[「Key Management Service デベロッパーガイド」の「キーの削除のスケジュールとキャンセル」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelKeyDeletion](#)」の「」を参照してください。

connect-custom-key-store

次のコード例は、connect-custom-key-store を使用する方法を示しています。

AWS CLI

カスタムキーストアを接続するには

次のconnect-custom-key-store例では、指定されたカスタムキーストアを再接続します。このようなコマンドを使用して、カスタムキーストアを初めて接続したり、切断されたキーストアを再接続したりできます。

このコマンドを使用して、AWS CloudHSM キーストアまたは外部キーストアを接続できます。

```
aws kms connect-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

このコマンドは出力を返しません。コマンドが有効であることを確認するには、describe-custom-key-stores コマンドを使用します。

AWS CloudHSM キーストアの接続については、[「Key Management Service デベロッパーガイド」の AWS 「CloudHSM キーストアの接続と切断」](#)を参照してください。AWS

外部キーストアの接続については、[「Key Management Service デベロッパーガイド」の「外部キーストアの接続と切断」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ConnectCustomKeyStore](#)」の「」を参照してください。

create-alias

次の例は、create-alias を使用する方法を説明しています。

AWS CLI

KMSキーのエイリアスを作成するには

次のcreate-aliasコマンドは、KMSキー ID で識別されるキー example-alias に という名前のエイリアスを作成します1234abcd-12ab-34cd-56ef-1234567890ab。

エイリアス名は alias/ で始める必要があります。で始まるエイリアス名は使用しないでくださいalias/aws。これらは が使用するために予約されています AWS。

```
aws kms create-alias \  
  --alias-name alias/example-alias \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドは出力を返しません。新しいエイリアスを確認するには、list-aliases コマンドを使用します。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[エイリアスの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAlias](#)」の「」を参照してください。

create-custom-key-store

次の例は、create-custom-key-store を使用する方法を説明しています。

AWS CLI

例 1: AWS CloudHSM キーストアを作成するには

次のcreate-custom-key-store例では、必要なパラメータを使用して AWS CloudHSM クラスタによってバックアップされた AWS CloudHSM キーストアを作成します。を追加することもできますcustom-key-store-type`parameter with the default value: ``AWS_CLOUDHSM。

でtrust-anchor-certificateコマンドのファイル入力を指定するには AWS CLI、file://プレフィックスが必要です。

```
aws kms create-custom-key-store \  
  --custom-key-store-name ExampleCloudHSMKeyStore \  
  --cloud-hsm-cluster-id cluster-1a23b4cdefg \  
  --key-store-password kmsPswd \  
  --trust-anchor-certificate file://customerCA.crt
```

出力:

```
{  
  "CustomKeyId": cks-1234567890abcdef0  
}
```

詳細については、「[Key Management Service デベロッパーガイド](#)」の AWS 「[クラウドHSM キーストアの作成](#)」を参照してください。 AWS

例 2: パブリックエンドポイント接続を使用して外部キーストアを作成するには

次のcreate-custom-key-store例では、インターネット経由でと AWS KMS通信する外部キーストア (XKS) を作成します。

この例では、はオプションのプレフィックス `XksProxyUriPath` を使用します `example-prefix`。

NOTE: バージョン 1.0 を使用する場合は AWS CLI、パラメータなどの HTTP または HTTPS 値を持つ `XksProxyUriEndpoint` パラメータを指定する前に、次のコマンドを実行します。

```
aws configure set cli_follow_urlparam false
```

それ以外の場合、AWS CLI バージョン 1.0 はパラメータ値をその URI アドレスにあるコンテンツに置き換えます。

```
aws kms create-custom-key-store \  
  --custom-key-store-name ExamplePublicEndpointXKS \  
  --custom-key-store-type EXTERNAL_KEY_STORE \  
  --xks-proxy-connectivity PUBLIC_ENDPOINT \  
  --xks-proxy-uri-endpoint "https://myproxy.xks.example.com" \  
  --xks-proxy-uri-path "/example-prefix/kms/xks/v1" \  
  --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
RawSecretAccessKey=DXjSUawne12fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

出力:

```
{  
  "CustomKeyStoreId": cks-2234567890abcdef0  
}
```

詳細については、[「Key Management Service デベロッパーガイド」の「外部キーストアの作成AWS」](#)を参照してください。

例 3: VPC エンドポイントサービス接続を使用して外部キーストアを作成するには

次の `create-custom-key-store` 例では、Amazon VPC エンドポイントサービスを使用してと通信する外部キーストア (XKS) を作成します AWS KMS。

NOTE: バージョン 1.0 を使用する場合は AWS CLI、パラメータなどの HTTP または HTTPS 値を持つ `XksProxyUriEndpoint` パラメータを指定する前に、次のコマンドを実行します。

```
aws configure set cli_follow_urlparam false
```

それ以外の場合、AWS CLI バージョン 1.0 はパラメータ値をその URI アドレスにあるコンテンツに置き換えます。

```
aws kms create-custom-key-store \  
  --custom-key-store-name ExampleVPCEndpointXKS \  
  --custom-key-store-type EXTERNAL_KEY_STORE \  
  --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \  
  --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \  
  --xks-proxy-uri-path "/kms/xks/v1" \  
  --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-example1" \  
  --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,RawSecretAccessKey=DXjSUawne12fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

出力:

```
{  
  "CustomKeyId": cks-3234567890abcdef0  
}
```

詳細については、[「Key Management Service デベロッパーガイド」の「外部キーストアの作成AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomKeyStore](#)」の「」を参照してください。

create-grant

次の例は、create-grant を使用する方法を説明しています。

AWS CLI

権限を作成するには

次のcreate-grant例では、exampleUserユーザーが1234abcd-12ab-34cd-56ef-1234567890abサンプルKMSキーで decrypt コマンドを使用できるようにする許可を作成します。廃止するプリンシパルは adminRole ロールです。この権限は EncryptionContextSubset 権限の制約を使用して "Department": "IT" キー値ペアを含む decrypt リクエストでの暗号化のみを許可します。

```
aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::123456789012:user/exampleUser \  
  --operations Decrypt \  
  --
```

```
--constraints EncryptionContextSubset={Department=IT} \  
--retiring-principal arn:aws:iam::123456789012:role/adminRole
```

出力:

```
{  
  "GrantId": "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",  
  "GrantToken": "<grant token here>"  
}
```

権限に関する詳細情報を表示するには、`list-grants` コマンドを使用します。

詳細については、AWS 「Key Management Service デベロッパーガイド」の「[のグラント AWS KMS](#)」を参照してください。

- API 詳細については、「[コマンドリファレンスCreateGrant](#)」の「」を参照してください。
AWS CLI

create-key

次のコード例は、`create-key` を使用する方法を示しています。

AWS CLI

例 1: でカスターマネージドKMSキーを作成するには AWS KMS

次の`create-key`例では、対称暗号化KMSキーを作成します。

対称暗号化KMSキーである基本キーを作成するには、パラメータを指定する必要はありません。対称暗号化キーは、これらのパラメータのデフォルト値によって作成されます。

このコマンドはキーポリシーを指定しないため、KMSキーはプログラムで作成された[キーのデフォルトのキーポリシー](#)を取得します。KMSキーポリシーを表示するには、`get-key-policy` コマンドを使用します。キーポリシーを変更するには、`put-key-policy` コマンドを使用します。

```
aws kms create-key
```

`create-key` コマンドは、キー ID やARN新しいキーを含むKMSキーメタデータを返します。これらの値を使用して、他の AWS KMSオペレーションのKMSキーを識別できます。出力には

タグは含まれません。KMS キーのタグを表示するには、`list-resource-tags` command。

出力:

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2017-07-05T14:04:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_KMS"
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ]
  }
}
```

注: `create-key` コマンドではエイリアスを指定できません。新しいKMSキーのエイリアスを作成するには、`create-alias` コマンドを使用します。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[キーの作成](#)」を参照してください。

例 2: 暗号化と復号用の非対称RSAKMSキーを作成するには

次の`create-key`例では、暗号化と復号用の非対称KMSキーペアを含むRSAキーを作成します。

```
aws kms create-key \
  --key-spec RSA_4096 \
  --key-usage ENCRYPT_DECRYPT
```

出力:

```
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2021-04-05T14:04:55-07:00",
    "CustomerMasterKeySpec": "RSA_4096",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "RSAES_OAEP_SHA_1",
      "RSAES_OAEP_SHA_256"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "RSA_4096",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_KMS"
  }
}
```

詳細については、「Key Management Service デベロッパーガイド」の [「の非対称 AWS KMS キー」](#) を参照してください。AWS

例 3: 署名と検証用の非対称楕円曲線 KMS キーを作成するには

署名と検証用の非対称楕円曲線 (ECC) KMS キーペアを含む非対称キーを作成するには。は ECCKMS キーの唯一の有効な値ですが `SIGN_VERIFY`、`--key-usage` パラメータは必須です。

```
aws kms create-key \
  --key-spec ECC_NIST_P521 \
  --key-usage SIGN_VERIFY
```

出力:

```
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
```

```

    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "ECC_NIST_P521",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "ECC_NIST_P521",
    "KeyState": "Enabled",
    "KeyUsage": "SIGN_VERIFY",
    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "SigningAlgorithms": [
      "ECDSA_SHA_512"
    ]
  }
}

```

詳細については、「[Key AWS Management Service デベロッパーガイド](#)」の「[の非対称 AWS KMS キー](#)」を参照してください。

例 4: HMACKMS キーを作成するには

次の create-key 例では、384 ビット HMACKMS キーを作成します。HMAC KMS キーの有効な GENERATE_VERIFY_MAC 値は パラメータのみですが、--key-usage パラメータの値は必須です。

```

aws kms create-key \
  --key-spec HMAC_384 \
  --key-usage GENERATE_VERIFY_MAC

```

出力:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-04-05T14:04:55-07:00",
    "CustomerMasterKeySpec": "HMAC_384",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",

```

```

    "KeySpec": "HMAC_384",
    "KeyState": "Enabled",
    "KeyUsage": "GENERATE_VERIFY_MAC",
    "MacAlgorithms": [
      "HMAC_SHA_384"
    ],
    "MultiRegion": false,
    "Origin": "AWS_KMS"
  }
}

```

詳細については、[HMAC 「Key Management Service デベロッパーガイド AWS KMSAWS」の「のキー」](#)を参照してください。

例 4: マルチリージョンのプライマリKMSキーを作成するには

次の create-key の例は、マルチリージョンのプライマリ対称暗号化キーを作成します。すべてのパラメータのデフォルト値は対称暗号化キーを作成するため、このKMSキーに必要なのは --multi-region パラメータのみです。で AWS CLI、ブールパラメータが true であることを示すには、パラメータ名を指定します。

```

aws kms create-key \
  --multi-region

```

出力:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2021-09-02T016:15:21-09:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",

```

```

    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": true,
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "PRIMARY",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
        "Region": "us-west-2"
      },
      "ReplicaKeys": []
    },
    "Origin": "AWS_KMS"
  }
}

```

詳細については、「Key Management Service デベロッパーガイド」の「[の非対称 AWS KMS キー](#)」を参照してください。AWS

例 5: インポートされた KMS キーマテリアルのキーを作成するには

次の create-key 例では、が KMS キーマテリアルのないキーを作成します。オペレーションが完了したら、独自のキーマテリアルを KMS キーにインポートできます。この KMS キーを作成するには、--origin パラメータを に設定します EXTERNAL。

```

aws kms create-key \
  --origin EXTERNAL

```

出力:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": false,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",

```

```

    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "PendingImport",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL"
  }
}

```

詳細については、[「Key Management Service デベロッパーガイド」の「キーでの AWS KMS キーマテリアルのインポート」](#)を参照してください。AWS

例 6: AWS CloudHSM KMSキーストアでキーを作成するには

次のcreate-key例では、が指定された AWS CloudHSM KMSキーストアにキーを作成します。オペレーションは、でKMS AWS KMSキーとそのメタデータを作成し、カスタムキーストアに関連付けられた AWS CloudHSM クラスターにキーマテリアルを作成します。--custom-key-store-id および --origin パラメータが必要です。

```

aws kms create-key \
  --origin AWS_CLOUDHSM \
  --custom-key-store-id cks-1234567890abcdef0

```

出力:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
  }
}

```

```
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_CLOUDHSM"
  }
}
```

詳細については、「[Key Management Service デベロッパーガイド](#)」の[AWS 「クラウドHSM キーストア」](#)を参照してください。AWS

例 7: 外部KMSキーストアにキーを作成するには

次のcreate-key例では、が指定された外部KMSキーストアにキーを作成します。このコマンドでは、--custom-key-store-id、--origin、および --xks-key-id パラメータが必須です。

--xks-key-id パラメータは、外部キーマネージャーにある既存の対称暗号化キーの ID を指定します。このキーは、KMSキーの外部キーマテリアルとして機能します。--originパラメータの値は である必要がありますEXTERNAL_KEY_STORE。custom-key-store-idパラメータは、外部キーストアプロキシに接続されている外部キーストアを識別する必要があります。

```
aws kms create-key \
  --origin EXTERNAL_KEY_STORE \
  --custom-key-store-id cks-9876543210fedcba9 \
  --xks-key-id bb8562717f809024
```

出力:

```
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "CustomKeyStoreId": "cks-9876543210fedcba9",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
  }
}
```

```
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「[外部キーストア](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateKey](#)」の「」を参照してください。

decrypt

次の例は、decrypt を使用する方法を説明しています。

AWS CLI

例 1: 対称KMSキーを使用して暗号化されたメッセージを復号するには (Linux および macOS)

次のdecryptコマンド例は、を使用してデータを復号化するための推奨方法を示しています。AWS CLI。このバージョンは、対称KMSキーでデータを復号する方法を示しています。

file.in の暗号文に --ciphertext-blobパラメータの値を指定し、fileb://プレフィックスを使用します。プレフィックスは、にバイナリファイルからデータを読み取るCLIように指示します。ファイルが現在のディレクトリにない場合は、ファイルへのフルパスを入力します。ファイルからパラメータ値を読み取る AWS CLI方法の詳細については、AWS 「コマンドラインインターフェイスユーザーガイド」の「ファイル <<https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html>> からのパラメータのロード AWS CLI」およびAWS 「コマンドラインツールブログ<https://aws.amazon.com/blogs/best-practices-for-local>」の「ローカルファイルパラメータのベストプラクティス」を参照してください。暗号文を復号するKMSキーを指定します。対称KMSキーで復号する場合、--key-idパラメータは必要ありません。AWS KMSは、暗号文のメタデータからデータを暗号化するために使用されたKMSキーのキー ID を取得できます。ただし、使用しているKMSキーを指定するのは常にベストプラクティスです。この練習により、意図したKMSキーを使用し、信頼KMSできないキーを使用して誤って暗号文を復号化することを防ぎます。プレーンテキスト出力をテキスト値としてリクエストします。 --

queryパラメータは、出力からPlaintextフィールドの値のみを取得するCLIように指示します。--output パラメータは出力を text.Base64 でデコードしたプレーンテキストとして返し、ファイルに保存します。次の例では、Plaintext パラメータの値を Base64 ユーティリティにパイプ (|) して、Base64 ユーティリティでデコードします。次に、デコードされた出力を ExamplePlaintext ファイルにリダイレクト (>) します。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。

```
aws kms decrypt \  
  --ciphertext-blob fileb://ExampleEncryptedFile \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query Plaintext | base64 \  
  --decode > ExamplePlaintextFile
```

このコマンドでは何も出力されません。decrypt コマンドからの出力は base64 でデコードされ、ファイルに保存されます。

詳細については、AWS「[キー管理サービスAPIリファレンス](#)」の「[復号化](#)」を参照してください。

例 2: 対称KMSキーを使用して暗号化されたメッセージを復号するには (Windows コマンドプロンプト)

次の例は、certutil ユーティリティを使用してプレーンテキストデータを Base64 でデコードする点を除いて、前の例と同じです。この手順には、次の例に示すように 2 つのコマンドが必要です。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。

```
aws kms decrypt ^  
  --ciphertext-blob fileb://ExampleEncryptedFile ^  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^  
  --output text ^  
  --query Plaintext > ExamplePlaintextFile.base64
```

certutil コマンドを実行します。

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

出力:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

詳細については、AWS「[キー管理サービスAPIリファレンス](#)」の「[復号化](#)」を参照してください。

例 3: 非対称KMSキーを使用して暗号化されたメッセージを復号するには (Linux および macOS)

次のdecryptコマンド例は、RSA非対称KMSキーで暗号化されたデータを復号する方法を示しています。

非対称KMSキーを使用する場合、プレーンテキストの暗号化に使用されるアルゴリズムを指定する encryption-algorithmパラメータが必要です。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 0987dcb-a09fe-87dc-65ba-ab0987654321 \
  --encryption-algorithm RSAES_OAEP_SHA_256 \
  --output text \
  --query Plaintext | base64 \
  --decode > ExamplePlaintextFile
```

このコマンドでは何も出力されません。decrypt コマンドからの出力は base64 でデコードされ、ファイルに保存されます。

詳細については、「[Key Management Service デベロッパーガイド](#)」の「[の非対称 AWS KMSキー](#)」を参照してください。AWS

- API 詳細については、AWS CLI「[コマンドリファレンス](#)」の「[復号化](#)」を参照してください。

delete-alias

次の例は、delete-alias を使用する方法を説明しています。

AWS CLI

エイリアスを削除するには AWS KMS

次の `delete-alias` の例は、エイリアス `alias/example-alias` を削除します。エイリアス名は `alias/` で始まる必要があります。

```
aws kms delete-alias \  
  --alias-name alias/example-alias
```

このコマンドでは何も出力されません。エイリアスを見つけるには、`list-aliases` コマンドを使用します。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[エイリアスの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAlias](#)」の「」を参照してください。

`delete-custom-key-store`

次のコード例は、`delete-custom-key-store` を使用する方法を示しています。

AWS CLI

カスタムキーストアを削除するには

次の `delete-custom-key-store` 例では、指定されたカスタムキーストアを削除します。

AWS CloudHSM キーストアを削除しても、関連付けられた CloudHSM クラスターには影響しません。外部キーストアを削除しても、関連付けられた外部キーストアプロキシ、外部キーマネージャー、または外部キーには影響しません。

NOTE : カスタムキーストアを削除する前に、カスタムキーストア内のすべての KMS キーの削除をスケジュールし、それらの KMS キーの削除を待つ必要があります。次に、カスタムキーストアを切断する必要があります。カスタム KMS キーストア内のキーの検索については、AWS 「[Key Management Service デベロッパーガイド](#)」の [AWS 「クラウド HSM キーストアの削除 \(API\)」](#) を参照してください。

```
delete-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

このコマンドは出力を返しません。カスタムキーストアが削除されていることを確認するには、`describe-custom-key-stores` コマンドを使用します。

AWS CloudHSM キーストアの削除については、「Key Management Service デベロッパーガイド」の[AWS 「CloudHSM キーストアの削除」](#)を参照してください。AWS

外部キーストアの削除については、「Key Management Service デベロッパーガイド」の[「外部キーストアの削除」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCustomKeyStore](#)」の「」を参照してください。

`delete-imported-key-material`

次のコード例は、`delete-imported-key-material` を使用する方法を示しています。

AWS CLI

インポートされたキーマテリアルをKMSキーから削除するには

次の`delete-imported-key-material`例では、キーにインポートされたKMSキーマテリアルを削除します。

```
aws kms delete-imported-key-material \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドでは何も出力されません。キーマテリアルが削除されていることを確認するには、`describe-key` コマンドを使用して `PendingImport` または のキー状態を検索します `PendingDeletion`。

詳細については、AWS 「Key Management Service デベロッパーガイド」の「インポートされたキーマテリアル <<https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-delete-key-material.html>> の削除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteImportedKeyMaterial](#)」の「」を参照してください。

`derive-shared-secret`

次のコード例は、`derive-shared-secret` を使用する方法を示しています。

AWS CLI

共有シークレットを取得するには

次のderive-shared-secret例では、キー契約アルゴリズムを使用して共有シークレットを導出します。

KEY_AGREEMENT を呼び出すには、KeyUsage の値を持つ非対称のNIST推奨楕円曲線 (ECC) または SM2 (中国リージョンのみ) KMSキーペアを使用する必要があります DeriveSharedSecret。

```
aws kms derive-shared-secret \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-agreement-algorithm ECDH \  
  --public-  
key "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvH3Yj0wbkLEpU195Cv1cJVjsVNSjwGq3tCLnzXfhVwV
```

出力:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
  "SharedSecret": "MEYCIQCKZLWyTk5runarx6XiAkU9gv31bwP0/pHa  
+DXFehzdDwIhANwpsIV2g/9SPWLLsF6p/hiSskuIXMTRwqrMdVKWMTMHG",  
  "KeyAgreementAlgorithm": "ECDH",  
  "KeyOrigin": "AWS_KMS"  
}
```

詳細については、「キー管理サービスリファレンス[DeriveSharedSecret](#)」の「」を参照してください。AWS API

- API 詳細については、AWS CLI 「コマンドリファレンス[DeriveSharedSecret](#)」の「」を参照してください。

describe-custom-key-stores

次のコード例は、describe-custom-key-stores を使用する方法を示しています。

AWS CLI

例 1: AWS CloudHSM キーストアの詳細を取得するには

次のdescribe-custom-key-store例では、指定された AWS CloudHSM キーストアの詳細を表示します。コマンドはすべてのタイプのカスタムキーストアで同じですが、出力はキーストアタイプと外部キーストアの接続オプションによって異なります。

デフォルトでは、このコマンドはアカウントとリージョンのすべてのカスタムキーストアに関する情報を表示します。特定のカスタムキーストアに関する情報を表示するには、custom-key-store-name または custom-key-store-idパラメータを使用します。

```
aws kms describe-custom-key-stores \  
  --custom-key-store-name ExampleCloudHSMKeyStore
```

このコマンドの出力には、接続状態 () を含む AWS CloudHSM キーストアに関する便利な詳細が含まれますConnectionState。接続状態が の場合FAILED、出力には問題を説明するConnectionErrorCodeフィールドが含まれます。

出力:

```
{  
  "CustomKeyStores": [  
    {  
      "CloudHsmClusterId": "cluster-1a23b4cdefg",  
      "ConnectionState": "CONNECTED",  
      "CreationDate": "2022-04-05T14:04:55-07:00",  
      "CustomKeyId": "cks-1234567890abcdef0",  
      "CustomKeyName": "ExampleExternalKeyStore",  
      "TrustAnchorCertificate": "<certificate appears here>"  
    }  
  ]  
}
```

詳細については、AWS 「[Key Management Service デベロッパーガイド](#)」の [AWS 「クラウド HSMキーストアの表示](#)」を参照してください。

例 2: パブリックエンドポイント接続を持つ外部キーストアの詳細を取得するには

次のdescribe-custom-key-store例では、指定された外部キーストアの詳細を表示します。コマンドはすべてのタイプのカスタムキーストアで同じですが、出力はキーストアタイプと外部キーストアの接続オプションによって異なります。

デフォルトでは、このコマンドはアカウントとリージョンのすべてのカスタムキーストアに関する情報を表示します。特定のカスタムキーストアに関する情報を表示するには、`custom-key-store-name` または `custom-key-store-id` パラメータを使用します。

```
aws kms describe-custom-key-stores \  
  --custom-key-store-id cks-9876543210fedcba9
```

このコマンドの出力には、接続状態 () など、外部キーストアに関する便利な詳細が含まれます。接続状態が `FAILED` の場合、出力には問題を説明する `ConnectionErrorCode` フィールドが含まれます。

出力:

```
{  
  "CustomKeyStores": [  
    {  
      "CustomKeyId": "cks-9876543210fedcba9",  
      "CustomKeyName": "ExampleXKS",  
      "ConnectionState": "CONNECTED",  
      "CreationDate": "2022-12-02T07:48:55-07:00",  
      "CustomKeyType": "EXTERNAL_KEY_STORE",  
      "XksProxyConfiguration": {  
        "AccessKeyId": "ABCDE12345670EXAMPLE",  
        "Connectivity": "PUBLIC_ENDPOINT",  
        "UriEndpoint": "https://myproxy.xks.example.com",  
        "UriPath": "/example-prefix/kms/xks/v1"  
      }  
    }  
  ]  
}
```

詳細については、「[Key AWS Management Service デベロッパーガイド](#)」の「[外部キーストアの表示](#)」を参照してください。

例 3: VPCエンドポイントサービス接続を持つ外部キーストアの詳細を取得するには

次の `describe-custom-key-store` 例では、指定された外部キーストアの詳細を表示します。コマンドはすべてのタイプのカスタムキーストアで同じですが、出力はキーストアタイプと外部キーストアの接続オプションによって異なります。

デフォルトでは、このコマンドはアカウントとリージョンのすべてのカスタムキーストアに関する情報を表示します。特定のカスタムキーストアに関する情報を表示するには、`custom-key-store-name` または `custom-key-store-id` パラメータを使用します。

```
aws kms describe-custom-key-stores \  
  --custom-key-store-id cks-2234567890abcdef0
```

このコマンドの出力には、接続状態 () など、外部キーストアに関する便利な詳細が含まれます `ConnectionState`。接続状態が `FAILED`、出力には問題を説明する `ConnectionErrorCode` フィールドが含まれます。

出力:

```
{  
  "CustomKeyStores": [  
    {  
      "CustomKeyId": "cks-3234567890abcdef0",  
      "CustomKeyName": "ExampleVPCEExternalKeyStore",  
      "ConnectionState": "CONNECTED",  
      "CreationDate": "2022-12-22T07:48:55-07:00",  
      "CustomKeyType": "EXTERNAL_KEY_STORE",  
      "XksProxyConfiguration": {  
        "AccessKeyId": "ABCDE12345670EXAMPLE",  
        "Connectivity": "VPC_ENDPOINT_SERVICE",  
        "UriEndpoint": "https://myproxy-private.xks.example.com",  
        "UriPath": "/kms/xks/v1",  
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-  
example1"  
      }  
    }  
  ]  
}
```

詳細については、「[Key AWS Management Service デベロッパーガイド](#)」の「[外部キーストアの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス DescribeCustomKeyStores](#)」の「」を参照してください。

describe-key

次のコード例は、`describe-key` を使用する方法を示しています。

AWS CLI

例 1: KMSキーに関する詳細情報を検索するには

次のdescribe-key例では、サンプルアカウントとリージョンの Amazon S3 の AWS マネージドキーに関する詳細情報を取得します。このコマンドを使用して、AWS マネージドキーとカスタマーマネージドキーの詳細を検索できます。

KMS キーを指定するには、key-id/パラメータを使用します。この例ではエイリアス名の値を使用しますが、このコマンドARNでキー ID、キー ARN、エイリアス名、またはエイリアスを使用できます。

```
aws kms describe-key \  
  --key-id alias/aws/s3
```

出力:

```
{  
  "KeyMetadata": {  
    "AWSAccountId": "846764612917",  
    "KeyId": "b8a9477d-836c-491f-857e-07937918959b",  
    "Arn": "arn:aws:kms:us-west-2:846764612917:key/  
b8a9477d-836c-491f-857e-07937918959b",  
    "CreationDate": 2017-06-30T21:44:32.140000+00:00,  
    "Enabled": true,  
    "Description": "Default KMS key that protects my S3 objects when no other  
key is defined",  
    "KeyUsage": "ENCRYPT_DECRYPT",  
    "KeyState": "Enabled",  
    "Origin": "AWS_KMS",  
    "KeyManager": "AWS",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "EncryptionAlgorithms": [  
      "SYMMETRIC_DEFAULT"  
    ]  
  }  
}
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「[キーの表示](#)」を参照してください。

例 2: RSA非対称KMSキーの詳細を取得するには

次のdescribe-key例では、署名と検証に使用される非対称RSAKMSキーに関する詳細情報を取得します。

```
aws kms describe-key \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "KeyMetadata": {  
    "AWSAccountId": "111122223333",  
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
    "Arn": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
    "CreationDate": "2019-12-02T19:47:14.861000+00:00",  
    "CustomerMasterKeySpec": "RSA_2048",  
    "Enabled": false,  
    "Description": "",  
    "KeyState": "Disabled",  
    "Origin": "AWS_KMS",  
    "MultiRegion": false,  
    "KeyManager": "CUSTOMER",  
    "KeySpec": "RSA_2048",  
    "KeyUsage": "SIGN_VERIFY",  
    "SigningAlgorithms": [  
      "RSASSA_PKCS1_V1_5_SHA_256",  
      "RSASSA_PKCS1_V1_5_SHA_384",  
      "RSASSA_PKCS1_V1_5_SHA_512",  
      "RSASSA_PSS_SHA_256",  
      "RSASSA_PSS_SHA_384",  
      "RSASSA_PSS_SHA_512"  
    ]  
  }  
}
```

例 3: マルチリージョンのレプリカキーに関する詳細を取得するには

次の describe-key の例は、マルチリージョンのレプリカキーのメタデータを取得します。このマルチリージョンキーは対称暗号化キーです。マルチリージョンキーの describe-key コマンドの出力は、プライマリキーとそのすべてのレプリカに関する情報を返します。

```
aws kms describe-key \  

```

```
--key-id arn:aws:kms:ap-northeast-1:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab
```

出力:

```
{  
  "KeyMetadata": {  
    "MultiRegion": true,  
    "AWSAccountId": "111122223333",  
    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab",  
    "CreationDate": "2021-06-28T21:09:16.114000+00:00",  
    "Description": "",  
    "Enabled": true,  
    "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",  
    "KeyManager": "CUSTOMER",  
    "KeyState": "Enabled",  
    "KeyUsage": "ENCRYPT_DECRYPT",  
    "Origin": "AWS_KMS",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "EncryptionAlgorithms": [  
      "SYMMETRIC_DEFAULT"  
    ],  
    "MultiRegionConfiguration": {  
      "MultiRegionKeyType": "PRIMARY",  
      "PrimaryKey": {  
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab",  
        "Region": "us-west-2"  
      },  
      "ReplicaKeys": [  
        {  
          "Arn": "arn:aws:kms:eu-west-1:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab",  
          "Region": "eu-west-1"  
        },  
        {  
          "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab",  
          "Region": "ap-northeast-1"  
        },  
        {  
          "Arn": "arn:aws:kms:us-east-1:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab",  
          "Region": "us-east-1"  
        }  
      ]  
    }  
  }  
}
```

```
        "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "sa-east-1"
    }
]
}
}
```

例 4: HMACKMSキーの詳細を取得するには

次のdescribe-key例では、HMACKMSキーに関する詳細情報を取得します。

```
aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{
  "KeyMetadata": {
    "AWSAccountId": "123456789012",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Arn": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2022-04-03T22:23:10.194000+00:00",
    "Enabled": true,
    "Description": "Test key",
    "KeyUsage": "GENERATE_VERIFY_MAC",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "CUSTOMER",
    "CustomerMasterKeySpec": "HMAC_256",
    "MacAlgorithms": [
      "HMAC_SHA_256"
    ],
    "MultiRegion": false
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeKey](#)」の「」を参照してください。

disable-key-rotation

次のコード例は、disable-key-rotation を使用する方法を示しています。

AWS CLI

KMSキーの自動ローテーションを無効にするには

次のdisable-key-rotation例では、カスタマーマネージドKMSキーの自動ローテーションを無効にします。自動ローテーションを再度有効にするには、enable-key-rotation コマンドを使用します。

```
aws kms disable-key-rotation \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドでは何も出力されません。KMS キーの自動ローテーションが無効であることを確認するには、get-key-rotation-status コマンドを使用します。

詳細については、AWS 「Key Management Service デベロッパーガイド」の「[キーのローテーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableKeyRotation](#)」の「」を参照してください。

disable-key

次の例は、disable-key を使用する方法を説明しています。

AWS CLI

KMSキーを一時的に無効にするには

次の例では、disable-key コマンドを使用してカスタマーマネージドKMSキーを無効にします。KMS キーを再度有効にするには、enable-key コマンドを使用します。

```
aws kms disable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドでは何も出力されません。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[キーの有効化と無効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableKey](#)」の「」を参照してください。

disconnect-custom-key-store

次のコード例は、disconnect-custom-key-store を使用する方法を示しています。

AWS CLI

カスタムキーストアを切断するには

次のdisconnect-custom-key-store例では、カスタムキーストアを AWS CloudHSM クラスタから切断します。キーストアを切断して、問題のトラブルシューティング、設定の更新、キーストア内のKMSキーが暗号化オペレーションで使用されないようにすることができます。

このコマンドは、AWS クラウドHSMキーストアや外部キーストアなど、すべてのカスタムキーストアで同じです。

このコマンドを実行する前に、例のカスタムキーストア ID を有効な ID に置き換えます。

```
$ aws kms disconnect-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

このコマンドは出力を生成しません。コマンドが有効であることを確認し、describe-custom-key-stores コマンドを使用します。

AWS CloudHSM キーストアの切断の詳細については、AWS 「Key Management Service デベロッパーガイド」の [AWS 「CloudHSM キーストアの接続と切断](#)」を参照してください。

外部キーストアの切断の詳細については、AWS 「Key Management Service デベロッパーガイド」の「[外部キーストアの接続と切断](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisconnectCustomKeyStore](#)」の「」を参照してください。

enable-key-rotation

次のコード例は、enable-key-rotation を使用する方法を示しています。

AWS CLI

KMSキーの自動ローテーションを有効にするには

次のenable-key-rotation例では、ローテーション期間が 180 日のカスタマーマネージド KMSキーの自動ローテーションを有効にします。KMS キーは、このコマンドが完了した日から 1 年 (約 365 日) 後にローテーションされ、その後は毎年ローテーションされます。

--key-id パラメータはKMSキーを識別します。この例ではキーARN値を使用しますが、キー ID またはKMSキーARNの を使用できます。 --rotation-period-in-daysパラメータは、各ローテーション日の間の日数を指定します。90~2560 日の値を指定します。値が指定されていない場合、デフォルト値は 365 日です。

```
aws kms enable-key-rotation \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
  --rotation-period-in-days 180
```

このコマンドでは何も出力されません。KMS キーが有効になっていることを確認するには、get-key-rotation-status コマンドを使用します。

詳細については、AWS 「Key Management Service デベロッパーガイド」の「[キーのローテーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableKeyRotation](#)」の「」を参照してください。

enable-key

次のコード例は、enable-key を使用する方法を示しています。

AWS CLI

KMSキーを有効にするには

次の enable-key の例は、カスタマーマネージドキーを有効にします。このようなコマンドを使用して、disable-key コマンドを使用して一時的に無効にしたKMSキーを有効にすることができます。また、削除がスケジュールされ、削除がキャンセルされたため、無効にされたKMSキーを有効にするためにも使用できます。

KMS キーを指定するには、key-idパラメータを使用します。この例ではキー ID 値を使用しますが、このコマンドではキー ID またはキーARN値を使用できます。

このコマンドを実行する前に、サンプルキー ID を有効なキー ID に置き換えます。

```
aws kms enable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドでは何も出力されません。KMS キーが有効になっていることを確認するには、describe-key コマンドを使用します。describe-key 出力の KeyState および Enabled フィールドの値を参照してください。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[キーの有効化と無効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableKey](#)」の「」を参照してください。

encrypt

次の例は、encrypt を使用する方法を説明しています。

AWS CLI

例 1: Linux または MacOS でファイルの内容を暗号化するには

次のencryptコマンドは、 でデータを暗号化する推奨方法を示しています AWS CLI。

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  --decode > ExampleEncryptedFile
```

このコマンドは、以下の操作を行います。

--plaintext パラメータを使用して暗号化するデータを指定します。このパラメータ値は base64 でエンコードされている必要があります。plaintext パラメータの値は base64 でエンコードされている必要があります。または、 にファイルからバイナリデータを読み取るように指示 AWS CLIするfileb://プレフィックスを使用する必要があります。ファイルが現在のディレクトリにない場合は、ファイルへのフルパスを入力します。例えば、fileb:///var/tmp/ExamplePlaintextFile、fileb://C:\Temp\ExamplePlaintextFile などです。ファイルからパラメータ値を読み取る AWS CLI方法の詳細については、 コマンドラインイン

ターフェイスユーザーガイドの「[ファイルからのパラメータのロード](#)」と、コマンド AWS ラインツールブログの「[ローカルファイルパラメータのベストプラクティス](#)」を参照してください。--output および --query パラメータを使用してコマンドの出力を制御します。これらのパラメータは、暗号化されたデータを抽出します。AWS 暗号文と呼ばれるコマンドの出力から。出力の制御の詳細については、コマンド AWS ラインインターフェイスユーザーガイドの「[コマンド出力の制御](#)」を参照してください。base64 コーティリティを使用して、抽出された出力をバイナリデータにデコードします。成功した encrypt コマンドによって返される暗号文は base64 でエンコードされたテキストです。を使用して復号する前に、このテキストを AWS CLI デコードする必要があります。バイナリ暗号文をファイルに保存します。コマンド (> ExampleEncryptedFile) の最後の部分は、復号を容易にするためにバイナリ暗号文をファイルに保存しています。を使用して AWS CLI データを復号するコマンドの例については、復号の例を参照してください。

例 2: を使用して Windows でデータを暗号化する AWS CLI

この例は前の例と同じですが、base64 の代わりに certutil ツールを使用します。この手順には、次の例に示すように 2 つのコマンドが必要です。

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob > C:\Temp\ExampleEncryptedFile.base64  
  
certutil -decode C:\Temp\ExampleEncryptedFile.base64 C:\Temp\ExampleEncryptedFile
```

例 3: 非対称 KMS キーによる暗号化

次の encrypt コマンドは、非対称 KMS キーを使用してプレーンテキストを暗号化する方法を示しています。--encryption-algorithm パラメータは必須です。すべての encrypt CLI コマンドと同様に、plaintext パラメータは base64 でエンコードされているか、fileb://プレフィックスを使用する必要があります。プレフィックスは、に AWS CLI ファイルからバイナリデータを読み取るように指示します。

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --encryption-algorithm RSAES_OAEP_SHA_256 \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  \
```

```
--decode > ExampleEncryptedFile
```

このコマンドでは何も出力されません。

- API 詳細については、「コマンドリファレンス」の「[暗号化](#)」を参照してください。AWS CLI

generate-data-key-pair-without-plaintext

次の例は、generate-data-key-pair-without-plaintext を使用する方法を説明しています。

AWS CLI

ECC NIST P384 非対称データキーペアを生成するには

次のgenerate-data-key-pair-without-plaintext例では、ECCNISTP384 キーペアをの外部で使用することをリクエストしています AWS。

コマンドは、プレーンテキストのパブリックキーと、指定されたキーで暗号化されたプライベートKMSキーのコピーを返します。プレーンテキストのプライベートキーは返されません。暗号化されたプライベートキーを暗号化されたデータに安全に保存し、使用する必要がある場合は呼び出し AWS KMSでプライベートキーを復号化できます。

ECC NIST P384 非対称データキーペアをリクエストするには、key-pair-specパラメータの値で使用しますECC_NIST_P384。

指定するKMSキーは、対称暗号化KMSキー、つまり KeySpec の値を持つKMSキーである必要がありますSYMMETRIC_DEFAULT。

NOTE: この例の出力の値は、表示のために切り捨てられます。

```
aws kms generate-data-key-pair-without-plaintext \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-pair-spec ECC_NIST_P384
```

出力:

```
{  
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkkK6Fbh0tQkM1QJJH3PdtHvS/y  
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
```

```
"PublicKey":
  "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+Ld1JE1oUoQV5HpEuHAVbd0yND
+NmYDH/mL10SIEuLrcdZ5hrMH4pk83r401...",
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyPairSpec": "ECC_NIST_P384"
}
```

PublicKey と PrivateKeyCiphertextBlob は base64 でエンコードされた形式で返されま
す。

詳細については、「Key Management Service デベロッパーガイド」の「[データキーペア](#)」を参
照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレン
ス[GenerateDataKeyPairWithoutPlaintext](#)」の「」を参照してください。

generate-data-key-pair

次のコード例は、generate-data-key-pair を使用する方法を示しています。

AWS CLI

2048 ビットのRSA非対称データキーペアを生成するには

次のgenerate-data-key-pair例では、2048 ビットのRSA非対称データキーペアを の外部で
使用することをリクエストしています AWS。コマンドは、即時使用および削除のためにプレー
ンテキストのパブリックキーとプレーンテキストのプライベートキー、および指定されたキーで
暗号化されたプライベートKMSキーのコピーを返します。暗号化されたプライベートキーは、暗
号化されたデータと共に安全に保存できます。

2048 ビットのRSA非対称データキーペアをリクエストするには、key-pair-specパラメータ
を の値で使用しますRSA_2048。

指定するKMSキーは、対称暗号化KMSキー、つまり KeySpec の値を持つKMSキーである必要が
ありますSYMMETRIC_DEFAULT。

NOTE: この例の出力の値は、表示のために切り捨てられます。

```
aws kms generate-data-key-pair \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-pair-spec RSA_2048
```

出力:

```
{
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkk6Fbh0tQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
  "PrivateKeyPlaintext": "MIIG/
QIBADANBgkqhkiG9w0BAQEFAASCBUcwggbjAgEAAoIBgQDcDd4YzI
+u9Kfv4t2UkTWhShBXkekS4cBVt07I0P4ZgMf+YvU5IgS4ut...",
  "PublicKey":
  "MIIB0jANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+LdlJE1oUoQV5HpEuHAVbd0yND
+NmYDH/mL10SIEuLrzdZ5hrMH4pk83r401...",
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyPairSpec": "RSA_2048"
}
```

PublicKey、PrivateKeyPlaintext、PrivateKeyCiphertextBlobは base64 エンコード形式で返されます。

詳細については、「Key Management Service デベロッパーガイド」の「[データキーペア](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateDataKeyPair](#)」の「」を参照してください。

generate-data-key-without-plaintext

次の例は、generate-data-key-without-plaintext を使用する方法を説明しています。

AWS CLI

プレーンテキストキーを使用せずに 256 ビットの対称データキーを生成するには

次の generate-data-key-without-plaintext の例は、256 ビットの対称データキーの暗号化コピーを AWS の外部で使用するようリクエストします。データキーを使用する準備ができたら、`aws kms generate-data-key-without-plaintext` を呼び出し AWS KMS でデータキーを復号できます。

256 ビットのデータキーをリクエストするには、値 `AES_256` を含む `key-spec` パラメータを使用します。128 ビットのデータキーをリクエストするには、値 `AES_128` を含む `key-spec` パラメータを使用します。その他のすべてのデータキー長には、`number-of-bytes` パラメータを使用します。

指定するKMSキーは、対称暗号化KMSキー、つまりKMSキー仕様値が `SYMMETRIC_` のキーである必要がありますDEFAULT。

```
aws kms generate-data-key-without-plaintext \  
  --key-id "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --key-spec AES_256
```

出力:

```
{  
  "CiphertextBlob":  
  "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZiHvcNAQcGoG8wbQIBADBoBgkqhkiG0C..."  
  "KeyId": "arn:aws:kms:us-  
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

CiphertextBlob (暗号化されたデータキー) は base64 でエンコードされた形式で返されます。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[データキー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateDataKeyWithoutPlaintext](#)」の「」を参照してください。

generate-data-key

次の例は、generate-data-key を使用する方法を説明しています。

AWS CLI

例 1: 256 ビットの対称データキーを生成するには

次のgenerate-data-key例では、外で使用するために 256 ビット対称データキーをリクエストします AWS。コマンドは、すぐに使用および削除できるようにプレーンテキストデータキーと、指定されたキーで暗号化されたそのデータKMSキーのコピーを返します。暗号化されたデータとともに、暗号化されたデータキーを安全に保存できます。

256 ビットのデータキーをリクエストするには、値 `AES_256` を含む `key-spec` パラメータを使用します。128 ビットのデータキーをリクエストするには、値 `AES_128` を含む `key-spec` パラ

メータを使用します。その他のすべてのデータキー長には、`number-of-bytes` パラメータを使用します。

指定するKMSキーは、対称暗号化KMSキー、つまりKMSキー仕様値が `SYMMETRIC_` のキーである必要がありますDEFAULT。

```
aws kms generate-data-key \  
  --key-id alias/ExampleAlias \  
  --key-spec AES_256
```

出力:

```
{  
  "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",  
  "KeyId": "arn:aws:kms:us-  
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "CiphertextBlob":  
  "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIHvcNAQcGoG8wbQIBADBoBgkqhki+YdhV8MrkBQPeac0ReRVNDt9qlAt+SHgIRF8P0H+7U=" }  
}
```

Plaintext (プレーンテキストデータキー) と CiphertextBlob (暗号化データキー) は base64 でエンコードされた形式で返されます。

詳細については、AWS 「Key Management Service デベロッパーガイド」の「データキー」<<https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys>> を参照してください。

例 2: 512 ビットの対称データキーを生成するには

次の `generate-data-key` の例は、暗号化と復号用の 512 ビットの対称データキーを要求します。コマンドは、すぐに使用および削除できるようにプレーンテキストデータキーと、指定されたキーで暗号化されたそのデータKMSキーのコピーを返します。暗号化されたデータとともに、暗号化されたデータキーを安全に保存できます。

128 ビットまたは 256 ビット以外のキー長をリクエストするには、`number-of-bytes` パラメータを使用します。512 ビットのデータキーをリクエストするために、次の例では値が 64 (バイト) の `number-of-bytes` パラメータを使用します。

指定するKMSキーは、対称暗号化KMSキー、つまりKMSキー仕様値が `SYMMETRIC_` のキーである必要がありますDEFAULT。

NOTE: この例の出力の値は、表示のために切り捨てられます。

```
aws kms generate-data-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --number-of-bytes 64
```

出力:

```
{  
  "CiphertextBlob": "AQIBAHi6LtupRpdK12aJTzkk6Fbh0tQkMlQJJH3PdtHvS/y+hAEnX/  
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSiB3DQEHBqCCA1UwggJRAgEAMIICSgYJKoZ...",  
  "Plaintext": "ty8Lr0Bk60F07M2Bwt6qbFdNB  
+G00ZLtf5MSEb4a13R2UKWGOp06njAwy2n72VRm2m7z/  
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm40ovEXiVfBveyX3DQxDzRSwbKDPk/...",  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

Plaintext (プレーンテキストデータキー) と CiphertextBlob (暗号化データキー) は base64 でエンコードされた形式で返されます。

詳細については、AWS 「Key Management Service デベロッパーガイド」の「データキー <<https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys>>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GenerateDataKey](#)」の「」を参照してください。

generate-random

次のコード例は、generate-random を使用する方法を示しています。

AWS CLI

例 1: 256 ビットのランダムバイト文字列を生成するには (Linux または macOS)

次の generate-random の例は、256 ビット (32 バイト) の base64 でエンコードされたランダムバイト文字列を生成します。この例は、バイト文字列をデコードしてランダムファイルに保存します。

このコマンドを実行する際は、number-of-bytes パラメーターを使用してランダム値の長さをバイト単位で指定する必要があります。

このコマンドを実行するときにKMSキーを指定しません。ランダムバイト文字列は、どのKMSキーとも関係ありません。

デフォルトでは、AWS KMS は乱数を生成します。ただし、カスタムキーストア <<https://docs.aws.amazon.com/kms/latest/developerguide/custom-key-store-overview.html>>を指定すると、ランダムなバイト文字列がカスタムキーストアに関連付けられた AWS CloudHSM クラスターで生成されます。

この例は以下のパラメータと値を使用します。

値の必須 `--number-of-bytes` パラメータを使用して 32 バイト (256 ビット) 32 をリクエストします。string.it は の値の `--output` パラメータを使用して、text AWS CLI に出力をテキストとして返すように指示します。の代わりにJSON、を使用して response.it パイプ (|) から Plaintext プロパティの値を `--query parameter` 抽出し、コマンドの出力を base64 ユーティリティ、は、抽出された output.it をデコードし、リダイレクト演算子 (>) を使用してデコードされたバイト文字列を ExampleRandom file.it に保存し、リダイレクト演算子 (>) を使用してバイナリ暗号文をファイルに保存します。

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext | base64 --decode > ExampleRandom
```

このコマンドでは何も出力されません。

詳細については、「キー管理サービスリファレンス [GenerateRandom](#)」の「」を参照してください。AWS API

例 2: 256 ビットの乱数を生成するには (Windows コマンドプロンプト)

次の例は、generate-random コマンドを使用して、base64 でエンコードされた 256 ビット (32 バイト) のランダムバイト文字列を生成します。この例は、バイト文字列をデコードしてランダムファイルに保存します。この例は前の例と同じですが、Windows の certutil のユーティリティを使用して、ランダムバイト文字列を base64 でデコードしてからファイルに保存する点が異なります。

最初に base64 でエンコードされたランダムバイト文字列を生成し、テンポラリファイル (ExampleRandom.base64) に保存します。

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text | certutil -decode - > ExampleRandom.base64
```



```
--output text \  
--query Plaintext > ExampleRandom.base64
```

generate-random コマンドの出力はファイルに保存されるため、この例では何も出力されません。

次に、certutil -decode コマンドを使用して、ExampleRandom.base64 ファイル内の base64 でエンコードされたバイト文字列をデコードします。その後、デコードされたバイト文字列を ExampleRandom ファイルに保存します。

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

出力:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

詳細については、「キー管理サービスリファレンス[GenerateRandom](#)」の「」を参照してください。AWS API

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateRandom](#)」の「」を参照してください。

get-key-policy

次の例は、get-key-policy を使用する方法を説明しています。

AWS CLI

キーポリシーを 1 つのキーから別の KMS キーにコピーするには KMS

次の get-key-policy 例では、1 つのキーから KMS キーポリシーを取得し、テキストファイルに保存します。次に、テキストファイルをポリシー入力として使用して、別の KMS キーのポリシーを置き換えます。

の --policy パラメータには文字列 put-key-policy が必要なため、--output text オプションを使用して、出力を の代わりにテキスト文字列として返す必要があります JSON。

```
aws kms get-key-policy \  
--policy-name default \  
--output text
```

```
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
--query Policy \  
--output text > policy.txt  
  
aws kms put-key-policy \  
--policy-name default \  
--key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
--policy file://policy.txt
```

このコマンドでは何も出力されません。

詳細については、「リファレンス[PutKeyPolicy](#)」の「」を参照してください。AWS KMS API
• API 詳細については、AWS CLI 「コマンドリファレンス[GetKeyPolicy](#)」の「」を参照してください。

get-key-rotation-status

次の例は、get-key-rotation-status を使用する方法を説明しています。

AWS CLI

KMSキーのローテーションステータスを取得するには。

次のget-key-rotation-status例では、自動ローテーションが有効になっているかどうか、ローテーション期間、次に予定されているローテーション日など、指定されたKMSキーのローテーションステータスに関する情報を返します。このコマンドは、カスターマネージドKMSキーとAWS マネージドKMSキーで使用できます。ただし、すべてのAWS マネージドKMSキーは毎年自動的にローテーションされます。

```
aws kms get-key-rotation-status \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeyRotationEnabled": true,  
  "NextRotationDate": "2024-02-14T18:14:33.587000+00:00",  
  "RotationPeriodInDays": 365  
}
```

詳細については、AWS 「[Key Management Service デベロッパーガイド](#)」の「[キーのローテーション](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetKeyRotationStatus](#)」の「」を参照してください。

get-parameters-for-import

次の例は、get-parameters-for-import を使用する方法を説明しています。

AWS CLI

キーマテリアルをKMSキーにインポートするために必要な項目を取得するには

次のget-parameters-for-import例では、キーマテリアルをキーにインポートするために必要なパブリックKMSキーとインポートトークンを取得します。import-key-material コマンドを使用する場合は、同じget-parameters-for-importコマンドで返されたパブリックキーによって暗号化されたインポートトークンとキーマテリアルを必ず使用してください。また、このコマンドで指定するラッピングアルゴリズムは、パブリックキーでキーマテリアルを暗号化するために使用するアルゴリズムである必要があります。

KMS キーを指定するには、key-idパラメータを使用します。この例ではキー ID を使用しますが、このコマンドARNではキー ID またはキーを使用できます。

```
aws kms get-parameters-for-import \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --wrapping-algorithm RSAES_OAEP_SHA_256 \  
  --wrapping-key-spec RSA_2048
```

出力:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "PublicKey": "<public key base64 encoded data>",  
  "ImportToken": "<import token base64 encoded data>",  
  "ParametersValidTo": 1593893322.32  
}
```

詳細については、「[Key Management Service デベロッパーガイド](#)」の「[パブリックキーをダウンロードしてトークンをインポートする](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParametersForImport](#)」の「」を参照してください。

get-public-key

次の例は、get-public-key を使用する方法を説明しています。

AWS CLI

例 1: 非対称キーのパブリックKMSキーをダウンロードするには

次のget-public-key例では、非対称キーのパブリックKMSキーをダウンロードします。

パブリックキーを返すだけでなく、出力には、キーの使用状況やサポートされている暗号化アルゴリズムなど AWS KMS、 の外部でパブリックキーを安全に使用するために必要な情報が含まれます。

```
aws kms get-public-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "PublicKey": "jANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA15epvg1/  
QtJhXSi2g9SDEVg8QV/...",  
  "CustomerMasterKeySpec": "RSA_4096",  
  "KeyUsage": "ENCRYPT_DECRYPT",  
  "EncryptionAlgorithms": [  
    "RSAES_OAEP_SHA_1",  
    "RSAES_OAEP_SHA_256"  
  ]  
}
```

で非対称KMSキーを使用する方法の詳細については AWS KMS、AWS 「キー管理サービスAPIリファレンス」の「[対称キーと非対称キーの使用](#)」を参照してください。

例 2: パブリックキーをDER形式に変換するには (Linux および macOS)

次のget-public-key例では、非対称キーのパブリックKMSキーをダウンロードし、DERファイルに保存します。

で `get-public-key` コマンドを使用すると AWS CLI、Base64 DER でエンコードされた というエンコードされた X.509 パブリックキーが返されます。Base64-encoded この例では、`PublicKey` プロパティの値をテキストとして取得します。Base64-decodes `PublicKey` し、`public_key.der` ファイルに保存します。output パラメータは、出力をではなくテキストとして返します JSON。--query パラメータは `PublicKey` プロパティのみを取得し、 の外部でパブリックキーを安全に使用するために必要なプロパティは取得しません AWS KMS。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。

```
aws kms get-public-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query PublicKey | base64 --decode > public_key.der
```

このコマンドでは何も出力されません。

での非対称 KMS キーの使用の詳細については AWS KMS、AWS キー管理サービス API リファレンスの「[対称キーと非対称キーの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetPublicKey](#)」の「」を参照してください。

import-key-material

次の例は、`import-key-material` を使用する方法を説明しています。

AWS CLI

キーマテリアルを KMS キーにインポートするには

次の `import-key-material` 例では、キーマテリアルなしで作成された KMS キーにキーマテリアルをアップロードします。キーの KMS キー状態は である必要があります `PendingImport`。

このコマンドは、`get-parameters-for-import` コマンドが返したパブリックキーで暗号化したキーマテリアルを使用します。また、同じ `get-parameters-for-import` コマンドからのインポートトークンも使用します。

`expiration-model` パラメータは、キーマテリアルが `valid-to` パラメータで指定された日付と時刻に自動的に期限切れになることを示します。キーマテリアルの有効期限が切れる

と、AWS KMS はキーマテリアルを削除Pending importし、KMSキーのキー状態は に変わり、KMSキーは使用できなくなります。KMS キーを復元するには、同じキーマテリアルを再インポートする必要があります。異なるキーマテリアルを使用するには、新しいKMSキーを作成する必要があります。

このコマンドを実行する前に、サンプルキー ID をアカウントの有効なキー ID またはキーに置き換えARNます AWS 。

```
aws kms import-key-material \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --encrypted-key-material fileb://EncryptedKeyMaterial.bin \  
  --import-token fileb://ImportToken.bin \  
  --expiration-model KEY_MATERIAL_EXPIRES \  
  --valid-to 2021-09-21T19:00:00Z
```

このコマンドでは何も出力されません。

キーマテリアルのインポートの詳細については、[「キーマネジメントサービスデベロッパーガイド」の「キーマテリアルのインポート」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportKeyMaterial](#)」の「」を参照してください。

list-aliases

次の例は、list-aliases を使用する方法を説明しています。

AWS CLI

例 1: AWS アカウントとリージョン内のすべてのエイリアスを一覧表示するには

次の例では、list-aliases コマンドを使用して、AWS アカウントのデフォルトのリージョン内のすべてのエイリアスを一覧表示します。出力には、AWS マネージドKMSキーとカスタマーマネージドKMSキーに関連付けられたエイリアスが含まれます。

```
aws kms list-aliases
```

出力:

```
{  
  "Aliases": [  
    {
```

```

    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/testKey",
    "AliasName": "alias/testKey",
    "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  {
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/FinanceDept",
    "AliasName": "alias/FinanceDept",
    "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
  },
  {
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
    "AliasName": "alias/aws/dynamodb",
    "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
  },
  {
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
    "AliasName": "alias/aws/ebs",
    "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef"
  },
  ...
]
}

```

例 2: 特定のKMSキーのすべてのエイリアスを一覧表示するには

次の例では、`list-aliases` コマンドとその `key-id` パラメータを使用して、特定のKMSキーに関連付けられているすべてのエイリアスを一覧表示します。

各エイリアスは1つのKMSキーにのみ関連付けられますが、KMSキーには複数のエイリアスを含めることができます。このコマンドは、コンソールがAWS KMS各KMSキーに1つのエイリアスのみを一覧表示するため、非常に便利です。KMS キーのすべてのエイリアスを検索するには、`list-aliases` コマンドを使用する必要があります。

この例では、`--key-id` パラメータに KMS キーのキー ID を使用しますが、このコマンドARNでキー ID、キー ARN、エイリアス名、またはエイリアスを使用できます。

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{
  "Aliases": [
```

```
{
  "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/oregon-test-key",
  "AliasName": "alias/oregon-test-key"
},
{
  "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project121-test",
  "AliasName": "alias/project121-test"
}
]
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「[エイリアスの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAliases](#)」の「」を参照してください。

list-grants

次のコード例は、list-grants を使用する方法を示しています。

AWS CLI

キーの AWS KMS 許可を表示するには

次の list-grants 例では、アカウントの Amazon DynamoDB の指定された AWS マネージド KMS キーに対するすべての許可を表示します。この許可により、DynamoDB はユーザーに代わって KMS キーを使用して DynamoDB テーブルを暗号化してからディスクに書き込むことができます。このようなコマンドを使用して、AWS アカウントとリージョンの AWS マネージド KMS キーとカスタマー マネージド KMS キーの権限を表示できます。

このコマンドは、キー ID を持つ key-id パラメータを使用して KMS キーを識別します。キー ID またはキーを使用して KMS キー ARN を識別できます。AWS マネージド キー ARN のキー ID または KMS キーを取得するには、list-keys または list-aliases コマンドを使用します。

```
aws kms list-grants \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力は、グラントが暗号化オペレーションに KMS キーを使用するアクセス許可を Amazon DynamoDB に付与し、KMS キーの詳細 (DescribeKey) を表示し、グラントを廃止するアクセス

許可 () を付与することを示しています `RetireGrant`。 `EncryptionContextSubset` の制約により、これらの権限は指定された暗号化コンテキストペアを含むリクエストに限定されます。そのため、この権限のアクセス許可は指定されたアカウントと DynamoDB テーブルでのみ有効です。

```
{
  "Grants": [
    {
      "Constraints": {
        "EncryptionContextSubset": {
          "aws:dynamodb:subscriberId": "123456789012",
          "aws:dynamodb:tableName": "Services"
        }
      },
      "IssuingAccount": "arn:aws:iam::123456789012:root",
      "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
      "Operations": [
        "Decrypt",
        "Encrypt",
        "GenerateDataKey",
        "ReEncryptFrom",
        "ReEncryptTo",
        "RetireGrant",
        "DescribeKey"
      ],
      "GrantId":
        "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59",
      "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
      "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
      "CreationDate": "2021-05-13T18:32:45.144000+00:00"
    }
  ]
}
```

詳細については、AWS 「[Key Management Service デベロッパーガイド](#)」の「[のグラント AWS KMS](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス `ListGrants`](#)」の「」を参照してください。

list-key-policies

次の例は、list-key-policies を使用する方法を説明しています。

AWS CLI

キーのKMSキーポリシーの名前を取得するには

次の list-key-policies の例は、サンプルアカウントとリージョンのカスタマーマネージドキーのキーポリシーの名前を取得します。このコマンドを使用して、AWS マネージドキーとカスタマーマネージドキーのキーポリシーの名前を検索できます。

有効なキーポリシー名は default のみであるため、このコマンドは有用ではありません。

KMS キーを指定するには、key-idパラメータを使用します。この例ではキー ID 値を使用しますが、このコマンドARNではキー ID またはキーを使用できます。

```
aws kms list-key-policies \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "PolicyNames": [  
    "default"  
  ]  
}
```

キーポリシーの詳細については AWS KMS、[「Key Management Service デベロッパーガイド」](#)の「[でのキーポリシーの使用 AWS KMS](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスListKeyPolicies](#)」の「」を参照してください。

list-key-rotations

次の例は、list-key-rotations を使用する方法を説明しています。

AWS CLI

完了したすべてのキーマテリアルローテーションに関する情報を取得するには

次のlist-key-rotations例は、指定されたキーに対して完了したすべてのKMSキーマテリアルローテーションに関する情報を示しています。

```
aws kms list-key-rotations \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "Rotations": [  
    {  
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
      "RotationDate": "2024-03-02T10:11:36.564000+00:00",  
      "RotationType": "AUTOMATIC"  
    },  
    {  
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
      "RotationDate": "2024-04-05T15:14:47.757000+00:00",  
      "RotationType": "ON_DEMAND"  
    }  
  ],  
  "Truncated": false  
}
```

詳細については、AWS 「Key Management Service デベロッパーガイド」の [「キーのローテーション」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListKeyRotations](#)」の「」を参照してください。

list-keys

次のコード例は、list-keys を使用する方法を示しています。

AWS CLI

アカウントとリージョンのKMSキーを取得するには

次のlist-keys例では、アカウントとリージョンのKMSキーを取得します。このコマンドは、AWS マネージドキーとカスタマーマネージドキーの両方を返します。

aws kms list-keys

出力:

```
{
  "Keys": [
    {
      "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
    },
    {
      "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
      "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    }
  ]
}
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「[キーの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListKeys](#)」の「」を参照してください。

list-resource-tags

次の例は、list-resource-tags を使用する方法を説明しています。

AWS CLI

KMSキーのタグを取得するには

次のlist-resource-tags例では、KMSキーのタグを取得します。KMS キーにリソースタグを追加または置き換えるには、tag-resource コマンドを使用します。出力は、このKMSキーに2つのリソースタグがあり、それぞれにキーと値があることを示しています。

KMS キーを指定するには、`key-id`パラメータを使用します。この例ではキー ID 値を使用しますが、このコマンドARNではキー ID またはキーを使用できます。

```
aws kms list-resource-tags \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "Tags": [  
    {  
      "TagKey": "Dept",  
      "TagValue": "IT"  
    },  
    {  
      "TagKey": "Purpose",  
      "TagValue": "Test"  
    }  
  ],  
  "Truncated": false  
}
```

でのタグの使用の詳細については AWS KMS、AWS 「Key Management Service デベロッパーガイド」の「キーの[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceTags](#)」の「」を参照してください。

list-retirable-grants

次の例は、`list-retirable-grants` を使用する方法を説明しています。

AWS CLI

プリンシパルが廃止できる許可を表示するには

次の`list-retirable-grants`例では、`ExampleAdmin`ユーザーが AWS アカウントとリージョンのKMSキーで廃止できるすべての許可を表示します。このようなコマンドを使用して、アカウントプリンシパルが AWS アカウントとリージョンのKMSキーに対して廃止できる権限を表示できます。

必須 `retiring-principal` パラメータの値は、アカウント、ユーザー、またはロールの Amazon リソースネーム (ARN) である必要があります。

このコマンド `retiring-principal` で の値にサービスを指定することはできません。ただし、サービスが廃止プリンシパルである可能性があります。特定のサービスが廃止プリンシパルである許可を見つけるには、`list-grants` コマンドを使用します。

出力は、`ExampleAdmin` ユーザーがアカウントとリージョンの 2 つの異なる KMS キーに対する許可を廃止するアクセス許可を持っていることを示しています。アカウントには、廃止プリンシパルに加えて、アカウント内の許可を廃止するアクセス許可があります。

```
aws kms list-retirable-grants \  
--retiring-principal arn:aws:iam::111122223333:user/ExampleAdmin
```

出力:

```
{  
  "Grants": [  
    {  
      "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
      "GrantId":  
"156b69c63cb154aa21f59929fff19760717be8d9d82b99df53e18b94a15a5e88e",  
      "Name": "",  
      "CreationDate": 2021-01-14T20:17:36.419000+00:00,  
      "GranteePrincipal": "arn:aws:iam::111122223333:user/ExampleUser",  
      "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",  
      "IssuingAccount": "arn:aws:iam::111122223333:root",  
      "Operations": [  
        "Encrypt"  
      ],  
      "Constraints": {  
        "EncryptionContextSubset": {  
          "Department": "IT"  
        }  
      }  
    },  
    {  
      "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",  
      "GrantId":  
"8c94d1f12f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
```

```
    "Name": "",
    "CreationDate": "2021-02-02T19:49:49.638000+00:00",
    "GranteePrincipal": "arn:aws:iam::111122223333:role/ExampleRole",
    "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
    "IssuingAccount": "arn:aws:iam::111122223333:root",
    "Operations": [
      "Decrypt"
    ],
    "Constraints": {
      "EncryptionContextSubset": {
        "Department": "IT"
      }
    }
  },
  "Truncated": false
}
```

詳細については、AWS 「Key Management Service デベロッパーガイド」の「[のグラント AWS KMS](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRetirableGrants](#)」の「」を参照してください。

put-key-policy

次のコード例は、put-key-policy を使用する方法を示しています。

AWS CLI

キーのKMSキーポリシーを変更するには

次の put-key-policy の例は、カスタマーマネージドキーのキーポリシーを変更します。

開始するには、キーポリシーを作成し、ローカルJSONファイルに保存します。この例では、key_policy.json がこのファイルです。また、キーポリシーを policy パラメータの文字列値として指定することもできます。

このキーポリシーの最初のステートメントは、AWS アカウントに対し、IAM ポリシーを使用してKMSキーへのアクセスを制御するアクセス許可を付与します。2 番目のステートメントは、KMSキーで describe-key および list-keys コマンドを実行するアクセス許可を test-user ユーザーに付与します。

key_policy.json の内容:

```
{
  "Version" : "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [
    {
      "Sid" : "Enable IAM User Permissions",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : "kms:*",
      "Resource" : "*"
    },
    {
      "Sid" : "Allow Use of Key",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:user/test-user"
      },
      "Action" : [
        "kms:DescribeKey",
        "kms:ListKeys"
      ],
      "Resource" : "*"
    }
  ]
}
```

KMS キーを識別するには、この例ではキー ID を使用しますが、キーを使用することもできます ARN。キーポリシーを指定するために、このコマンドは `policy` パラメータを使用します。ポリシーがファイル内にあることを示すには、必要な `file://` プレフィックスを使用します。このプレフィックスは、サポートされているすべてのオペレーティングシステムでファイルを識別するために必要です。最後に、コマンドは値が `default` の `policy-name` パラメータを使用します。ポリシー名が指定されていない場合、デフォルト値は `default` です。唯一の有効な値は `default` です。

```
aws kms put-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
```



```
--policy file://key_policy.json
```

このコマンドでは、出力が生成されません。コマンドが有効であることを確認するには、`get-key-policy` コマンドを使用します。次のコマンド例では、同じキーのKMSキーポリシーを取得します。値が `text` の `output` パラメータは、読みやすいテキスト形式を返します。

```
aws kms get-key-policy \  
  --policy-name default \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text
```

出力:

```
{  
  "Version" : "2012-10-17",  
  "Id" : "key-default-1",  
  "Statement" : [  
    {  
      "Sid" : "Enable IAM User Permissions",  
      "Effect" : "Allow",  
      "Principal" : {  
        "AWS" : "arn:aws:iam::111122223333:root"  
      },  
      "Action" : "kms:*",  
      "Resource" : "*" ,  
    },  
    {  
      "Sid" : "Allow Use of Key",  
      "Effect" : "Allow",  
      "Principal" : {  
        "AWS" : "arn:aws:iam::111122223333:user/test-user"  
      },  
      "Action" : [ "kms:Describe", "kms:List" ],  
      "Resource" : "*" ,  
    }  
  ]  
}
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「[キーポリシーの変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutKeyPolicy](#)」の「」を参照してください。

re-encrypt

次のコード例は、re-encrypt を使用する方法を示しています。

AWS CLI

例 1: 暗号化されたメッセージを別の対称KMSキー (Linux および macOS) で再暗号化するには。

次のre-encryptコマンド例は、 でデータを再暗号化する推奨方法を示しています AWS CLI。

file.in の暗号文に `--ciphertext-blob` パラメータの値を指定し、`fileb://`プレフィックスを使用します。プレフィックスは、 にバイナリファイルからデータを読み取るCLIのように指示します。ファイルが現在のディレクトリにない場合は、ファイルへのフルパスを入力します。ファイルからパラメータ値を読み取る AWS CLI方法の詳細については、AWS 「コマンドラインインターフェイスユーザーガイド」の「ファイル <<https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html>> からのパラメータのロード AWS CLI」とAWS 「コマンドラインツールブログ <https://aws.amazon.com/blogs/best-practices-for-local>」の「ローカルファイルパラメータのベストプラクティス」を参照してください。暗号文を復号するソースKMSキーを指定します。対称暗号化KMSキーで復号する場合、 `--source-key-id` パラメータは必要ありません。AWS KMS は、暗号文 BLOB のメタデータからデータを暗号化するために使用されたKMSキーを取得できます。ただし、使用しているKMSキーを指定するのは常にベストプラクティスです。この練習により、意図したKMSキーを使用し、信頼KMSできないキーを使用して誤って暗号文を復号化することを防ぐことができます。送信先KMSキーを指定して、データを再暗号化します。 `--destination-key-id` パラメータは常に必要です。この例ではキーを使用しますがARN、任意の有効なキー識別子を使用できます。プレーンテキスト出力をテキスト値としてリクエストします。 `--query` パラメータは に、出力からPlaintextフィールドの値のみを取得するCLIのように指示します。 `--output` パラメータは出力を `text.Base64` でデコードしたプレーンテキストとして返し、ファイルに保存します。次の例では、Plaintext パラメータの値を Base64 ユーティリティにパイプ (`|`) して、Base64 ユーティリティでデコードします。次に、デコードされた出力を `ExamplePlaintext` ファイルにリダイレクト (`>`) します。

このコマンドを実行する前に、サンプルキーを AWS アカウントのIDs有効なキー識別子に置き換えます。

```
aws kms re-encrypt \
```

```
--ciphertext-blob fileb://ExampleEncryptedFile \  
--source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
--destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
--query CiphertextBlob \  
--output text | base64 --decode > ExampleReEncryptedFile
```

このコマンドでは何も出力されません。re-encrypt コマンドからの出力は base64 でデコードされ、ファイルに保存されます。

詳細については、「キー管理サービスリファレンス」の ReEncrypt 「< https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html」を参照してください。

AWS API

例 2: 暗号化されたメッセージを別の対称KMSキーで再暗号化するには (Windows コマンドプロンプト)。

次の re-encrypt の例は、certutil ユーティリティを使用してプレーンテキストデータを Base64 でデコードする点を除いて、前の例と同じです。この手順には、次の例に示すように 2 つのコマンドが必要です。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。

```
aws kms re-encrypt ^  
--ciphertext-blob fileb://ExampleEncryptedFile ^  
--source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^  
--destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 ^  
--query CiphertextBlob ^  
--output text > ExampleReEncryptedFile.base64
```

その後、certutil ユーティリティを使用します。

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

出力:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

詳細については、「Key Management Service リファレンス」の ReEncrypt 「< https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html」を参照してください。

AWS API

- API 詳細については、AWS CLI 「コマンドリファレンス[ReEncrypt](#)」の「」を参照してください。

retire-grant

次の例は、retire-grant を使用する方法を説明しています。

AWS CLI

カスタマーマスターキーの権限を廃止するには

次のretire-grant例では、KMSキーから許可を削除します。

次のコマンド例は、grant-id パラメータと key-id パラメータを指定します。key-id パラメータの値は、キーARNの KMS キーである必要があります。

```
aws kms retire-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドでは何も出力されません。権限が廃止されたことを確認するには、list-grants コマンドを使用します。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[グラントの使用停止と取り消し](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RetireGrant](#)」の「」を参照してください。

revoke-grant

次のコード例は、revoke-grant を使用する方法を示しています。

AWS CLI

カスタマーマスターキーの権限を取り消すには

次のrevoke-grant例では、KMSキーから許可を削除します。次のコマンド例は、grant-idパラメータとkey-idパラメータを指定します。key-idパラメータの値は、キーARNのキーIDまたはKMSキーにすることができます。

```
aws kms revoke-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドでは何も出力されません。権限が取り消されたことを確認するには、list-grants コマンドを使用します。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[グラントの使用停止と取り消し](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RevokeGrant](#)」の「」を参照してください。

rotate-key-on-demand

次の例は、rotate-key-on-demand を使用する方法を説明しています。

AWS CLI

KMSキーのオンデマンドローテーションを実行するには

次のrotate-key-on-demand例では、指定したキーのKMSキーマテリアルのローテーションをすぐに開始します。

```
aws kms rotate-key-on-demand \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

出力:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

詳細については、「Key Management Service デベロッパーガイド」の「[オンデマンドキーローテーションの実行方法](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [RotateKeyOnDemand](#)」の「」を参照してください。

schedule-key-deletion

次のコード例は、schedule-key-deletion を使用する方法を示しています。

AWS CLI

カスタマーマネージドKMSキーの削除をスケジュールするには。

次のschedule-key-deletion例では、指定されたカスタマーマネージドKMSキーを 15 日後に削除するようにスケジュールします。

--key-id パラメータはKMSキーを識別します。この例ではキーARN値を使用しますが、キーID またはKMSキーARNの を使用できます。 --pending-window-in-daysパラメータは、7 ~ 30 日間の待機期間の長さを指定します。デフォルトでは、待機時間は 30 日です。この例では、15 の値を指定します。この値は、 コマンドの完了から 15 日後にKMSキーを完全に削除する AWS ように に指示します。

```
aws kms schedule-key-deletion \  
  --key-id arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
  --pending-window-in-days 15
```

レスポンスにはARN、キー、キーの状態、待機期間 (PendingWindowInDays)、Unix 時間の削除日が含まれます。削除日を現地時間で表示するには、 AWS KMSコンソールを使用します。KMS キー状態のPendingDeletionキーは、暗号化オペレーションでは使用できません。

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "DeletionDate": "2022-06-18T23:43:51.272000+00:00",  
  "KeyState": "PendingDeletion",  
  "PendingWindowInDays": 15  
}
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「[Deleting keys](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ScheduleKeyDeletion](#)」の「」を参照してください。

sign

次の例は、sign を使用する方法を説明しています。

AWS CLI

例 1: メッセージのデジタル署名を生成するには

次のsign例では、短いメッセージの暗号化署名を生成します。コマンドの出力には、verify コマンドを使用して検証できる base-64 エンコードされたSignatureフィールドが含まれています。

署名するメッセージと、非対称KMSキーがサポートする署名アルゴリズムを指定する必要があります。KMS キーの署名アルゴリズムを取得するには、describe-key コマンドを使用します。

2.0 AWS CLIでは、messageパラメータの値は Base64-encodedである必要があります。または、メッセージをファイルに保存し、fileb://プレフィックスを使用できます。プレフィックスは、に AWS CLIファイルからバイナリデータを読み取るように指示します。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。キー ID は、KMSキーの使用量が SIGN_ の非対称キーを表す必要がありますVERIFY。

```
msg=(echo 'Hello World' | base64)

aws kms sign \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://UnsignedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256
```

出力:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
```

```
}
```

で非対称KMSキーを使用する方法の詳細については AWS KMS、AWS 「Key Management Service デベロッパーガイド」の「[の非対称キー AWS KMS](#)」を参照してください。

例 2: デジタル署名をファイルに保存するには (Linux と macOS)

次のsign例では、ローカルファイルに保存されている短いメッセージの暗号化署名を生成します。コマンドはレスポンスから Signatureプロパティも取得し、Base64-decodesして ExampleSignature ファイルに保存します。署名を検証するverifyコマンドで署名ファイルを使用できます。

sign コマンドには、Base64-encodedされたメッセージと、非対称KMSキーがサポートする署名アルゴリズムが必要です。KMS キーがサポートする署名アルゴリズムを取得するには、describe-key コマンドを使用します。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。キー ID は、KMSキーの使用量が SIGN_ の非対称キーを表す必要がありますVERIFY。

```
echo 'hello world' | base64 > EncodedMessage

aws kms sign \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://EncodedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
  --output text \
  --query Signature | base64 --decode > ExampleSignature
```

このコマンドでは何も出力されません。この例では、出力の Signatureプロパティを抽出し、ファイルに保存します。

で非対称KMSキーを使用する方法の詳細については AWS KMS、AWS 「Key Management Service デベロッパーガイド」の「[の非対称キー AWS KMS](#)」を参照してください。

- API 詳細については、「[コマンドAWS CLI リファレンス](#)で[サインイン](#)」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

KMSキーにタグを追加するには

次のtag-resource例では、カスターマネージドKMSキーに "Purpose":"Test"と "Dept":"IT" タグを追加します。このようなタグを使用してKMSキーにラベルを付け、アクセス許可と監査用のKMSキーのカテゴリを作成できます。

KMS キーを指定するには、key-idパラメータを使用します。この例ではキー ID 値を使用しますが、このコマンドARNではキー ID またはキーを使用できます。

```
aws kms tag-resource \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

このコマンドでは何も出力されません。KMS キーの AWS KMSタグを表示するには、list-resource-tags コマンドを使用します。

でのタグの使用の詳細については AWS KMS、AWS 「Key Management Service デベロッパーガイド」の「キーの[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

KMSキーからタグを削除するには

次のuntag-resource例では、カスターマネージド"Purpose"キーから KMS キーを持つ タグを削除します。

KMS キーを指定するには、key-idパラメータを使用します。この例ではキー ID 値を使用しますが、このコマンドARNではキー ID またはキーを使用できます。このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。

```
aws kms untag-resource \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

```
--tag-key 'Purpose'
```

このコマンドでは何も出力されません。KMS キーの AWS KMS タグを表示するには、`list-resource-tags` コマンドを使用します。

でのタグの使用の詳細については AWS KMS、AWS 「Key Management Service デベロッパーガイド」の「キーの[タグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-alias

次のコード例は、`update-alias` を使用する方法を示しています。

AWS CLI

エイリアスを別の KMS キーに関連付けるには

次の `update-alias` 例では、エイリアスを別の KMS キー `alias/test-key` に関連付けます。

`--alias-name` パラメータはエイリアス名を指定します。エイリアス名値は `alias/` で始まる必要があります。 `--target-key-id` パラメータは、エイリアスに関連付ける KMS キーを指定します。エイリアスの現在の KMS キーを指定する必要はありません。

```
aws kms update-alias \  
  --alias-name alias/test-key \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

このコマンドでは何も出力されません。エイリアスを見つけるには、`list-aliases` コマンドを使用します。

詳細については、「AWS Key Management Service デベロッパーガイド」の「[エイリアスの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAlias](#)」の「」を参照してください。

update-custom-key-store

次のコード例は、`update-custom-key-store` を使用する方法を示しています。

AWS CLI

例 1: カスタムキーストアのわかりやすい名前を編集するには

次のupdate-custom-key-store例では、カスタムキーストアの名前を変更します。この例では、AWS CloudHSM キーストアまたは外部キーストアで機能します。

custom-key-store-id を使用してキーストアを識別します。new-custom-key-store-name パラメータを使用して、新しいフレンドリ名を指定します。

AWS CloudHSM キーストアのわかりやすい名前を更新するには、disconnect-custom-key-store コマンドを使用するなどして、まずキーストアを切断する必要があります。外部キーストアのフレンドリ名は、接続中または切断中に更新できます。カスタムキーストアの接続状態を確認するには、describe-custom-key-store コマンドを使用します。

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --new-custom-key-store-name ExampleKeyStore
```

このコマンドはデータを返しません。コマンドが機能したことを確認するには、describe-custom-key-stores コマンドを使用します。

AWS CloudHSM キーストアの更新の詳細については、AWS 「Key Management Service デベロッパーガイド」の[AWS 「CloudHSM キーストア設定の編集」](#)を参照してください。

外部キーストアの更新の詳細については、AWS 「Key Management Service デベロッパーガイド」の[「外部キーストアプロパティの編集」](#)を参照してください。

例 2: AWS CloudHSM キーストアの kmsuser パスワードを編集するには

次のupdate-custom-key-store例では、kmsuserパスワードの値を、指定されたキーストアに関連付けられた CloudHSM クラスターkmsuser内の現在のパスワードに更新します。このコマンドは、クラスターのkmsuserパスワードを変更しません。現在のパスワードのみを伝え AWS KMSます。に現在のkmsuserパスワードKMSがない場合は、AWS CloudHSM キーストアに接続できません。

NOTE : AWS CloudHSM キーストアを更新する前に、それを切断する必要があります。disconnect-custom-key-storeコマンドを実行します。コマンドが完了したら、AWS CloudHSM キーストアを再接続できます。connect-custom-key-storeコマンドを実行します。

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --key-store-password ExamplePassword
```

このコマンドは出力を返しません。変更が有効であることを確認するには、`describe-custom-key-stores` コマンドを使用します。

AWS CloudHSM キーストアの更新の詳細については、AWS 「Key Management Service デベロッパーガイド」の[AWS 「CloudHSM キーストア設定の編集」](#)を参照してください。

例 3: AWS CloudHSM キーストアの AWS CloudHSM クラスターを編集するには

次の例では、AWS CloudHSM キーストアに関連付けられている AWS CloudHSM クラスターを、同じクラスターの別のバックアップなど、関連するクラスターに変更します。

NOTE : AWS CloudHSM キーストアを更新する前に、切断する必要があります。 `disconnect-custom-key-store` コマンドを実行します。コマンドが完了したら、AWS CloudHSM キーストアを再接続できます。 `connect-custom-key-store` コマンドを実行します。

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

このコマンドは出力を返しません。変更が有効であることを確認するには、`describe-custom-key-stores` コマンドを使用します。

AWS CloudHSM キーストアの更新の詳細については、AWS 「Key Management Service デベロッパーガイド」の[AWS 「CloudHSM キーストア設定の編集」](#)を参照してください。

例 4: 外部キーストアのプロキシ認証情報を編集するには

次の例では、外部キーストアのプロキシ認証情報を更新します。値のいずれかのみを変更する場合でも `access-key-id`、`raw-secret-access-key` との両方を指定する必要があります。この機能を使用して、無効な認証情報を修正したり、外部キーストアプロキシが認証情報をローテーションしたときに認証情報を変更したりできます。

外部キーストアでの AWS KMS プロキシ認証情報を確立します。次に、このコマンドを使用して認証情報を提供します AWS KMS。AWS KMS は、この認証情報を使用して、外部キーストアプロキシへのリクエストに署名します。

外部キーストアの接続または切断中にプロキシ認証情報を更新できます。カスタムキーストアの接続状態を確認するには、`describe-custom-key-store` コマンドを使用します。

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
RawSecretAccessKey=DXjSUawneL2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

このコマンドは出力を返しません。変更が有効であることを確認するには、`describe-custom-key-stores` コマンドを使用します。

外部キーストアの更新の詳細については、AWS「[Key Management Service デベロッパーガイド](#)」の「[外部キーストアプロパティの編集](#)」を参照してください。

例 5: 外部キーストアのプロキシ接続を編集するには

次の例では、外部キーストアプロキシ接続オプションをパブリックエンドポイント接続から VPC エンドポイントサービス接続に変更します。xks-proxy-connectivity 値の変更に加えて、VPC エンドポイントサービスに関連付けられたプライベート DNS 名を反映するように xks-proxy-uri-endpoint 値を変更する必要があります。また、xks-proxy-vpc-endpoint-service-name 値を追加する必要があります。

NOTE : 外部ストアのプロキシ接続を更新する前に、接続を切断する必要があります。disconnect-custom-key-store コマンドを実行します。コマンドが完了したら、connect-custom-key-store コマンドを使用して外部キーストアを再接続できます。

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \  
  --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \  
  --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-  
example"
```

このコマンドは出力を返しません。変更が有効であることを確認するには、`describe-custom-key-stores` コマンドを使用します。

外部キーストアの更新の詳細については、AWS「[Key Management Service デベロッパーガイド](#)」の「[外部キーストアプロパティの編集](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス UpdateCustomKeyStore](#)」の「」を参照してください。

update-key-description

次のコード例は、update-key-description を使用する方法を示しています。

AWS CLI

例 1: カスタマーマネージドKMSキーに説明を追加または変更するには

次のupdate-key-description例では、カスタマーマネージドKMSキーに説明を追加します。同じコマンドを使用して、既存の説明を変更できます。

--key-id パラメータは、コマンド内のKMSキーを識別します。この例ではキーARN値を使用しますが、キー ID またはKMSキーARNのキーを使用できます。--descriptionパラメータは新しい説明を指定します。このパラメータの値は、KMSキーの現在の説明がある場合は置き換えます。

```
aws kms update-key-description \  
  --key-id arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
  --description "IT Department test key"
```

このコマンドでは何も出力されません。KMS キーの説明を表示するには、describe-key コマンドを使用します。

詳細については、「キー管理サービスリファレンス[UpdateKeyDescription](#)」の「」を参照してください。AWS API

例 2: カスタマーマネージドKMSキーの説明を削除するには

次のupdate-key-description例では、カスタマーマネージドKMSキーの説明を削除します。

--key-id パラメータは、コマンド内のKMSキーを識別します。この例ではキー ID 値を使用しますが、キー ID またはKMSキーARNのキーを使用できます。空の文字列値 ("") を持つ --descriptionパラメータは、既存の説明を削除します。

```
aws kms update-key-description \  
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
  --description ''
```

このコマンドでは何も出力されません。KMS キーの説明を表示するには、describe-key コマンドを使用します。

詳細については、「キー管理サービスリファレンス[UpdateKeyDescription](#)」の「」を参照してください。AWS API

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateKeyDescription](#)」の「」を参照してください。

verify

次の例は、verify を使用する方法を説明しています。

AWS CLI

デジタル署名を検証するには

次のverify例では、Base64-encodedされた短いメッセージの暗号化署名を検証します。キーID、メッセージ、メッセージタイプ、署名アルゴリズムは、メッセージの署名に使用されたものと同じである必要があります。指定する署名は base64 でエンコードすることはできません。sign コマンドが返す署名のデコードについては、sign コマンドの例を参照してください。

コマンドの出力には、署名が検証されたことを示すブール値SignatureValidフィールドが含まれます。署名の検証が失敗した場合、verify コマンドも失敗します。

このコマンドを実行する前に、サンプルキー ID を AWS アカウントの有効なキー ID に置き換えます。

```
aws kms verify \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --message fileb://EncodedMessage \  
  --message-type RAW \  
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \  
  --signature fileb://ExampleSignature
```

出力:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "SignatureValid": true,  
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"  
}
```

で非対称KMSキーを使用する方法の詳細については AWS KMS、AWS 「キー管理服务デベロッパーガイド」の「[非対称キーの使用](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[検証](#)」を参照してください。AWS CLI

を使用した Lake Formation の例 AWS CLI

次のコード例は、Lake Formation AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-lf-tags-to-resource

次のコード例は、add-lf-tags-to-resource を使用する方法を示しています。

AWS CLI

既存のリソースに 1 つ以上の LF タグをアタッチするには

次のadd-lf-tags-to-resource例では、特定の LF タグをテーブルリソースにアタッチします。

```
aws lakeformation add-lf-tags-to-resource \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",
```



```
"Resource": {
  "Table": {
    "CatalogId": "123456789111",
    "DatabaseName": "tpc",
    "Name": "dl_tpc_promotion"
  }
},
"LFTags": [{
  "CatalogId": "123456789111",
  "TagKey": "usergroup",
  "TagValues": [
    "analyst"
  ]
}]
}
```

出力:

```
{
  "Failures": []
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「データカタログリソースへの LF タグの割り当て」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AddLFTagsToResource](#)」の「」を参照してください。

batch-grant-permissions

次の例は、batch-grant-permissions を使用する方法を説明しています。

AWS CLI

プリンシパルにリソースに対するアクセス許可を一括付与するには

次のbatch-grant-permissions例では、指定されたリソースへのアクセスをプリンシパルに一括で許可します。

```
aws lakeformation batch-grant-permissions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "CatalogId": "123456789111",
  "Entries": [{
    "Id": "1",
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
    },
    "Resource": {
      "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_promotion"
      }
    },
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": [
      "ALL"
    ]
  },
  {
    "Id": "2",
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
    },
    "Resource": {
      "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_customer"
      }
    },
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": [
      "ALL"
    ]
  },
}
```

```
{
  "Id": "3",
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
  },
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "Name": "dl_tpc_promotion"
    }
  },
  "Permissions": [
    "ALL"
  ],
  "PermissionsWithGrantOption": [
    "ALL"
  ]
},
{
  "Id": "4",
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
developer"
  },
  "Resource": {
    "DataCellsFilter": {
      "TableCatalogId": "123456789111",
      "DatabaseName": "tpc",
      "TableName": "dl_tpc_item",
      "Name": "developer_item"
    }
  },
  "Permissions": [
    "SELECT"
  ],
  "PermissionsWithGrantOption": []
}
]
```

出力:

```
{
  "Failures": []
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[Data Catalog リソースに対するアクセス許可の付与と取り消し](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGrantPermissions](#)」の「」を参照してください。

batch-revoke-permissions

次のコード例は、batch-revoke-permissions を使用方法を示しています。

AWS CLI

プリンシパルからリソースに対するアクセス許可を一括で取り消すには

次の一括batch-revoke-permissions例では、プリンシパルから指定されたリソースへのアクセスを取り消します。

```
aws lakeformation batch-revoke-permissions \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "CatalogId": "123456789111",
  "Entries": [{
    "Id": "1",
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
      "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_promotion"
      }
    }
  }
],
}
```

```
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": [
      "ALL"
    ]
  },
  {
    "Id": "2",
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
business-analyst"
    },
    "Resource": {
      "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_promotion"
      }
    },
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": [
      "ALL"
    ]
  }
]
```

出力:

```
{
  "Failures": []
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「Data Catalog リソースに対するアクセス許可の付与と取り消し」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchRevokePermissions](#)」の「」を参照してください。

cancel-transaction

次のコード例は、cancel-transaction を使用する方法を示しています。

AWS CLI

トランザクションをキャンセルするには

次のcancel-transaction例では、トランザクションをキャンセルします。

```
aws lakeformation cancel-transaction \  
  --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの [「トランザクション内のデータレイクの読み取りと書き込み」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CancelTransaction](#)」の「」を参照してください。

commit-transaction

次のコード例は、commit-transaction を使用する方法を示しています。

AWS CLI

トランザクションをコミットするには

次のcommit-transaction例では、トランザクションをコミットします。

```
aws lakeformation commit-transaction \  
  --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

出力:

```
{  
  "TransactionStatus": "committed"  
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「トランザクション内のデータレイクの読み取りと書き込み」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CommitTransaction](#)」の「」を参照してください。

create-data-cells-filter

次のコード例は、create-data-cells-filter を使用する方法を示しています。

AWS CLI

例 1: データセルフィルターを作成するには

次のcreate-data-cells-filter例では、データセルフィルターを作成して、行条件に基づいて特定の列へのアクセスを許可できるようにします。

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "TableData": {  
    "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],  
    "DatabaseName": "tpc",  
    "Name": "developer_promotion",  
    "RowFilter": {  
      "FilterExpression": "p_promo_name='ese'"  
    },  
    "TableCatalogId": "123456789111",  
    "TableName": "dl_tpc_promotion"  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation のデータフィルタリングとセルレベルのセキュリティAWS」](#)を参照してください。

例 2: 列フィルターを作成するには

次のcreate-data-cells-filter例では、特定の列へのアクセスを許可するためのデータフィルターを作成します。

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "TableData": {  
    "ColumnNames": ["p_channel_details", "p_start_date_sk", "p_promo_name"],  
    "DatabaseName": "tpc",  
    "Name": "developer_promotion_allrows",  
    "RowFilter": {  
      "AllRowsWildcard": {}  
    },  
    "TableCatalogId": "123456789111",  
    "TableName": "dl_tpc_promotion"  
  }  
}
```

このコマンドでは何も出力されません。

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation のデータフィルタリングとセルレベルのセキュリティAWS」](#)を参照してください。

例 3: 除外列でデータフィルターを作成するには

次のcreate-data-cells-filter例では、データフィルターを作成して、前述の列を除くすべてのにアクセスを許可できるようにします。

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "TableData": {  
    "ColumnWildcard": {  
      "ExcludedColumnNames": ["p_channel_details", "p_start_date_sk"]  
    },  
    "DatabaseName": "tpc",  
    "Name": "developer_promotion_excludecolumn",  
    "RowFilter": {
```



```
    "AllRowsWildcard": {}  
  },  
  "TableCatalogId": "123456789111",  
  "TableName": "dl_tpc_promotion"  
}  
}
```

このコマンドでは何も出力されません。

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation のデータフィルタリングとセルレベルのセキュリティAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDataCellsFilter](#)」の「」を参照してください。

create-lf-tag

次の例は、create-lf-tag を使用する方法を説明しています。

AWS CLI

LF タグを作成するには

次のcreate-lf-tag例では、指定された名前と値を持つ LF タグを作成します。

```
aws lakeformation create-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup' \  
  --tag-values '["developer","analyst","campaign"]'
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの[「メタデータアクセスコントロールの LF タグの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLfTag](#)」の「」を参照してください。

delete-data-cells-filter

次のコード例は、delete-data-cells-filter を使用する方法を示しています。

AWS CLI

データセルフィルターを削除するには

次のdelete-data-cells-filter例では、指定されたデータセルフィルターを削除します。

```
aws lakeformation delete-data-cells-filter \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "TableCatalogId": "123456789111",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_promotion",  
  "Name": "developer_promotion"  
}
```

このコマンドでは何も出力されません。

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation のデータフィルタリングとセルレベルのセキュリティAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDataCellsFilter](#)」の「」を参照してください。

delete-lf-tag

次の例は、delete-lf-tag を使用する方法を説明しています。

AWS CLI

LF タグ定義を削除するには

次のdelete-lf-tag例では、LF タグ定義を削除します。

```
aws lakeformation delete-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup'
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[メタデータアクセスコントロールの LF タグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLfTag](#)」の「」を参照してください。

delete-objects-on-cancel

次のコード例は、delete-objects-on-cancel を使用する方法を示しています。

AWS CLI

トランザクションがキャンセルされたときにオブジェクトを削除するには

次のdelete-objects-on-cancel例では、トランザクションがキャンセルされると、リストされている s3 オブジェクトを削除します。

```
aws lakeformation delete-objects-on-cancel \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_household_demographics_gov",  
  "TransactionId": "1234d972ca8347b89825e33c5774aec4",  
  "Objects": [{  
    "Uri": "s3://lf-data-lake-012345678901/target/  
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-  
ff26b17504414fe88b302cd795eabd00.parquet",  
    "ETag": "1234ab1fc50a316b149b4e1f21a73800"  
  }]  
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクション内のデータレイクの読み取りと書き込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteObjectsOnCancel](#)」の「」を参照してください。

deregister-resource

次のコード例は、deregister-resource を使用する方法を示しています。

AWS CLI

データレイクストレージの登録を解除するには

次のderegister-resource例では、Lake Formation によって管理されているリソースを登録解除します。

```
aws lakeformation deregister-resource \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123"  
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの [「データレイクへの Amazon S3 の場所の追加」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterResource](#)」の「」を参照してください。

describe-transaction

次のコード例は、describe-transaction を使用する方法を示しています。

AWS CLI

トランザクションの詳細を取得するには

次のdescribe-transaction例では、単一のトランザクションの詳細を返します。

```
aws lakeformation describe-transaction \  
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

出力:

```
{
  "TransactionDescription": {
    "TransactionId": "12345972ca8347b89825e33c5774aec4",
    "TransactionStatus": "committed",
    "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
    "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
  }
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクション内のデータレイクの読み取りと書き込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTransaction](#)」の「」を参照してください。

extend-transaction

次のコード例は、extend-transaction を使用する方法を示しています。

AWS CLI

トランザクションを拡張するには

次のextend-transaction例では、トランザクションを拡張します。

```
aws lakeformation extend-transaction \  
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクション内のデータレイクの読み取りと書き込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ExtendTransaction](#)」の「」を参照してください。

get-data-lake-settings

次の例は、get-data-lake-settings を使用する方法を説明しています。

AWS CLI

AWS Lake Formation が管理するデータレイク設定を取得するには

次のget-data-lake-settings例では、データレイク管理者およびその他のデータレイク設定のリストを取得します。

```
aws lakeformation get-data-lake-settings \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111"  
}
```

出力:

```
{  
  "DataLakeSettings": {  
    "DataLakeAdmins": [{  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"  
    }],  
    "CreateDatabaseDefaultPermissions": [],  
    "CreateTableDefaultPermissions": [  
      {  
        "Principal": {  
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"  
        },  
        "Permissions": [  
          "ALL"  
        ]  
      }  
    ],  
    "TrustedResourceOwners": [],  
    "AllowExternalDataFiltering": true,  
    "ExternalDataFilteringAllowList": [{  
      "DataLakePrincipalIdentifier": "123456789111"  
    }],  
    "AuthorizedSessionTagValueList": [  
      "Amazon EMR"  
    ]  
  }  
}
```

```
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[データレイクのデフォルトのセキュリティ設定を変更する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDataLakeSettings](#)」の「」を参照してください。

get-effective-permissions-for-path

次のコード例は、get-effective-permissions-for-path を使用する方法を示しています。

AWS CLI

特定のパスにあるリソースに対するアクセス許可を取得するには

次のget-effective-permissions-for-path例では、Amazon S3 のパスにある指定されたテーブルまたはデータベースリソースの Lake Formation アクセス許可を返します。

```
aws lakeformation get-effective-permissions-for-path \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"  
}
```

出力:

```
{  
  "Permissions": [{  
    "Principal": {  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-campaign-manager"  
    },  
    "Resource": {  
      "Database": {  
        "Name": "tpc"  
      }  
    },  
  },  
]
```

```
    "Permissions": [
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": []
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/EMR-
RuntimeRole"
    },
    "Resource": {
      "Database": {
        "Name": "tpc"
      }
    },
    "Permissions": [
      "ALL"
    ],
    "PermissionsWithGrantOption": []
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:saml-
provider/oktaSAMLProvider:user/emr-developer"
    },
    "Resource": {
      "Database": {
        "Name": "tpc"
      }
    },
    "Permissions": [
      "ALL",
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": []
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    },
    "Resource": {
      "Database": {
        "Name": "tpc"
      }
    }
  }
}
```



```

    }
  },
  "Permissions": [
    "ALL",
    "ALTER",
    "CREATE_TABLE",
    "DESCRIBE",
    "DROP"
  ],
  "PermissionsWithGrantOption": [
    "ALL",
    "ALTER",
    "CREATE_TABLE",
    "DESCRIBE",
    "DROP"
  ]
},
{
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/LF-
GlueServiceRole"
  },
  "Resource": {
    "Database": {
      "Name": "tpc"
    }
  },
  "Permissions": [
    "CREATE_TABLE"
  ],
  "PermissionsWithGrantOption": []
}
],
"NextToken":
"E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjBUywYVc5dU1qcDdJbk5sWTI5dVpITW1P
}

```

詳細については、[「Lake Formation デベロッパーガイド」](#)の「[Lake Formation アクセス許可の管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスGetEffectivePermissionsForPath](#)」の「」を参照してください。

get-lf-tag

次のコード例は、get-lf-tag を使用する方法を示しています。

AWS CLI

LF タグ定義を取得するには

次のget-lf-tag例では、LF タグ定義を取得します。

```
aws lakeformation get-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup'
```

出力:

```
{  
  "CatalogId": "123456789111",  
  "TagKey": "usergroup",  
  "TagValues": [  
    "analyst",  
    "campaign",  
    "developer"  
  ]  
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[メタデータアクセスコントロールの LF タグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLfTag](#)」の「」を参照してください。

get-query-state

次のコード例は、get-query-state を使用する方法を示しています。

AWS CLI

送信されたクエリの状態を取得するには

次のget-query-state例では、以前に送信されたクエリの状態を返します。

```
aws lakeformation get-query-state \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

出力:

```
{  
  "State": "FINISHED"  
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクションデータオペレーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetQueryState](#)」の「」を参照してください。

get-query-statistics

次の例は、get-query-statistics を使用する方法を説明しています。

AWS CLI

クエリ統計を取得するには

次のget-query-statistics例では、クエリの計画と実行に関する統計を取得します。

```
aws lakeformation get-query-statistics \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

出力:

```
{  
  "ExecutionStatistics": {  
    "AverageExecutionTimeMillis": 0,  
    "DataScannedBytes": 0,  
    "WorkUnitsExecutedCount": 0  
  },  
  "PlanningStatistics": {  
    "EstimatedDataToScanBytes": 43235,  
    "PlanningTimeMillis": 2377,  
    "QueueTimeMillis": 440,  
  }  
}
```

```
    "WorkUnitsGeneratedCount": 1
  },
  "QuerySubmissionTime": "2022-08-11T02:14:38.641870+00:00"
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクションデータオペレーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetQueryStatistics](#)」の「」を参照してください。

get-resource-lf-tags

次のコード例は、get-resource-lf-tags を使用する方法を示しています。

AWS CLI

LF タグを一覧表示するには

次のlist-lf-tags例では、リクエストが表示するアクセス許可を持つ LF タグのリストを返します。

```
aws lakeformation list-lf-tags \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "ResourceShareType": "ALL",  
  "MaxResults": 2  
}
```

出力:

```
{  
  "LFTags": [{  
    "CatalogId": "123456789111",  
    "TagKey": "category",  
    "TagValues": [  
      "value1",  
      "value2"  
    ]  
  }  
]
```

```
        "private",
        "public"
    ]
},
{
    "CatalogId": "123456789111",
    "TagKey": "group",
    "TagValues": [
        "analyst",
        "campaign",
        "developer"
    ]
}],
"NextToken": "kIiwiZXhwaXJhdGlvbiI6eyJzZWVbmRzIjoxNjYwMDY4dCI6ZmFsc2V9"
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[メタデータアクセスコントロールの LF タグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceLfTags](#)」の「」を参照してください。

get-table-objects

次の例は、get-table-objects を使用する方法を説明しています。

AWS CLI

管理対象テーブルのオブジェクトを一覧表示するには

次のget-table-objects例では、指定された管理対象テーブルを構成する Amazon S3 オブジェクトのセットを返します。

```
aws lakeformation get-table-objects \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",
```

```
"TableName": "dl_tpc_household_demographics_gov",
"QueryAsOfTime": "2022-08-10T15:00:00"
}
```

出力:

```
{
  "Objects": [{
    "PartitionValues": [],
    "Objects": [{
      "Uri": "s3://lf-data-lake-012345678901/target/
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-
ff26b17504414fe88b302cd795eabd00.parquet",
      "ETag": "12345b1fc50a316b149b4e1f21a73800",
      "Size": 43235
    }]
  }]
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクション内のデータレイクの読み取りと書き込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTableObjects](#)」の「」を参照してください。

get-work-unit-results

次の例は、get-work-unit-results を使用する方法を説明しています。

AWS CLI

特定のクエリの作業単位を取得するには

次のget-work-unit-results例では、クエリの結果である作業単位を返します。

```
aws lakeformation get-work-units \
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b' \
  --work-unit-id '0' \
  --work-unit-token 'B2fMSdmQXe9umX8Ux8XCo4=' outfile
```

出力:

```
outfile with Blob content.
```

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクションデータオペレーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWorkUnitResults](#)」の「」を参照してください。

get-work-units

次の例は、get-work-units を使用する方法を説明しています。

AWS CLI

作業単位を取得するには

次のget-work-units例では、StartQueryPlanning オペレーションによって生成された作業単位を取得します。

```
aws lakeformation get-work-units \
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

出力:

```
{
  "WorkUnitRanges": [{
    "WorkUnitIdMax": 0,
    "WorkUnitIdMin": 0,
    "WorkUnitToken":
      "1234eMAk4kL04umqEL4Z5WuxL04AXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREEwYm9QbkhINmFYTWphbmMxZW
      +f88jzGrYq22gE6jkQ1p0B
      +0et2eqNumFudAAAAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIB3DQEHATAeBg1ghkgBZQMEAS4wEQQMCOEWRda
      wAAAAEAAAAAAAAAAAAAAAAAAEAAACX3/w5h75QAPomfKH+cyEKYU1yccUmB1
      +VSojiG0tdsUk7vcjYXUUb0Ym3dvqRqX2s4gROM0n
      +Ij8R0/8jYmnHkpvYAFNVRPyETyIKg7k5Z9+5I1c2d3446Jw/moWGGxjH8AEG9h27ytm0hozxDOEi/
      F2ZoXz6w1GDfGUo/2WxCKY0hTyNaw6TM
      +7d1TM7yrW4iNVLUM0LX0xnFjIAhLhooWJek6vjQZUAZzB1AjBH8okRtYP8R7AY2W1s/
      hqFBhG0V4l42AC0LxsuZbMQrE2SzwZUZ0E9Uew7/n0cyX4CMQDR79INyv4ysMByW9kKGGKyba+cCNk1ExMR
      +btBQBmMuB2fMSdmQXe9umX8Ux8XCo4="
  }],
  "QueryId": "1234273f-4a62-4cda-8d98-69615ee8be9b"
```

```
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクションデータオペレーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWorkUnits](#)」の「」を参照してください。

grant-permissions

次のコード例は、grant-permissions を使用する方法を示しています。

AWS CLI

例 1: LF タグを使用してリソースのプリンシパルにアクセス許可を付与するには

次のgrant-permissions例では、LF タグポリシーに一致するデータベースリソースのプリンシパルにアクセスALL許可を付与します。

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-admin"  
  },  
  "Resource": {  
    "LFTagPolicy": {  
      "CatalogId": "123456789111",  
      "ResourceType": "DATABASE",  
      "Expression": [{  
        "TagKey": "usergroup",  
        "TagValues": [  
          "analyst",  
          "developer"  
        ]  
      }]  
    }  
  }  
}
```



```
  },
  "Permissions": [
    "ALL"
  ],
  "PermissionsWithGrantOption": [
    "ALL"
  ]
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの [「Data Catalog リソースに対するアクセス許可の付与と取り消し」](#) を参照してください。

例 2: プリンシパルに列レベルのアクセス許可を付与するには

次のgrant-permissions例では、プリンシパルに特定の列を選択するアクセス許可を付与します。

```
aws lakeformation grant-permissions \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "CatalogId": "123456789111",
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
  },
  "Resource": {
    "TableWithColumns": {
      "CatalogId": "123456789111",
      "ColumnNames": ["p_end_date_sk"],
      "DatabaseName": "tpc",
      "Name": "dl_tpc_promotion"
    }
  },
  "Permissions": [
    "SELECT"
  ],
  "PermissionsWithGrantOption": []
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの [「Data Catalog リソースに対するアクセス許可の付与と取り消し」](#) を参照してください。

例 3: プリンシパルにテーブルアクセス許可を付与するには

次のgrant-permissions例では、特定のデータベースのすべてのテーブルに対する選択アクセス許可をプリンシパルに付与します。

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"  
  },  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "TableWildcard": {}  
    }  
  },  
  "Permissions": [  
    "SELECT"  
  ],  
  "PermissionsWithGrantOption": []  
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの [「Data Catalog リソースに対するアクセス許可の付与と取り消し」](#) を参照してください。

例 4: LF タグに対するアクセス許可をプリンシパルに付与するには

次のgrant-permissions例では、LF タグに対するアクセス許可をプリンシパルに付与します。

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"  
  },  
  "Resource": {  
    "LFTag": {  
      "CatalogId": "123456789111",  
      "TagKey": "category",  
      "TagValues": [  
        "private", "public"  
      ]  
    }  
  },  
  "Permissions": [  
    "ASSOCIATE"  
  ],  
  "PermissionsWithGrantOption": []  
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの [「Data Catalog リソースに対するアクセス許可の付与と取り消し」](#) を参照してください。

例 5: データロケーションに対するアクセス許可をプリンシパルに付与するには

次のgrant-permissions例では、プリンシパルにデータの場所に対するアクセス許可を付与します。

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{
```

```
"CatalogId": "123456789111",
"Principal": {
  "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
},
"Resource": {
  "DataLocation": {
    "CatalogId": "123456789111",
    "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"
  }
},
"Permissions": [
  "DATA_LOCATION_ACCESS"
],
"PermissionsWithGrantOption": []
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[Data Catalog リソースに対するアクセス許可の付与と取り消し](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GrantPermissions](#)」の「」を参照してください。

list-data-cells-filter

次のコード例は、list-data-cells-filter を使用する方法を示しています。

AWS CLI

データセルフィルターを一覧表示するには

次のlist-data-cells-filter例では、特定のテーブルのデータセルフィルターを一覧表示します。

```
aws lakeformation list-data-cells-filter \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "MaxResults": 2,
```



```
  ],  
  "NextToken": "2MDA2MTgwNiwibmFub3MiOjE0MDAwMDAwMH19"  
}
```

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation のデータフィルタリングとセルレベルのセキュリティAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDataCellsFilter](#)」の「」を参照してください。

list-permissions

次の例は、list-permissions を使用する方法を説明しています。

AWS CLI

例 1: リソースのプリンシパルアクセス許可のリストを取得するには

次のlist-permissions例では、データベースリソースに対するプリンシパルアクセス許可のリストを返します。

```
aws lakeformation list-permissions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "ResourceType": "DATABASE",  
  "MaxResults": 2  
}
```

出力:

```
{  
  "PrincipalResourcePermissions": [{  
    "Principal": {  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-  
campaign-manager"  
    },  
    "Resource": {  
      "Database": {
```

```

        "CatalogId": "123456789111",
        "Name": "tpc"
      }
    },
    "Permissions": [
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": []
  ]],
  "NextToken":
  "E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjBUywYVc5dU1qcDdJbk5sWTI5dVpITW1P
}

```

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation アクセス許可の管理」](#)を参照してください。AWS

例 2: データフィルターを使用してテーブルのプリンシパルアクセス許可のリストを取得するには次のlist-permissions例では、プリンシパルに付与された関連するデータフィルターを持つテーブルのアクセス許可を一覧表示します。

```

aws lakeformation list-permissions \
  --cli-input-json file://input.json

```

input.json の内容:

```

{
  "CatalogId": "123456789111",
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "Name": "dl_tpc_customer"
    }
  },
  "IncludeRelated": "TRUE",
  "MaxResults": 10
}

```

出力:

```

{

```

```
"PrincipalResourcePermissions": [{
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
  },
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "customer",
      "Name": "customer_invoice"
    }
  },
  "Permissions": [
    "ALL",
    "ALTER",
    "DELETE",
    "DESCRIBE",
    "DROP",
    "INSERT"
  ],
  "PermissionsWithGrantOption": [
    "ALL",
    "ALTER",
    "DELETE",
    "DESCRIBE",
    "DROP",
    "INSERT"
  ]
}],
{
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
  },
  "Resource": {
    "TableWithColumns": {
      "CatalogId": "123456789111",
      "DatabaseName": "customer",
      "Name": "customer_invoice",
      "ColumnWildcard": {}
    }
  },
  "Permissions": [
    "SELECT"
  ]
}
```



```

    ],
    "PermissionsWithGrantOption": [
        "SELECT"
    ]
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:role/
Admin"
    },
    "Resource": {
      "DataCellsFilter": {
        "TableCatalogId": "123456789111",
        "DatabaseName": "customer",
        "TableName": "customer_invoice",
        "Name": "dl_us_customer"
      }
    },
    "Permissions": [
      "DESCRIBE",
      "SELECT",
      "DROP"
    ],
    "PermissionsWithGrantOption": []
  }
],
"NextToken": "VyeUFjY291bnRQZXJtaXNzaW9ucyI6ZmFsc2V9"
}

```

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation アクセス許可の管理」](#)を参照してください。AWS

例 3: LF タグのプリンシパルアクセス許可のリストを取得するには

次のlist-permissions例では、プリンシパルに付与された LF タグに対するアクセス許可を一覧表示します。

```
aws lakeformation list-permissions \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
```

```

    "CatalogId": "123456789111",
    "Resource": {
      "LFTag": {
        "CatalogId": "123456789111",
        "TagKey": "category",
        "TagValues": [
          "private"
        ]
      }
    },
    "MaxResults": 10
  }
}

```

出力:

```

{
  "PrincipalResourcePermissions": [{
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    },
    "Resource": {
      "LFTag": {
        "CatalogId": "123456789111",
        "TagKey": "category",
        "TagValues": [
          "*"
        ]
      }
    },
    "Permissions": [
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": [
      "DESCRIBE"
    ]
  },
  {
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    },
    "Resource": {

```

```
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "*"
            ]
        },
        "Permissions": [
            "ASSOCIATE"
        ],
        "PermissionsWithGrantOption": [
            "ASSOCIATE"
        ]
    },
    "NextToken": "EJwY21GMGFjX0VJanA3SW50cm1pc3Npb25zIjpmYWxzZX0="
}
```

詳細については、[「Lake Formation デベロッパーガイド」の「Lake Formation アクセス許可の管理」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPermissions](#)」の「」を参照してください。

list-resources

次のコード例は、list-resources を使用する方法を示しています。

AWS CLI

Lake Formation によって管理されるリソースを一覧表示するには

次のlist-resources例では、Lake Formation によって管理される条件に一致するリソースを一覧表示します。

```
aws lakeformation list-resources \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{
```

```
"FilterConditionList": [{
  "Field": "ROLE_ARN",
  "ComparisonOperator": "CONTAINS",
  "StringValueList": [
    "123456789111"
  ]
}],
"MaxResults": 10
}
```

出力:

```
{
  "ResourceInfoList": [{
    "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111",
    "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole",
    "LastModified": "2022-07-21T02:12:46.669000+00:00"
  },
  {
    "ResourceArn": "arn:aws:s3:::lf-emr-test-123456789111",
    "RoleArn": "arn:aws:iam::123456789111:role/EMRLFS3Role",
    "LastModified": "2022-07-29T16:22:03.211000+00:00"
  }
  ]
}
```

詳細については、[「Lake Formation デベロッパーガイド」](#)の「[Lake Formation アクセス許可の管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスListResources](#)」の「」を参照してください。

list-transactions

次の例は、list-transactions を使用する方法を説明しています。

AWS CLI

すべてのトランザクションの詳細を一覧表示するには

次のlist-transactions例では、トランザクションとそのステータスに関するメタデータを返します。

```
aws lakeformation list-transactions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "StatusFilter": "ALL",  
  "MaxResults": 3  
}
```

出力:

```
{  
  "Transactions": [{  
    "TransactionId": "1234569f08804cb790d950d4d0fe485e",  
    "TransactionStatus": "committed",  
    "TransactionStartTime": "2022-08-10T14:32:29.220000+00:00",  
    "TransactionEndTime": "2022-08-10T14:32:33.751000+00:00"  
  },  
  {  
    "TransactionId": "12345972ca8347b89825e33c5774aec4",  
    "TransactionStatus": "committed",  
    "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",  
    "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"  
  },  
  {  
    "TransactionId": "12345daf6cb047dbba8ad9b0414613b2",  
    "TransactionStatus": "committed",  
    "TransactionStartTime": "2022-08-10T13:56:51.261000+00:00",  
    "TransactionEndTime": "2022-08-10T13:56:51.547000+00:00"  
  }  
],  
  "NextToken": "77X1ebypsI7os+X2lhHsZLGNCDK3nNgpWRdFpicS0HgcX1/  
QMoniUAKcpR3kj3ts3PVdMA=="  
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「トランザクション内のデータレイクの読み取りと書き込み」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListTransactions](#)」の「」を参照してください。

put-data-lake-settings

次のコード例は、put-data-lake-settings を使用する方法を示しています。

AWS CLI

AWS Lake Formation が管理するデータレイクの設定を行うには

次のput-data-lake-settings例では、データレイク管理者およびその他のデータレイク設定のリストを設定します。

```
aws lakeformation put-data-lake-settings \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "DataLakeSettings": {  
    "DataLakeAdmins": [{  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-  
admin"  
    }  
  ],  
  "CreateDatabaseDefaultPermissions": [],  
  "CreateTableDefaultPermissions": [],  
  "TrustedResourceOwners": [],  
  "AllowExternalDataFiltering": true,  
  "ExternalDataFilteringAllowList": [{  
    "DataLakePrincipalIdentifier": "123456789111"  
  }],  
  "AuthorizedSessionTagValueList": ["Amazon EMR"]  
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[データレイクのデフォルトのセキュリティ設定を変更する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutDataLakeSettings](#)」の「」を参照してください。

register-resource

次の例は、register-resource を使用する方法を説明しています。

AWS CLI

例 1: Service Linked Role を使用してデータレイクストレージを登録するには

次のregister-resource例では、サービスリンクロールを使用して Lake Formation によって管理されるリソースを登録します。

```
aws lakeformation register-resource \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",  
  "UseServiceLinkedRole": true  
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの [「データレイクへの Amazon S3 の場所の追加」](#) を参照してください。

例 2: カスタムロールを使用してデータレイクストレージを登録するには

次のregister-resource例では、カスタムロールを使用して Lake Formation によって管理されるリソースを登録します。

```
aws lakeformation register-resource \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",  
  "UseServiceLinkedRole": false,  
  "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole"
```

```
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[データレイクへの Amazon S3 の場所の追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterResource](#)」の「」を参照してください。

remove-lf-tags-from-resource

次のコード例は、remove-lf-tags-from-resource を使用する方法を示しています。

AWS CLI

リソースから LF タグを削除するには

次のremove-lf-tags-from-resource例では、テーブルリソースとの LF タグの関連付けを削除します。

```
aws lakeformation remove-lf-tags-from-resource \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  },  
  "LFTags": [{  
    "CatalogId": "123456789111",  
    "TagKey": "usergroup",  
    "TagValues": [  
      "developer"  
    ]  
  }  
}
```



```
    ]]  
  }
```

出力:

```
{  
  "Failures": []  
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「データカタログリソースへの LF タグの割り当て」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveLfTagsFromResource](#)」の「」を参照してください。

revoke-permissions

次のコード例は、revoke-permissions を使用する方法を示しています。

AWS CLI

プリンシパルからリソースに対するアクセス許可を取り消すには

次のrevoke-permissions例では、特定のデータベースの特定のテーブルへのプリンシパルアクセスを取り消します。

```
aws lakeformation revoke-permissions \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"  
  },  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  }  
}
```

```
    }
  },
  "Permissions": [
    "ALL"
  ],
  "PermissionsWithGrantOption": []
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[Data Catalog リソースに対するアクセス許可の付与と取り消し](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RevokePermissions](#)」の「」を参照してください。

search-databases-by-lf-tags

次のコード例は、search-databases-by-lf-tags を使用する方法を示しています。

AWS CLI

でデータベースリソースを検索するには LFTags

次のsearch-databases-by-lf-tags例では、LFTag式に一致するデータベースリソースを検索します。

```
aws lakeformation search-databases-by-lf-tags \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "MaxResults": 1,
  "CatalogId": "123456789111",
  "Expression": [{
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
}
```

```
}
```

出力:

```
{
  "DatabaseList": [{
    "Database": {
      "CatalogId": "123456789111",
      "Name": "tpc"
    },
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  }]
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「LF タグが割り当てられているリソースの表示」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SearchDatabasesByLfTags](#)」の「」を参照してください。

search-tables-by-lf-tags

次の例は、search-tables-by-lf-tags を使用する方法を説明しています。

AWS CLI

でテーブルリソースを検索するには LFTags

次のsearch-tables-by-lf-tags例では、LFTag式に一致するテーブルリソースを検索します。

```
aws lakeformation search-tables-by-lf-tags \
  --cli-input-json file://input.json
```

input.json の内容:

```
{
  "MaxResults": 2,
  "CatalogId": "123456789111",
  "Expression": [{
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
}
```

出力:

```
{
  "NextToken": "c2VhcmNoQWxsVGFnc0luVGFibGVzIjpmYWxzZX0=",
  "TableList": [{
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "Name": "dl_tpc_item"
    },
    "LFTagOnDatabase": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }
  ]},
  "LFTagsOnTable": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }],
  "LFTagsOnColumns": [{
    "Name": "i_item_desc",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }
  ]
}
```

```
    ]],
  },
  {
    "Name": "i_container",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_wholesale_cost",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_manufact_id",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_brand_id",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_formulation",
```

```
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_current_price",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_size",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_rec_start_date",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_manufact",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
```

```
        "developer"
      ]
    ]}
  },
  {
    "Name": "i_item_sk",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_manager_id",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_item_id",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
  {
    "Name": "i_class_id",
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  },
},
```

```
{
  "Name": "i_class",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_category",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_category_id",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_brand",
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
},
{
  "Name": "i_units",
  "LFTags": [{
    "CatalogId": "123456789111",
```



```
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }],
},
{
    "Name": "i_rec_end_date",
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
},
{
    "Name": "i_color",
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
},
{
    "Name": "i_product_name",
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
]
}]
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「LF タグが割り当てられているリソースの表示」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchTablesByLfTags](#)」の「」を参照してください。

start-query-planning

次のコード例は、start-query-planning を使用する方法を示しています。

AWS CLI

クエリステートメントを処理するには

次のstart-query-planning例では、クエリステートメントを処理するリクエストを送信します。

```
aws lakeformation start-query-planning \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "QueryPlanningContext": {  
    "CatalogId": "012345678901",  
    "DatabaseName": "tpc"  
  },  
  "QueryString": "select * from dl_tpc_household_demographics_gov where  
hd_income_band_sk=9"  
}
```

出力:

```
{  
  "QueryId": "772a273f-4a62-4cda-8d98-69615ee8be9b"  
}
```

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクション内のデータレイクの読み取りと書き込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartQueryPlanning](#)」の「」を参照してください。

start-transaction

次のコード例は、start-transaction を使用する方法を示しています。

AWS CLI

新しいトランザクションを開始するには

次のstart-transaction例では、新しいトランザクションを開始し、そのトランザクション ID を返します。

```
aws lakeformation start-transaction \  
  --transaction-type = 'READ_AND_WRITE'
```

出力:

```
{  
  "TransactionId": "b014d972ca8347b89825e33c5774aec4"  
}
```

詳細については、AWS Lake Formation デベロッパーガイドの [「トランザクション内のデータレイクの読み取りと書き込み」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartTransaction](#)」の「」を参照してください。

update-lf-tag

次の例は、update-lf-tag を使用する方法を説明しています。

AWS CLI

LF タグ定義を更新するには

次のupdate-lf-tag例では、LF タグ定義を更新します。

```
aws lakeformation update-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup' \  
  --tag-values-to-add ['admin']
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[メタデータアクセスコントロールの LF タグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLfTag](#)」の「」を参照してください。

update-table-objects

次のコード例は、update-table-objects を使用する方法を示しています。

AWS CLI

管理対象テーブルのオブジェクトを変更するには

次のupdate-table-objects例では、指定された管理対象テーブルに指定された S3 オブジェクトを追加します。

```
aws lakeformation update-table-objects \  
  --cli-input-json file://input.json
```

input.json の内容:

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_household_demographics_gov",  
  "TransactionId": "12347a9f75424b9b915f6ff201d2a190",  
  "WriteOperations": [{  
    "AddObject": {  
      "Uri": "s3://lf-data-lake-012345678901/target/  
dl_tpc_household_demographics_gov/run-unnamed-1-part-block-0-r-00000-snappy-  
ff26b17504414fe88b302cd795eabd00.parquet",  
      "ETag": "1234ab1fc50a316b149b4e1f21a73800",  
      "Size": 42200  
    }  
  }  
}]  
}
```

このコマンドでは何も出力されません。

詳細については、AWS Lake Formation デベロッパーガイドの「[トランザクション内のデータレイクの読み取りと書き込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTableObjects](#)」の「」を参照してください。

を使用した Lambda の例 AWS CLI

次のコード例は、Lambda AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-layer-version-permission

次の例は、add-layer-version-permission を使用する方法を説明しています。

AWS CLI

レイヤーバージョンにアクセス許可を追加するには

次のadd-layer-version-permission例では、レイヤーのバージョン 1 を使用するアクセス許可を指定されたアカウントに付与しますmy-layer。

```
aws lambda add-layer-version-permission \  
  --layer-name my-layer \  
  --statement-id xaccount \  
  --action lambda:GetLayerVersion \  
  --principal 123456789012 \  
  --version-number 1
```

出力:

```
{
  "RevisionId": "35d87451-f796-4a3f-a618-95a3671b0a0c",
  "Statement":
  {
    "Sid": "xaccount",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::210987654321:root"
    },
    "Action": "lambda:GetLayerVersion",
    "Resource": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1"
  }
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[AddLayerVersionPermission](#)」の「」を参照してください。

add-permission

次の例は、add-permission を使用する方法を説明しています。

AWS CLI

既存の Lambda 関数にアクセス許可を追加するには

次のadd-permission例では、という名前の関数を呼び出すアクセス許可を Amazon SNSサービスに付与しますmy-function。

```
aws lambda add-permission \
  --function-name my-function \
  --action lambda:InvokeFunction \
  --statement-id sns \
  --principal sns.amazonaws.com
```

出力:

```
{
  "Statement":
```

```
{
  "Sid": "sns",
  "Effect": "Allow",
  "Principal": {
    "Service": "sns.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "arn:aws:lambda:us-east-2:123456789012:function:my-function"
}
```

詳細については、[AWS「Lambda デベロッパーガイド」の「Lambda でのリソースベースのポリシーの使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス [AddPermission](#)」の「」を参照してください。

create-alias

次の例は、create-alias を使用する方法を説明しています。

AWS CLI

Lambda 関数のエイリアスを作成する方法

次の create-alias の例では、my-function Lambda 関数のバージョン 1 を参照する LIVE という名前のエイリアスを作成します。

```
aws lambda create-alias \
  --function-name my-function \
  --description "alias for live version of function" \
  --function-version 1 \
  --name LIVE
```

出力:

```
{
  "FunctionVersion": "1",
  "Name": "LIVE",
  "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
  "RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",
  "Description": "alias for live version of function"
```

```
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda 関数エイリアスの設定」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAlias](#)」の「」を参照してください。

create-event-source-mapping

次のコード例は、create-event-source-mapping を使用する方法を示しています。

AWS CLI

イベントソースと AWS Lambda 関数間のマッピングを作成するには

次のcreate-event-source-mapping例では、SQSキューと my-function Lambda 関数の間にマッピングを作成します。

```
aws lambda create-event-source-mapping \  
  --function-name my-function \  
  --batch-size 5 \  
  --event-source-arn arn:aws:sqs:us-west-2:123456789012:mySQSqueue
```

出力:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569284520.333,  
  "BatchSize": 5,  
  "State": "Creating",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda イベントソースマッピング」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEventSourceMapping](#)」の「」を参照してください。

create-function

次の例は、create-function を使用する方法を説明しています。

AWS CLI

Lambda 関数を作成するには

次の create-function の例では、my-function という名前の Lambda 関数を作成します。

```
aws lambda create-function \  
  --function-name my-function \  
  --runtime nodejs18.x \  
  --zip-file fileb://my-function.zip \  
  --handler my-function.handler \  
  --role arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-tges6bf4
```

my-function.zip の内容:

```
This file is a deployment package that contains your function code and any dependencies.
```

出力:

```
{  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "PFn4S+er27qk+UuZSTKEQfNKG/XNn7QJs90mJgq6oH8=",  
  "FunctionName": "my-function",  
  "CodeSize": 308,  
  "RevisionId": "873282ed-4cd3-4dc8-a069-d0c647e470c6",  
  "MemorySize": 128,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "Version": "$LATEST",  
  "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-zgur6bf4",  
  "Timeout": 3,  
  "LastModified": "2023-10-14T22:26:11.234+0000",  
  "Handler": "my-function.handler",  
  "Runtime": "nodejs18.x",  
  "Description": ""  
}
```

```
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda 関数の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFunction](#)」の「」を参照してください。

delete-alias

次のコード例は、delete-alias を使用する方法を示しています。

AWS CLI

Lambda 関数のエイリアスを削除する方法

次の delete-alias の例では、my-function Lambda 関数から LIVE という名前のエイリアスを削除します。

```
aws lambda delete-alias \  
  --function-name my-function \  
  --name LIVE
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda 関数エイリアスの設定](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAlias](#)」の「」を参照してください。

delete-event-source-mapping

次のコード例は、delete-event-source-mapping を使用する方法を示しています。

AWS CLI

イベントソースと AWS Lambda 関数間のマッピングを削除するには

次のdelete-event-source-mapping例では、SQSキューと my-function Lambda 関数間のマッピングを削除します。

```
aws lambda delete-event-source-mapping \  
  --uuid a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

出力:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569285870.271,  
  "BatchSize": 5,  
  "State": "Deleting",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda イベントソースマッピング」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEventSourceMapping](#)」の「」を参照してください。

delete-function-concurrency

次の例は、delete-function-concurrency を使用する方法を説明しています。

AWS CLI

関数から同時実行制限を削除する方法

次の delete-function-concurrency の例では、予約済みの同時実行制限を my-function 関数から削除します。

```
aws lambda delete-function-concurrency \  
  --function-name my-function
```

このコマンドでは何も出力されません。

詳細については、「AWS Lambda デベロッパーガイド」の「[Lambda 関数の同時実行数を予約する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFunctionConcurrency](#)」の「」を参照してください。

delete-function-event-invoke-config

次のコード例は、delete-function-event-invoke-config を使用する方法を示しています。

AWS CLI

非同期呼び出し設定を削除するには

次のdelete-function-event-invoke-config例では、指定された関数のGREENエイリアスの非同期呼び出し設定を削除します。

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFunctionEventInvokeConfig](#)」の「」を参照してください。

delete-function

次の例は、delete-function を使用する方法を説明しています。

AWS CLI

例 1: Lambda 関数を関数名で削除するには

次の delete-function の例では、関数名を指定して my-function という Lambda 関数を削除します。

```
aws lambda delete-function \  
  --function-name my-function
```

このコマンドでは何も出力されません。

例 2: 関数別に Lambda 関数を削除するには ARN

次のdelete-function例では、関数の を指定my-functionして、 という名前の Lambda 関数を削除しますARN。

```
aws lambda delete-function \  
  --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

このコマンドでは何も出力されません。

例 3: 部分関数で Lambda 関数を削除するには ARN

次のdelete-function例では、関数の部分 を指定my-functionして、 という名前の Lambda 関数を削除しますARN。

```
aws lambda delete-function \  
  --function-name 123456789012:function:my-function
```

このコマンドでは何も出力されません。

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda 関数の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFunction](#)」の「」を参照してください。

delete-layer-version

次のコード例は、delete-layer-version を使用する方法を示しています。

AWS CLI

Lambda レイヤーのバージョンを削除するには

次のdelete-layer-version例では、 という名前のレイヤーのバージョン 2 を削除しますmy-layer。

```
aws lambda delete-layer-version \  
  --layer-name my-layer \  
  --version-number 2
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLayerVersion](#)」の「」を参照してください。

delete-provisioned-concurrency-config

次のコード例は、delete-provisioned-concurrency-config を使用する方法を示しています。

AWS CLI

プロビジョニングされた同時実行数の設定を削除する方法

次の delete-provisioned-concurrency-config の例では、指定した関数の GREEN エイリアスのプロビジョニングされた同時実行設定を削除します。

```
aws lambda delete-provisioned-concurrency-config \  
  --function-name my-function \  
  --qualifier GREEN
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteProvisionedConcurrencyConfig](#)」の「」を参照してください。

get-account-settings

次のコード例は、get-account-settings を使用する方法を示しています。

AWS CLI

AWS リージョンのアカウントの詳細を取得するには

次の get-account-settings の例では、アカウントの Lambda 制限および使用情報を表示します。

```
aws lambda get-account-settings
```

出力:

```
{  
  "AccountLimit": {  
    "CodeSizeUnzipped": 262144000,  
    "UnreservedConcurrentExecutions": 1000,  
    "ConcurrentExecutions": 1000,  
    "CodeSizeZipped": 52428800,  
    "TotalCodeSize": 80530636800  
  },  
}
```

```
"AccountUsage": {
  "FunctionCount": 4,
  "TotalCodeSize": 9426
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda の制限](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAccountSettings](#)」の「」を参照してください。

get-alias

次の例は、get-alias を使用する方法を説明しています。

AWS CLI

関数エイリアスに関する詳細を取得する方法

次の get-alias の例では、my-function Lambda 関数に LIVE という名前のエイリアスの詳細が表示されます。

```
aws lambda get-alias \
  --function-name my-function \
  --name LIVE
```

出力:

```
{
  "FunctionVersion": "3",
  "Name": "LIVE",
  "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",
  "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",
  "Description": "alias for live version of function"
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda 関数エイリアスの設定](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAlias](#)」の「」を参照してください。

get-event-source-mapping

次の例は、get-event-source-mapping を使用する方法を説明しています。

AWS CLI

イベントソースマッピングの詳細を取得するには

次のget-event-source-mapping例では、SQSキューと my-function Lambda 関数間のマッピングの詳細を表示します。

```
aws lambda get-event-source-mapping \  
  --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

出力:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569284520.333,  
  "BatchSize": 5,  
  "State": "Enabled",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

詳細については、[AWS「Lambda デベロッパーガイド」の「Lambda イベントソースマッピング」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[GetEventSourceMapping](#)」の「」を参照してください。

get-function-concurrency

次のコード例は、get-function-concurrency を使用する方法を示しています。

AWS CLI

関数の予約済み同時実行設定を表示する方法

次の get-function-concurrency の例では、指定した関数の予約済み同時実行設定を取得します。


```
aws lambda get-function-concurrency \  
  --function-name my-function
```

出力:

```
{  
  "ReservedConcurrentExecutions": 250  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFunctionConcurrency](#)」の「」を参照してください。

get-function-configuration

次の例は、get-function-configuration を使用する方法を説明しています。

AWS CLI

Lambda 関数のバージョン固有設定を取得する方法

次の get-function-configuration の例では、my-function 関数のバージョン 2 の設定が表示されます。

```
aws lambda get-function-configuration \  
  --function-name my-function:2
```

出力:

```
{  
  "FunctionName": "my-function",  
  "LastModified": "2019-09-26T20:28:40.438+0000",  
  "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",  
  "MemorySize": 256,  
  "Version": "2",  
  "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qqq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qgzYUHaqR716pZ2dpkn/0J1FrzJm1KidWoaCgk=",  
}
```

```
"Description": "",
"VpcConfig": {
  "SubnetIds": [],
  "VpcId": "",
  "SecurityGroupIds": []
},
"CodeSize": 304,
"FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:2",
"Handler": "index.handler"
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda 関数の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFunctionConfiguration](#)」の「」を参照してください。

get-function-event-invoke-config

次の例は、get-function-event-invoke-config を使用する方法を説明しています。

AWS CLI

非同期呼び出し設定を表示するには

次のget-function-event-invoke-config例では、指定された関数のBLUEエイリアスの非同期呼び出し設定を取得します。

```
aws lambda get-function-event-invoke-config \
  --function-name my-function:BLUE
```

出力:

```
{
  "LastModified": 1577824396.653,
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
  "MaximumRetryAttempts": 0,
  "MaximumEventAgeInSeconds": 3600,
  "DestinationConfig": {
    "OnSuccess": {},
    "OnFailure": {
```

```
        "Destination": "arn:aws:sqs:us-east-2:123456789012:failed-invocations"
      }
    }
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFunctionEventInvokeConfig](#)」の「」を参照してください。

get-function

次のコード例は、get-function を使用する方法を示しています。

AWS CLI

関数に関する情報を取得するには

次の例 get-function では、my-function 関数の情報が表示されます。

```
aws lambda get-function \
  --function-name my-function
```

出力:

```
{
  "Concurrency": {
    "ReservedConcurrentExecutions": 100
  },
  "Code": {
    "RepositoryType": "S3",
    "Location": "https://awslambda-us-west-2-tasks.s3.us-west-2.amazonaws.com/snapshots/123456789012/my-function..."
  },
  "Configuration": {
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "Version": "$LATEST",
    "CodeSha256": "5tT2qqzYUHoqwR616pZ2dpkn/0J1FrzJm1KidWaaCgk=",
    "FunctionName": "my-function",
    "VpcConfig": {
      "SubnetIds": [],
      "VpcId": ""
    }
  }
}
```

```
    "SecurityGroupIds": [],
  },
  "MemorySize": 128,
  "RevisionId": "28f0fb31-5c5c-43d3-8955-03e76c5c1075",
  "CodeSize": 304,
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
  "Handler": "index.handler",
  "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-role-uy319qq",
  "Timeout": 3,
  "LastModified": "2019-09-24T18:20:35.054+0000",
  "Runtime": "nodejs10.x",
  "Description": ""
}
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda 関数の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFunction](#)」の「」を参照してください。

get-layer-version-by-arn

次の例は、get-layer-version-by-arn を使用する方法を説明しています。

AWS CLI

Lambda レイヤーバージョンに関する情報を取得するには

次のget-layer-version-by-arn例では、レイヤーバージョンと指定された Amazon リソースネーム () に関する情報を表示しますARN。

```
aws lambda get-layer-version-by-arn \
  --arn "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x:2"
```

出力:

```
{
  "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x:2",
```

```
"Description": "AWS Lambda SciPy layer for Python 3.11 (scipy-1.1.0,
numpy-1.15.4) https://github.com/scipy/scipy/releases/tag/v1.1.0 https://
github.com/numpy/numpy/releases/tag/v1.15.4",
"CreateDate": "2023-10-12T10:09:38.398+0000",
"LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-
SciPy1x",
"Content": {
  "CodeSize": 41784542,
  "CodeSha256": "GGmv8ocUw4cly0T8HL0Vx/f5V4RmSCGNjDIslY4VskM=",
  "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/
snapshots/123456789012/..."
},
"Version": 2,
"CompatibleRuntimes": [
  "python3.11"
],
"LicenseInfo": "SciPy: https://github.com/scipy/scipy/blob/main/LICENSE.txt,
NumPy: https://github.com/numpy/numpy/blob/main/LICENSE.txt"
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLayerVersionByArn](#)」の「」を参照してください。

get-layer-version-policy

次のコード例は、get-layer-version-policy を使用する方法を示しています。

AWS CLI

Lambda レイヤーバージョンのアクセス許可ポリシーを取得するには

次のget-layer-version-policy例では、 という名前のレイヤーのバージョン 1 に関するポリシー情報を表示しますmy-layer。

```
aws lambda get-layer-version-policy \
  --layer-name my-layer \
  --version-number 1
```

出力:

```
{
  "Policy": {
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
      {
        "Sid": "xaccount",
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::123456789012:root"},
        "Action": "lambda:GetLayerVersion",
        "Resource": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1"
      }
    ],
    "RevisionId": "c68f21d2-cbf0-4026-90f6-1375ee465cd0"
  }
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLayerVersionPolicy](#)」の「」を参照してください。

get-layer-version

次のコード例は、get-layer-version を使用する方法を示しています。

AWS CLI

Lambda レイヤーバージョンに関する情報を取得するには

次のget-layer-version例は、`my-layer` という名前のレイヤーのバージョン 1 に関する情報を示しています。

```
aws lambda get-layer-version \
  --layer-name my-layer \
  --version-number 1
```

出力:

```
{
```

```
"Content": {
  "Location": "https://awslambda-us-east-2-layers.s3.us-east-2.amazonaws.com/
snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?
versionId=27iWyA73cCAYqyH...",
  "CodeSha256": "tv9jJ0+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",
  "CodeSize": 169
},
"LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
"LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer:1",
"Description": "My Python layer",
"CreateDate": "2018-11-14T23:03:52.894+0000",
"Version": 1,
"LicenseInfo": "MIT",
"CompatibleRuntimes": [
  "python3.10",
  "python3.11"
]
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLayerVersion](#)」の「」を参照してください。

get-policy

次の例は、get-policy を使用する方法を説明しています。

AWS CLI

関数、バージョン、またはエイリアスのリソースベースのIAMポリシーを取得するには

次の get-policy の例では、my-function Lambda 関数に関する情報が表示されます。

```
aws lambda get-policy \
  --function-name my-function
```

出力:

```
{
  "Policy": {
    "Version": "2012-10-17",
```

```
    "Id": "default",
    "Statement":
    [
      {
        "Sid": "iot-events",
        "Effect": "Allow",
        "Principal": {"Service": "iotevents.amazonaws.com"},
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:us-west-2:123456789012:function:my-
function"
      }
    ],
    "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668"
  }
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda でのリソースベースのポリシーの使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicy](#)」の「」を参照してください。

get-provisioned-concurrency-config

次のコード例は、get-provisioned-concurrency-config を使用する方法を示しています。

AWS CLI

プロビジョニングされた同時実行設定を表示する方法

次の get-provisioned-concurrency-config の例では、指定した関数の BLUE エイリアスにプロビジョニングされた同時実行設定の詳細が表示されます。

```
aws lambda get-provisioned-concurrency-config \
  --function-name my-function \
  --qualifier BLUE
```

出力:

```
{
  "RequestedProvisionedConcurrentExecutions": 100,
  "AvailableProvisionedConcurrentExecutions": 100,
```



```
"AllocatedProvisionedConcurrentExecutions": 100,  
"Status": "READY",  
"LastModified": "2019-12-31T20:28:49+0000"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetProvisionedConcurrencyConfig](#)」の「」を参照してください。

invoke

次の例は、`invoke` を使用する方法を説明しています。

AWS CLI

例 1: Lambda 関数を同期的に呼び出すには

次の例 `invoke` では、`my-function` 関数を同期的に呼び出します。バージョン 2 を使用している場合 AWS CLI は、`cli-binary-format` オプションが必要です。詳細については、[AWS CLI 「コマンドラインインターフェイスユーザーガイド」の「サポートされているグローバルコマンドラインオプションAWS」](#) を参照してください。

```
aws lambda invoke \  
  --function-name my-function \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

出力:

```
{  
  "ExecutedVersion": "$LATEST",  
  "StatusCode": 200  
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[同期呼び出し](#)」を参照してください。

例 2: Lambda 関数を非同期で呼び出すには

次の例 `invoke` では、`my-function` 関数を非同期で呼び出します。バージョン 2 を使用している場合 AWS CLI は、`cli-binary-format` オプションが必要です。詳細については、[AWS](#)

CLI「[コマンドラインインターフェイスユーザーガイド](#)」の「[サポートされているグローバルコマンドラインオプションAWS](#)」を参照してください。

```
aws lambda invoke \  
  --function-name my-function \  
  --invocation-type Event \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

出力:

```
{  
  "StatusCode": 202  
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[非同期呼び出し](#)」を参照してください。

- API 詳細については、「[コマンドリファレンス](#)」の「[呼び出し](#)」を参照してください。AWS CLI

list-aliases

次のコード例は、list-aliases を使用する方法を示しています。

AWS CLI

Lambda 関数のエイリアスのリストを取得するには

次のlist-aliases例では、my-functionLambda 関数のエイリアスのリストを表示します。

```
aws lambda list-aliases \  
  --function-name my-function
```

出力:

```
{  
  "Aliases": [  
    {  
      "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-  
function:BETA",
```

```
    "RevisionId": "a410117f-ab16-494e-8035-7e204bb7933b",
    "FunctionVersion": "2",
    "Name": "BETA",
    "Description": "alias for beta version of function"
  },
  {
    "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:LIVE",
    "RevisionId": "21d40116-f8b1-40ba-9360-3ea284da1bb5",
    "FunctionVersion": "1",
    "Name": "LIVE",
    "Description": "alias for live version of function"
  }
]
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda 関数エイリアスの設定」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAliases](#)」の「」を参照してください。

list-event-source-mappings

次の例は、list-event-source-mappings を使用する方法を説明しています。

AWS CLI

関数のイベントソースマッピングを一覧表示するには

次のlist-event-source-mappings例では、my-functionLambda 関数のイベントソースマッピングのリストを表示します。

```
aws lambda list-event-source-mappings \
  --function-name my-function
```

出力:

```
{
  "EventSourceMappings": [
    {
      "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "StateTransitionReason": "USER_INITIATED",
```

```
    "LastModified": 1569284520.333,  
    "BatchSize": 5,  
    "State": "Enabled",  
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-  
function",  
    "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
  }  
]  
}
```

詳細については、[AWS「Lambda デベロッパーガイド」の「Lambda イベントソースマッピングAWS」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListEventSourceMappings](#)」の「」を参照してください。

list-function-event-invoke-configs

次の例は、list-function-event-invoke-configs を使用する方法を説明しています。

AWS CLI

非同期呼び出し設定のリストを表示するには

次のlist-function-event-invoke-configs例では、指定された関数の非同期呼び出し設定を一覧表示します。

```
aws lambda list-function-event-invoke-configs \  
  --function-name my-function
```

出力:

```
{  
  "FunctionEventInvokeConfigs": [  
    {  
      "LastModified": 1577824406.719,  
      "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-  
function:GREEN",  
      "MaximumRetryAttempts": 2,  
      "MaximumEventAgeInSeconds": 1800  
    },  
    {  
      "LastModified": 1577824396.653,
```

```
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-  
function:BLUE",  
    "MaximumRetryAttempts": 0,  
    "MaximumEventAgeInSeconds": 3600  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFunctionEventInvokeConfigs](#)」の「」を参照してください。

list-functions

次のコード例は、list-functions を使用する方法を示しています。

AWS CLI

Lambda 関数の一覧を取得するには

次の例 list-functions では、現在のユーザーのすべての関数を一覧表示します。

```
aws lambda list-functions
```

出力:

```
{  
  "Functions": [  
    {  
      "TracingConfig": {  
        "Mode": "PassThrough"  
      },  
      "Version": "$LATEST",  
      "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbl/RMr0fT/I=",  
      "FunctionName": "helloworld",  
      "MemorySize": 128,  
      "RevisionId": "1718e831-badf-4253-9518-d0644210af7b",  
      "CodeSize": 294,  
      "FunctionArn": "arn:aws:lambda:us-  
west-2:123456789012:function:helloworld",  
      "Handler": "helloworld.handler",  
      "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-  
role-zgur6bf4",
```

```
    "Timeout": 3,
    "LastModified": "2023-09-23T18:32:33.857+0000",
    "Runtime": "nodejs18.x",
    "Description": ""
  },
  {
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "Version": "$LATEST",
    "CodeSha256": "sU0cJ2/h0ZevwV/1TxCuQqK3gDZP3i8gUoqUUVRmY6E=",
    "FunctionName": "my-function",
    "VpcConfig": {
      "SubnetIds": [],
      "VpcId": "",
      "SecurityGroupIds": []
    },
    "MemorySize": 256,
    "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
    "CodeSize": 266,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qq",
    "Timeout": 3,
    "LastModified": "2023-10-01T16:47:28.490+0000",
    "Runtime": "nodejs18.x",
    "Description": ""
  },
  {
    "Layers": [
      {
        "CodeSize": 41784542,
        "Arn": "arn:aws:lambda:us-west-2:420165488524:layer:AWSLambda-
Python37-SciPy1x:2"
      },
      {
        "CodeSize": 4121,
        "Arn": "arn:aws:lambda:us-
west-2:123456789012:layer:pythonLayer:1"
      }
    ],
    "TracingConfig": {
```

```
        "Mode": "PassThrough"
    },
    "Version": "$LATEST",
    "CodeSha256": "ZQukCqxtkqFgyF2cU41Avj99TKQ/hNihPtDtRcc08mI=",
    "FunctionName": "my-python-function",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "MemorySize": 128,
    "RevisionId": "80b4eabc-acf7-4ea8-919a-e874c213707d",
    "CodeSize": 299,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
python-function",
    "Handler": "lambda_function.lambda_handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/my-python-function-
role-z5g7dr6n",
    "Timeout": 3,
    "LastModified": "2023-10-01T19:40:41.643+0000",
    "Runtime": "python3.11",
    "Description": ""
    }
]
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda 関数の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFunctions](#)」の「」を参照してください。

list-layer-versions

次の例は、list-layer-versions を使用する方法を説明しています。

AWS CLI

AWS Lambda レイヤーのバージョンを一覧表示するには

次のlist-layers-versions例は、 という名前のレイヤーのバージョンに関する情報を示していますmy-layer。

```
aws lambda list-layer-versions \  
  --layer-name my-layer
```

出力:

```
{  
  "Layers": [  
    {  
      "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-  
layer:2",  
      "Version": 2,  
      "Description": "My layer",  
      "CreateDate": "2023-11-15T00:37:46.592+0000",  
      "CompatibleRuntimes": [  
        "python3.10",  
        "python3.11"  
      ]  
    }  
  ]  
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLayerVersions](#)」の「」を参照してください。

list-layers

次のコード例は、list-layers を使用する方法を示しています。

AWS CLI

関数のランタイムと互換性のあるレイヤーを一覧表示するには

次のlist-layers例は、Python 3.11 ランタイムと互換性のあるレイヤーに関する情報を示しています。

```
aws lambda list-layers \  
  --compatible-runtime python3.11
```

出力:


```
{
  "Layers": [
    {
      "LayerName": "my-layer",
      "LayerArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-layer",
      "LatestMatchingVersion": {
        "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
        "Version": 2,
        "Description": "My layer",
        "CreateDate": "2023-11-15T00:37:46.592+0000",
        "CompatibleRuntimes": [
          "python3.10",
          "python3.11"
        ]
      }
    }
  ]
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLayers](#)」の「」を参照してください。

list-provisioned-concurrency-configs

次のコード例は、list-provisioned-concurrency-configs を使用する方法を示しています。

AWS CLI

プロビジョニングされた同時実行設定のリストを取得する方法

次の list-provisioned-concurrency-configs の例では、指定された関数にプロビジョニングされた同時実行設定がリストされます。

```
aws lambda list-provisioned-concurrency-configs \
  --function-name my-function
```

出力:

```
{
  "ProvisionedConcurrencyConfigs": [
    {
      "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:GREEN",
      "RequestedProvisionedConcurrentExecutions": 100,
      "AvailableProvisionedConcurrentExecutions": 100,
      "AllocatedProvisionedConcurrentExecutions": 100,
      "Status": "READY",
      "LastModified": "2019-12-31T20:29:00+0000"
    },
    {
      "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-
function:BLUE",
      "RequestedProvisionedConcurrentExecutions": 100,
      "AvailableProvisionedConcurrentExecutions": 100,
      "AllocatedProvisionedConcurrentExecutions": 100,
      "Status": "READY",
      "LastModified": "2019-12-31T20:28:49+0000"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProvisionedConcurrencyConfigs](#)」の「」を参照してください。

list-tags

次のコード例は、list-tags を使用する方法を示しています。

AWS CLI

Lambda 関数のタグのリストを取得する方法

次の list-tags の例では、my-function Lambda 関数にアタッチされたタグが表示されま

す。

```
aws lambda list-tags \
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function
```

出力:

```
{
  "Tags": {
    "Category": "Web Tools",
    "Department": "Sales"
  }
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[Lambda 関数をタグ付けする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTags](#)」の「」を参照してください。

list-versions-by-function

次のコード例は、list-versions-by-function を使用する方法を示しています。

AWS CLI

関数のバージョンのリストを取得する方法

次の list-versions-by-function の例は、my-function Lambda 関数のバージョンのリストが表示されます。

```
aws lambda list-versions-by-function \
  --function-name my-function
```

出力:

```
{
  "Versions": [
    {
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "Version": "$LATEST",
      "CodeSha256": "sU0cJ2/h0ZevwV/1TxCuQqK3gDZP3i8gUoqUUVRmY6E=",
      "FunctionName": "my-function",
      "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
      }
    }
  ]
}
```

```
    },
    "MemorySize": 256,
    "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
    "CodeSize": 266,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:$LATEST",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
    "Timeout": 3,
    "LastModified": "2019-10-01T16:47:28.490+0000",
    "Runtime": "nodejs10.x",
    "Description": ""
  },
  {
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "Version": "1",
    "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJmlKidWaaCgk=",
    "FunctionName": "my-function",
    "VpcConfig": {
      "SubnetIds": [],
      "VpcId": "",
      "SecurityGroupIds": []
    },
    "MemorySize": 256,
    "RevisionId": "949c8914-012e-4795-998c-e467121951b1",
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:1",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
    "Timeout": 3,
    "LastModified": "2019-09-26T20:28:40.438+0000",
    "Runtime": "nodejs10.x",
    "Description": "new version"
  },
  {
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "Version": "2",
```

```
    "CodeSha256": "sU0cJ2/h0ZevwV/1TxCuQqK3gDZP3i8gUoqUUVRmY6E=",
    "FunctionName": "my-function",
    "VpcConfig": {
      "SubnetIds": [],
      "VpcId": "",
      "SecurityGroupIds": []
    },
    "MemorySize": 256,
    "RevisionId": "cd669f21-0f3d-4e1c-9566-948837f2e2ea",
    "CodeSize": 266,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:2",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qq",
    "Timeout": 3,
    "LastModified": "2019-10-01T16:47:28.490+0000",
    "Runtime": "nodejs10.x",
    "Description": "newer version"
  }
]
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda 関数エイリアスの設定」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVersionsByFunction](#)」の「」を参照してください。

publish-layer-version

次の例は、publish-layer-version を使用する方法を説明しています。

AWS CLI

Lambda レイヤーバージョンを作成するには

次のpublish-layer-version例では、新しい Python ライブラリレイヤーバージョンを作成します。コマンドは、指定された S3 バケットlayer.zip内の という名前のファイルを含むレイヤーコンテンツを取得します。

```
aws lambda publish-layer-version \  
  --layer-name my-layer \  
  --code-s3-bucket my-bucket \  
  --code-s3-key my-layer.zip
```

```
--description "My Python layer" \  
--license-info "MIT" \  
--content S3Bucket=Lambda-layers-us-west-2-123456789012,S3Key=layer.zip \  
--compatible-runtimes python3.10 python3.11
```

出力:

```
{  
  "Content": {  
    "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/  
snapshots/123456789012/my-layer-4aaa2fbb-ff77-4b0a-ad92-5b78a716a96a?  
versionId=27iWyA73cCAYqyH...",  
    "CodeSha256": "tv9jJ0+rPbXUUXuRKi7CwHzKtLDkDRJLB3cC3Z/ouXo=",  
    "CodeSize": 169  
  },  
  "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer",  
  "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:my-layer:1",  
  "Description": "My Python layer",  
  "CreateDate": "2023-11-14T23:03:52.894+0000",  
  "Version": 1,  
  "LicenseInfo": "MIT",  
  "CompatibleRuntimes": [  
    "python3.10",  
    "python3.11"  
  ]  
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[PublishLayerVersion](#)」の「」を参照してください。

publish-version

次のコード例は、publish-version を使用する方法を示しています。

AWS CLI

関数の新しいバージョンを発行する方法

次の publish-version の例では、my-function Lambda 関数の新しいバージョンを発行します。

```
aws lambda publish-version \  
  --function-name my-function
```

出力:

```
{  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "dBG9m8SGdm1Ejw/JYX1hhvCrAv5TxvXsbl/RMr0fT/I=",  
  "FunctionName": "my-function",  
  "CodeSize": 294,  
  "RevisionId": "f31d3d39-cc63-4520-97d4-43cd44c94c20",  
  "MemorySize": 128,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:3",  
  "Version": "2",  
  "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-zgur6bf4",  
  "Timeout": 3,  
  "LastModified": "2019-09-23T18:32:33.857+0000",  
  "Handler": "my-function.handler",  
  "Runtime": "nodejs10.x",  
  "Description": ""  
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda 関数エイリアスの設定」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[PublishVersion](#)」の「」を参照してください。

put-function-concurrency

次の例は、put-function-concurrency を使用する方法を説明しています。

AWS CLI

関数に予約済み同時実行制限を設定する方法

次の put-function-concurrency の例では、my-function 関数に 100 個の予約済み同時実行数を設定します。

```
aws lambda put-function-concurrency \  
  --function-name my-function \  
  --reserved-concurrent-executions 100
```

出力:

```
{  
  "ReservedConcurrentExecutions": 100  
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[Lambda 関数の同時実行数を予約する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutFunctionConcurrency](#)」の「」を参照してください。

put-function-event-invoke-config

次のコード例は、put-function-event-invoke-config を使用する方法を示しています。

AWS CLI

非同期呼び出しのエラー処理を設定するには

次のput-function-event-invoke-config例では、最大イベント期間を 1 時間に設定し、指定された関数の再試行を無効にします。

```
aws lambda put-function-event-invoke-config \  
  --function-name my-function \  
  --maximum-event-age-in-seconds 3600 \  
  --maximum-retry-attempts 0
```

出力:

```
{  
  "LastModified": 1573686021.479,  
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:  
$LATEST",  
  "MaximumRetryAttempts": 0,  
  "MaximumEventAgeInSeconds": 3600,  
}
```



```
"DestinationConfig": {
  "OnSuccess": {},
  "OnFailure": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutFunctionEventInvokeConfig](#)」の「」を参照してください。

put-provisioned-concurrency-config

次のコード例は、put-provisioned-concurrency-config を使用する方法を示しています。

AWS CLI

プロビジョニングされた同時実行数を配分する方法

次の put-provisioned-concurrency-config の例では、指定した関数の BLUE エイリアスに 100 個のプロビジョニングされた同時実行数を割り当てます。

```
aws lambda put-provisioned-concurrency-config \
  --function-name my-function \
  --qualifier BLUE \
  --provisioned-concurrent-executions 100
```

出力:

```
{
  "Requested ProvisionedConcurrentExecutions": 100,
  "Allocated ProvisionedConcurrentExecutions": 0,
  "Status": "IN_PROGRESS",
  "LastModified": "2019-11-21T19:32:12+0000"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutProvisionedConcurrencyConfig](#)」の「」を参照してください。

remove-layer-version-permission

次のコード例は、remove-layer-version-permission を使用する方法を示しています。

AWS CLI

レイヤーバージョンアクセス許可を削除するには

次のremove-layer-version-permission例では、アカウントがレイヤーバージョンを設定するアクセス許可を削除します。

```
aws lambda remove-layer-version-permission \  
  --layer-name my-layer \  
  --statement-id xaccount \  
  --version-number 1
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda レイヤー」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveLayerVersionPermission](#)」の「」を参照してください。

remove-permission

次のコード例は、remove-permission を使用する方法を示しています。

AWS CLI

既存の Lambda 関数から許可を削除する方法

次の remove-permission の例では、my-function という名前の関数を呼び出す許可を削除します。

```
aws lambda remove-permission \  
  --function-name my-function \  
  --statement-id sns
```

このコマンドでは何も出力されません。

詳細については、[Lambda デベロッパーガイドの AWS 「Lambda でのリソースベースのポリシーの使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[RemovePermission](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

既存の Lambda 関数にタグを追加する方法

次の tag-resource の例では、指定した Lambda 関数に DEPARTMENT のキー名と Department A の値を持つタグを追加します。

```
aws lambda tag-resource \  
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --tags "DEPARTMENT=Department A"
```

このコマンドでは何も出力されません。

詳細については、「AWS Lambda デベロッパーガイド」の「[Lambda 関数をタグ付けする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

既存の Lambda 関数からタグを削除する方法

次の untag-resource の例では、DEPARTMENT タグというキー名のタグを my-function Lambda 関数から削除します。

```
aws lambda untag-resource \  
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --tag-keys DEPARTMENT
```

このコマンドでは何も出力されません。

詳細については、「AWS Lambda デベロッパーガイド」の「[Lambda 関数をタグ付けする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-alias

次の例は、update-alias を使用する方法を説明しています。

AWS CLI

関数エイリアスを更新する方法

次の update-alias の例では、my-function Lambda 関数のバージョン 3 を参照するように、LIVE という名前のエイリアスを更新します。

```
aws lambda update-alias \  
  --function-name my-function \  
  --function-version 3 \  
  --name LIVE
```

出力:

```
{  
  "FunctionVersion": "3",  
  "Name": "LIVE",  
  "AliasArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:LIVE",  
  "RevisionId": "594f41fb-b85f-4c20-95c7-6ca5f2a92c93",  
  "Description": "alias for live version of function"  
}
```

詳細については、[AWS 「Lambda デベロッパーガイド」の「Lambda 関数エイリアスの設定」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAlias](#)」の「」を参照してください。

update-event-source-mapping

次の例は、update-event-source-mapping を使用する方法を説明しています。

AWS CLI

イベントソースと AWS Lambda 関数間のマッピングを更新するには

次のupdate-event-source-mapping例では、指定したマッピングでバッチサイズを 8 に更新します。

```
aws lambda update-event-source-mapping \  
  --uuid "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE" \  
  --batch-size 8
```

出力:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569284520.333,  
  "BatchSize": 8,  
  "State": "Updating",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

詳細については、[AWS「Lambda デベロッパーガイド」の「Lambda イベントソースマッピング」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateEventSourceMapping](#)」の「」を参照してください。

update-function-code

次の例は、update-function-code を使用する方法を説明しています。

AWS CLI

Lambda 関数のコードを更新するには

次のupdate-function-code例では、my-function関数の未公開 (\$LATEST) バージョンのコードを、指定された zip ファイルの内容に置き換えます。

```
aws lambda update-function-code \  
  --function-name my-function \  
  --zip-file fileb://my-function.zip
```

出力:

```
{
  "FunctionName": "my-function",
  "LastModified": "2019-09-26T20:28:40.438+0000",
  "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",
  "MemorySize": 256,
  "Version": "$LATEST",
  "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qq",
  "Timeout": 3,
  "Runtime": "nodejs10.x",
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
  "Description": "",
  "VpcConfig": {
    "SubnetIds": [],
    "VpcId": "",
    "SecurityGroupIds": []
  },
  "CodeSize": 304,
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
  "Handler": "index.handler"
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda 関数の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFunctionCode](#)」の「」を参照してください。

update-function-configuration

次の例は、update-function-configuration を使用する方法を説明しています。

AWS CLI

関数の設定を変更するには

次のupdate-function-configuration例では、my-function関数の未公開 (\$LATEST) バージョンのメモリサイズを 256 MB に変更します。

```
aws lambda update-function-configuration \
```

```
--function-name my-function \  
--memory-size 256
```

出力:

```
{  
  "FunctionName": "my-function",  
  "LastModified": "2019-09-26T20:28:40.438+0000",  
  "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",  
  "MemorySize": 256,  
  "Version": "$LATEST",  
  "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9yq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",  
  "Description": "",  
  "VpcConfig": {  
    "SubnetIds": [],  
    "VpcId": "",  
    "SecurityGroupIds": []  
  },  
  "CodeSize": 304,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "Handler": "index.handler"  
}
```

詳細については、「AWS Lambda デベロッパーガイド」の「[AWS Lambda 関数の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFunctionConfiguration](#)」の「」を参照してください。

update-function-event-invoke-config

次のコード例は、update-function-event-invoke-config を使用する方法を示しています。

AWS CLI

非同期呼び出し設定を更新するには

次のupdate-function-event-invoke-config例では、指定された関数の既存の非同期呼び出し設定に障害発生時の送信先を追加します。

```
aws lambda update-function-event-invoke-config \  
  --function-name my-function \  
  --destination-config '{"OnFailure":{"Destination": "arn:aws:sqs:us-east-2:123456789012:destination"}}'
```

出力:

```
{  
  "LastModified": 1573687896.493,  
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:  
$LATEST",  
  "MaximumRetryAttempts": 0,  
  "MaximumEventAgeInSeconds": 3600,  
  "DestinationConfig": {  
    "OnSuccess": {},  
    "OnFailure": {  
      "Destination": "arn:aws:sqs:us-east-2:123456789012:destination"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFunctionEventInvokeConfig](#)」の「」を参照してください。

を使用した License Manager の例 AWS CLI

次のコード例は、License Manager AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

• [アクション](#)

アクション

create-license-configuration

次の例は、create-license-configuration を使用する方法を説明しています。

AWS CLI

例 1: ライセンス設定を作成するには

次のcreate-license-configuration例では、ハード制限が 10 コアのライセンス設定を作成します。

```
aws license-manager create-license-configuration --name my-license-configuration \  
  --license-counting-type Core \  
  --license-count 10 \  
  --license-count-hard-limit
```

出力:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE1111"  
}
```

例 2: ライセンス設定を作成するには

次のcreate-license-configuration例では、ソフト制限が 100 のライセンス設定を作成しますvCPUs。ルールを使用して vCPU 最適化を有効にします。

```
aws license-manager create-license-configuration --name my-license-configuration \  
  --license-counting-type vCPU \  
  --license-count 100 \  
  --license-rules "#honorVcpuOptimization=true"
```

出力:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE2222"  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLicenseConfiguration](#)」の「」を参照してください。

delete-license-configuration

次のコード例は、delete-license-configuration を使用する方法を示しています。

AWS CLI

ライセンス設定を削除するには

次のdelete-license-configuration例では、指定されたライセンス設定を削除します。

```
aws license-manager delete-license-configuration \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLicenseConfiguration](#)」の「」を参照してください。

get-license-configuration

次の例は、get-license-configuration を使用する方法を説明しています。

AWS CLI

ライセンス設定情報を取得するには

次のget-license-configuration例では、指定されたライセンス設定の詳細を表示します。

```
aws license-manager get-license-configuration \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

出力:

```
{  
  "LicenseConfigurationId": "lic-38b658717b87478aaa7c00883EXAMPLE",
```

```
"LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE",
  "Name": "my-license-configuration",
  "LicenseCountingType": "vCPU",
  "LicenseRules": [],
  "LicenseCountHardLimit": false,
  "ConsumedLicenses": 0,
  "Status": "AVAILABLE",
  "OwnerAccountId": "123456789012",
  "ConsumedLicenseSummaryList": [
    {
      "ResourceType": "EC2_INSTANCE",
      "ConsumedLicenses": 0
    },
    {
      "ResourceType": "EC2_HOST",
      "ConsumedLicenses": 0
    },
    {
      "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
      "ConsumedLicenses": 0
    }
  ],
  "ManagedResourceSummaryList": [
    {
      "ResourceType": "EC2_INSTANCE",
      "AssociationCount": 0
    },
    {
      "ResourceType": "EC2_HOST",
      "AssociationCount": 0
    },
    {
      "ResourceType": "EC2_AMI",
      "AssociationCount": 2
    },
    {
      "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
      "AssociationCount": 0
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLicenseConfiguration](#)」の「」を参照してください。

get-service-settings

次のコード例は、get-service-settings を使用する方法を示しています。

AWS CLI

License Manager の設定を取得するには

次のget-service-settings例は、現在のリージョンの License Manager のサービス設定を示しています。

```
aws license-manager get-service-settings
```

クロスアカウントリソース検出が無効になっている場合の出力例を次に示します。

```
{
  "OrganizationConfiguration": {
    "EnableIntegration": false
  },
  "EnableCrossAccountsDiscovery": false
}
```

クロスアカウントリソース検出が有効になっている場合の出力例を次に示します。

```
{
  "S3BucketArn": "arn:aws:s3:::aws-license-manager-service-c22d6279-35c4-47c4-bb",
  "OrganizationConfiguration": {
    "EnableIntegration": true
  },
  "EnableCrossAccountsDiscovery": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceSettings](#)」の「」を参照してください。

list-associations-for-license-configuration

次の例は、list-associations-for-license-configuration を使用する方法を説明しています。

AWS CLI

ライセンス設定の関連付けを取得するには

次のlist-associations-for-license-configuration例では、指定されたライセンス設定の関連付けに関する詳細情報を表示します。

```
aws license-manager list-associations-for-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

出力:

```
{
  "LicenseConfigurationAssociations": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0",
      "ResourceType": "EC2_AMI",
      "ResourceOwnerId": "123456789012",
      "AssociationTime": 1568825118.617
    },
    {
      "ResourceArn": "arn:aws:ec2:us-west-2::image/ami-0abcdef1234567890",
      "ResourceType": "EC2_AMI",
      "ResourceOwnerId": "123456789012",
      "AssociationTime": 1568825118.946
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAssociationsForLicenseConfiguration](#)」の「」を参照してください。

list-license-configurations

次のコード例は、list-license-configurations を使用する方法を示しています。

AWS CLI

例 1: すべてのライセンス設定を一覧表示するには

次の `list-license-configurations` 例では、すべてのライセンス設定を一覧表示します。

```
aws license-manager list-license-configurations
```

出力:

```
{
  "LicenseConfigurations": [
    {
      "LicenseConfigurationId": "lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
      "LicenseConfigurationArn": "arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE",
      "Name": "my-license-configuration",
      "LicenseCountingType": "Core",
      "LicenseRules": [],
      "LicenseCount": 10,
      "LicenseCountHardLimit": true,
      "ConsumedLicenses": 0,
      "Status": "AVAILABLE",
      "OwnerAccountId": "123456789012",
      "ConsumedLicenseSummaryList": [
        {
          "ResourceType": "EC2_INSTANCE",
          "ConsumedLicenses": 0
        },
        {
          "ResourceType": "EC2_HOST",
          "ConsumedLicenses": 0
        },
        {
          "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
          "ConsumedLicenses": 0
        }
      ],
      "ManagedResourceSummaryList": [
        {
          "ResourceType": "EC2_INSTANCE",
          "AssociationCount": 0
        }
      ]
    }
  ]
}
```

```
    {
      "ResourceType": "EC2_HOST",
      "AssociationCount": 0
    },
    {
      "ResourceType": "EC2_AMI",
      "AssociationCount": 0
    },
    {
      "ResourceType": "SYSTEMS_MANAGER_MANAGED_INSTANCE",
      "AssociationCount": 0
    }
  ]
},
{
  ...
}
]
```

例 2: 特定のライセンス設定を一覧表示するには

次のlist-license-configurations例では、指定されたライセンス設定のみを一覧表示します。

```
aws license-manager list-license-configurations \
  --license-configuration-arns arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLicenseConfigurations](#)」の「」を参照してください。

list-license-specifications-for-resource

次の例は、list-license-specifications-for-resource を使用する方法を説明しています。

AWS CLI

リソースのライセンス設定を一覧表示するには

次のlist-license-specifications-for-resource例では、指定された Amazon マシンイメージ () に関連付けられたライセンス設定を一覧表示しますAMI。

```
aws license-manager list-license-specifications-for-resource \  
--resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0
```

出力:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンスListLicenseSpecificationsForResource」の「」を参照してください。

list-resource-inventory

次のコード例は、list-resource-inventory を使用する方法を示しています。

AWS CLI

リソースインベントリ内のリソースを一覧表示するには

次のlist-resource-inventory例では、Systems Manager インベントリを使用して管理されるリソースを一覧表示します。

```
aws license-manager list-resource-inventory
```

出力:

```
{  
  "ResourceInventoryList": [  
    {  
      "Platform": "Red Hat Enterprise Linux Server",  
      "ResourceType": "EC2Instance",  
      "PlatformVersion": "7.4",  
      "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/  
i-05d3cdfb05bd36376",  
      "ResourceId": "i-05d3cdfb05bd36376",  
      "ResourceOwningAccountId": "1234567890129"  
    }  
  ]  
}
```



```
    },
    {
      "Platform": "Amazon Linux",
      "ResourceType": "EC2Instance",
      "PlatformVersion": "2",
      "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0b1d036cfd4594808",
      "ResourceId": "i-0b1d036cfd4594808",
      "ResourceOwningAccountId": "1234567890129"
    },
    {
      "Platform": "Microsoft Windows Server 2019 Datacenter",
      "ResourceType": "EC2Instance",
      "PlatformVersion": "10.0.17763",
      "ResourceArn": "arn:aws:ec2:us-west-2:1234567890129:instance/
i-0cdb3b54a2a8246ad",
      "ResourceId": "i-0cdb3b54a2a8246ad",
      "ResourceOwningAccountId": "1234567890129"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceInventory](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

ライセンス設定のタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたライセンス設定のタグを一覧表示します。

```
aws license-manager list-tags-for-resource \
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

出力:

```
{
```

```
    "Tags": [
      {
        "Key": "project",
        "Value": "lima"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-usage-for-license-configuration

次のコード例は、list-usage-for-license-configuration を使用する方法を示しています。

AWS CLI

ライセンス設定で使用されているライセンスを一覧表示するには

次のlist-usage-for-license-configuration例は、指定されたライセンス設定のライセンスを使用するリソースに関する情報を示しています。例えば、ライセンスタイプが v の場合 CPU、インスタンスは v ごとに 1 つのライセンスを使用しますCPU。

```
aws license-manager list-usage-for-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

出力:

```
{
  "LicenseConfigurationUsageList": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-04a636d18e83cfacb",
      "ResourceType": "EC2_INSTANCE",
      "ResourceStatus": "running",
      "ResourceOwnerId": "123456789012",
      "AssociationTime": 1570892850.519,
      "ConsumedLicenses": 2
    }
  ]
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUsageForLicenseConfiguration](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

ライセンス設定にタグを追加するには

次のtag-resource例では、指定されたタグ (キー名と値) を指定されたライセンス設定に追加します。

```
aws license-manager tag-resource \  
  --tags Key=project,Value=Lima \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

ライセンス設定からタグを削除するには

次のuntag-resource例では、指定されたライセンス設定から指定されたタグ (キー名とリソース) を削除します。

```
aws license-manager untag-resource \  
  --tag-keys project \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-license-configuration

次の例は、update-license-configuration を使用する方法を説明しています。

AWS CLI

ライセンス設定を更新するには

次のupdate-license-configuration例では、指定されたライセンス設定を更新してハード制限を削除します。

```
aws license-manager update-license-configuration \  
  --no-license-count-hard-limit \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

このコマンドでは何も出力されません。

次のupdate-license-configuration例では、指定されたライセンス設定を更新して、ステータスを `DISABLED` に変更します。

```
aws license-manager update-license-configuration \  
  --license-configuration-status DISABLED \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:880185128111:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLicenseConfiguration](#)」の「」を参照してください。

update-license-specifications-for-resource

次の例は、update-license-specifications-for-resource を使用する方法を説明しています。

AWS CLI

リソースのライセンス設定を更新するには

次のupdate-license-specifications-for-resource例では、1つのライセンス設定を削除して別のライセンス設定を追加することで、指定された Amazon マシンイメージ (AMI) に関連付けられたライセンス設定を置き換えます。

```
aws license-manager update-license-specifications-for-resource \  
  --resource-arn arn:aws:ec2:us-west-2:image/ami-1234567890abcdef0 \  
  --remove-license-specifications LicenseConfigurationArn=arn:aws:license-  
manager:us-west-2:123456789012:License-  
configuration:lic-38b658717b87478aaa7c00883EXAMPLE \  
  --add-license-specifications LicenseConfigurationArn=arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-42b6deb06e5399a980d555927EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateLicenseSpecificationsForResource](#)」の「」を参照してください。

update-service-settings

次の例は、update-service-settings を使用する方法を説明しています。

AWS CLI

License Manager の設定を更新するには

次のupdate-service-settings例では、現在の AWS リージョンで License Manager のクロスアカウントリソース検出を有効にします。Amazon S3 バケットは、Systems Manager インベントリに必要なリソースデータ同期です。

```
aws license-manager update-service-settings \  
  --organization-configuration EnableIntegration=true \  
  --enable-cross-accounts-discovery \  
  --s3-bucket-arn arn:aws:s3::aws-license-manager-service-abcd1234EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateServiceSettings](#)」の「」を参照してください。

を使用した Lightsail の例 AWS CLI

次のコード例は、Lightsail AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

allocate-static-ip

次の例は、allocate-static-ip を使用する方法を説明しています。

AWS CLI

静的 IP を作成するには

次のallocate-static-ip例では、インスタンスにアタッチできる指定された静的 IP を作成します。

```
aws lightsail allocate-static-ip \  
  --static-ip-name StaticIp-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "b5d06d13-2f19-4683-889f-dEXAMPLEed79",  
      "resourceName": "StaticIp-1",  
      "resourceType": "StaticIp",  
      "createdAt": 1571071325.076,  
      "location": {
```

```
        "availabilityZone": "all",
        "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "AllocateStaticIp",
    "status": "Succeeded",
    "statusChangedAt": 1571071325.274
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AllocateStaticIp](#)」の「」を参照してください。

attach-disk

次のコード例は、attach-disk を使用する方法を示しています。

AWS CLI

ブロックストレージディスクをインスタンスにアタッチするには

次のattach-disk例では、のディスクパスWordPress_Multisite-1を使用してディスクをインスタンスDisk-1にアタッチします。 /dev/xvdf

```
aws lightsail attach-disk \  
  --disk-name Disk-1 \  
  --disk-path /dev/xvdf \  
  --instance-name WordPress_Multisite-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "10a08267-19ce-43be-b913-6EXAMPLE7e80",  
      "resourceName": "Disk-1",  
      "resourceType": "Disk",  
      "createdAt": 1571071465.472,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      }  
    }  
  ]  
}
```

```

    },
    "isTerminal": false,
    "operationDetails": "WordPress_Multisite-1",
    "operationType": "AttachDisk",
    "status": "Started",
    "statusChangedAt": 1571071465.472
  },
  {
    "id": "2912c477-5295-4539-88c9-bEXAMPLEd1f0",
    "resourceName": "WordPress_Multisite-1",
    "resourceType": "Instance",
    "createdAt": 1571071465.474,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "Disk-1",
    "operationType": "AttachDisk",
    "status": "Started",
    "statusChangedAt": 1571071465.474
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachDisk](#)」の「」を参照してください。

attach-instances-to-load-balancer

次の例は、attach-instances-to-load-balancer を使用する方法を説明しています。

AWS CLI

ロードバランサーにインスタンスをアタッチするには

次のattach-instances-to-load-balancer例では、インスタンス MEAN-1、MEAN-2、をロードバランサー MEAN-3にアタッチしますLoadBalancer-1。

```

aws lightsail attach-instances-to-load-balancer \
  --instance-names {"MEAN-1","MEAN-2","MEAN-3"} \
  --load-balancer-name LoadBalancer-1

```


出力:

```
{
  "operations": [
    {
      "id": "8055d19d-abb2-40b9-b527-1EXAMPLE3c7b",
      "resourceName": "LoadBalancer-1",
      "resourceType": "LoadBalancer",
      "createdAt": 1571071699.892,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "MEAN-2",
      "operationType": "AttachInstancesToLoadBalancer",
      "status": "Started",
      "statusChangedAt": 1571071699.892
    },
    {
      "id": "c35048eb-8538-456a-a118-0EXAMPLEfb73",
      "resourceName": "MEAN-2",
      "resourceType": "Instance",
      "createdAt": 1571071699.887,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "LoadBalancer-1",
      "operationType": "AttachInstancesToLoadBalancer",
      "status": "Started",
      "statusChangedAt": 1571071699.887
    },
    {
      "id": "910d09e0-adc5-4372-bc2e-0EXAMPLEd891",
      "resourceName": "LoadBalancer-1",
      "resourceType": "LoadBalancer",
      "createdAt": 1571071699.882,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
```

```
    "operationDetails": "MEAN-3",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.882
  },
  {
    "id": "178b18ac-43e8-478c-9bed-1EXAMPLE4755",
    "resourceName": "MEAN-3",
    "resourceType": "Instance",
    "createdAt": 1571071699.901,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "LoadBalancer-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.901
  },
  {
    "id": "fb62536d-2a98-4190-a6fc-4EXAMPLE7470",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571071699.885,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "MEAN-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Started",
    "statusChangedAt": 1571071699.885
  },
  {
    "id": "787dac0d-f98d-46c3-8571-3EXAMPLE5a85",
    "resourceName": "MEAN-1",
    "resourceType": "Instance",
    "createdAt": 1571071699.901,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
  },
```

```

        "isTerminal": false,
        "operationDetails": "LoadBalancer-1",
        "operationType": "AttachInstancesToLoadBalancer",
        "status": "Started",
        "statusChangedAt": 1571071699.901
    }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachInstancesToLoadBalancer](#)」の「」を参照してください。

attach-load-balancer-tls-certificate

次のコード例は、attach-load-balancer-tls-certificate を使用する方法を示しています。

AWS CLI

TLS証明書をロードバランサーにアタッチするには

次のattach-load-balancer-tls-certificate例では、ロードバランサーTLS証明書をロードバランサー Certificate2にアタッチしますLoadBalancer-1。

```

aws lightsail attach-load-balancer-tls-certificate \
  --certificate-name Certificate2 \
  --load-balancer-name LoadBalancer-1

```

出力:

```

{
  "operations": [
    {
      "id": "cf1ad6e3-3cbb-4b8a-a7f2-3EXAMPLEa118",
      "resourceName": "LoadBalancer-1",
      "resourceType": "LoadBalancer",
      "createdAt": 1571072255.416,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "Certificate2",
    }
  ]
}

```

```
    "operationType": "AttachLoadBalancerTlsCertificate",
    "status": "Succeeded",
    "statusChangedAt": 1571072255.416
  },
  {
    "id": "dae1bcfb-d531-4c06-b4ea-bEXAMPLEc04e",
    "resourceName": "Certificate2",
    "resourceType": "LoadBalancerTlsCertificate",
    "createdAt": 1571072255.416,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "LoadBalancer-1",
    "operationType": "AttachLoadBalancerTlsCertificate",
    "status": "Succeeded",
    "statusChangedAt": 1571072255.416
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachLoadBalancerTlsCertificate](#)」の「」を参照してください。

attach-static-ip

次のコード例は、attach-static-ip を使用する方法を示しています。

AWS CLI

静的 IP をインスタンスにアタッチするには

次のattach-static-ip例では、静的 IP をインスタンス StaticIp-1 にアタッチしますMEAN-1。

```
aws lightsail attach-static-ip \
  --static-ip-name StaticIp-1 \
  --instance-name MEAN-1
```

出力:

```
{
  "operations": [
    {
      "id": "45e6fa13-4808-4b8d-9292-bEXAMPLE20b2",
      "resourceName": "StaticIp-1",
      "resourceType": "StaticIp",
      "createdAt": 1571072569.375,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "MEAN-1",
      "operationType": "AttachStaticIp",
      "status": "Succeeded",
      "statusChangedAt": 1571072569.375
    },
    {
      "id": "9ee09a17-863c-4e51-8a6d-3EXAMPLE5475",
      "resourceName": "MEAN-1",
      "resourceType": "Instance",
      "createdAt": 1571072569.376,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "StaticIp-1",
      "operationType": "AttachStaticIp",
      "status": "Succeeded",
      "statusChangedAt": 1571072569.376
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachStaticIp](#)」の「」を参照してください。

close-instance-public-ports

次の例は、close-instance-public-ports を使用する方法を説明しています。

AWS CLI

インスタンスのファイアウォールポートを閉じるには

次のclose-instance-public-ports例では、インスタンス 22のTCPポートを閉じま
すMEAN-2。

```
aws lightsail close-instance-public-ports \  
  --instance-name MEAN-2 \  
  --port-info fromPort=22,protocol=TCP,toPort=22
```

出力:

```
{  
  "operation": {  
    "id": "4f328636-1c96-4649-ae6d-1EXAMPLEf446",  
    "resourceName": "MEAN-2",  
    "resourceType": "Instance",  
    "createdAt": 1571072845.737,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": true,  
    "operationDetails": "22/tcp",  
    "operationType": "CloseInstancePublicPorts",  
    "status": "Succeeded",  
    "statusChangedAt": 1571072845.737  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CloseInstancePublicPorts](#)」の「」を参照してください。

copy-snapshot

次の例は、copy-snapshot を使用する方法を説明しています。

AWS CLI

例 1: 同じ AWS リージョン内でスナップショットをコピーするには

次のcopy-snapshot例では、インスタンススナップショットを同じ AWS リージョン MEAN-1-Copy内のインスタンススナップショットMEAN-1-1571075291としてコピーしますus-west-2。

```
aws lightsail copy-snapshot \  
  --source-snapshot-name MEAN-1-1571075291 \  
  --target-snapshot-name MEAN-1-Copy \  
  --source-region us-west-2
```

出力:

```
{  
  "operations": [  
    {  
      "id": "ced16fc1-f401-4556-8d82-1EXAMPLEb982",  
      "resourceName": "MEAN-1-Copy",  
      "resourceType": "InstanceSnapshot",  
      "createdAt": 1571075581.498,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "us-west-2:MEAN-1-1571075291",  
      "operationType": "CopySnapshot",  
      "status": "Started",  
      "statusChangedAt": 1571075581.498  
    }  
  ]  
}
```

詳細については、[「Amazon Lightsail の 1 つの AWS リージョンから別のリージョンへのスナップショットのコピー」](#)を参照してください。

例 2: あるリージョンから別の AWS リージョンにスナップショットをコピーするには

次のcopy-snapshot例では、インスタンススナップショットをインスタンススナップショットMEAN-1-1571075291として AWS リージョンMEAN-1-1571075291-Copyから にコピーus-west-2しますus-east-1。

```
aws lightsail copy-snapshot \  
  --source-snapshot-name MEAN-1-1571075291 \  
  --target-snapshot-name MEAN-1-1571075291-Copy \  
  --source-region us-east-1 \  
  --target-region us-west-2
```

```
--target-snapshot-name MEAN-1-1571075291-Copy \  
--source-region us-west-2 \  
--region us-east-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "91116b79-119c-4451-b44a-dEXAMPLEd97b",  
      "resourceName": "MEAN-1-1571075291-Copy",  
      "resourceType": "InstanceSnapshot",  
      "createdAt": 1571075695.069,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-east-1"  
      },  
      "isTerminal": false,  
      "operationDetails": "us-west-2:MEAN-1-1571075291",  
      "operationType": "CopySnapshot",  
      "status": "Started",  
      "statusChangedAt": 1571075695.069  
    }  
  ]  
}
```

詳細については、[「Amazon Lightsail の 1 つの AWS リージョンから別のリージョンへのスナップショットのコピー」](#)を参照してください。

例 3: 同じ AWS リージョン内で自動スナップショットをコピーするには

次のcopy-snapshot例では2019-10-14、インスタンスの自動スナップショットを AWS リージョン WordPress-1-10142019の手動スナップショットWordPress-1としてコピーしますus-west-2。

```
aws lightsail copy-snapshot \  
--source-resource-name WordPress-1 \  
--restore-date 2019-10-14 \  
--target-snapshot-name WordPress-1-10142019 \  
--source-region us-west-2
```

出力:


```
{
  "operations": [
    {
      "id": "be3e6754-cd1d-48e6-ad9f-2EXAMPLE1805",
      "resourceName": "WordPress-1-10142019",
      "resourceType": "InstanceSnapshot",
      "createdAt": 1571082412.311,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "us-west-2:WordPress-1",
      "operationType": "CopySnapshot",
      "status": "Started",
      "statusChangedAt": 1571082412.311
    }
  ]
}
```

詳細については、[「Amazon Lightsail でインスタンスまたはディスクの自動スナップショットを保持する」](#)を参照してください。

例 4: あるリージョンから別の AWS リージョンに自動スナップショットをコピーするには

次のcopy-snapshot例では2019-10-14、インスタンスの自動スナップショットを AWS リージョンWordPress-1-10142019から us-west-2に手動スナップショットWordPress-1としてコピーしますus-east-1。

```
aws lightsail copy-snapshot \
  --source-resource-name WordPress-1 \
  --restore-date 2019-10-14 \
  --target-snapshot-name WordPress-1-10142019 \
  --source-region us-west-2 \
  --region us-east-1
```

出力:

```
{
  "operations": [
    {
      "id": "dffa128b-0b07-476e-b390-bEXAMPLE3775",
```

```
    "resourceName": "WordPress-1-10142019",
    "resourceType": "InstanceSnapshot",
    "createdAt": 1571082493.422,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-east-1"
    },
    "isTerminal": false,
    "operationDetails": "us-west-2:WordPress-1",
    "operationType": "CopySnapshot",
    "status": "Started",
    "statusChangedAt": 1571082493.422
  }
]
}
```

詳細については、[「Amazon Lightsail でインスタンスまたはディスクの自動スナップショットを保持する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopySnapshot](#)」の「」を参照してください。

create-disk-from-snapshot

次のコード例は、`create-disk-from-snapshot` を使用する方法を示しています。

AWS CLI

ディスクスナップショットからディスクを作成するには

次の`create-disk-from-snapshot`例では、指定されたブロックストレージディスクスナップショット `Disk-2` から という名前のブロックストレージディスクを作成します。ディスクは、指定された AWS リージョンとアベイラビリティゾーンに 32 GB のストレージスペースで作成されます。

```
aws lightsail create-disk-from-snapshot \  
  --disk-name Disk-2 \  
  --disk-snapshot-name Disk-1-1566839161 \  
  --availability-zone us-west-2a \  
  --size-in-gb 32
```

出力:

```
{
  "operations": [
    {
      "id": "d42b605d-5ef1-4b4a-8791-7a3e8b66b5e7",
      "resourceName": "Disk-2",
      "resourceType": "Disk",
      "createdAt": 1569624941.471,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "CreateDiskFromSnapshot",
      "status": "Started",
      "statusChangedAt": 1569624941.791
    }
  ]
}
```

詳細については、[Amazon Lightsail のスナップショットからブロックストレージディスクを作成する](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDiskFromSnapshot](#)」の「」を参照してください。

create-disk-snapshot

次の例は、create-disk-snapshot を使用する方法を説明しています。

AWS CLI

例 1: ディスクのスナップショットを作成するには

次のcreate-disk-snapshot例では、指定されたブロックストレージディスクの という名前DiskSnapshot-1のスナップショットを作成します。

```
aws lightsail create-disk-snapshot \
  --disk-name Disk-1 \
  --disk-snapshot-name DiskSnapshot-1
```

出力:

```
{
  "operations": [
    {
      "id": "fa74c6d2-03a3-4f42-a7c7-792f124d534b",
      "resourceName": "DiskSnapshot-1",
      "resourceType": "DiskSnapshot",
      "createdAt": 1569625129.739,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "Disk-1",
      "operationType": "CreateDiskSnapshot",
      "status": "Started",
      "statusChangedAt": 1569625129.739
    },
    {
      "id": "920a25df-185c-4528-87cd-7b85f5488c06",
      "resourceName": "Disk-1",
      "resourceType": "Disk",
      "createdAt": 1569625129.739,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "DiskSnapshot-1",
      "operationType": "CreateDiskSnapshot",
      "status": "Started",
      "statusChangedAt": 1569625129.739
    }
  ]
}
```

例 2: インスタンスのシステムディスクのスナップショットを作成するには

次のcreate-disk-snapshot例では、指定されたインスタンスのシステムディスクのスナップショットを作成します。

```
aws lightsail create-disk-snapshot \
  --instance-name WordPress-1 \
```

```
--disk-snapshot-name SystemDiskSnapshot-1
```

出力:

```
{
  "operations": [
    {
      "id": "f508cf1c-6597-42a6-a4c3-4aebd75af0d9",
      "resourceName": "SystemDiskSnapshot-1",
      "resourceType": "DiskSnapshot",
      "createdAt": 1569625294.685,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "WordPress-1",
      "operationType": "CreateDiskSnapshot",
      "status": "Started",
      "statusChangedAt": 1569625294.685
    },
    {
      "id": "0bb9f712-da3b-4d99-b508-3bf871d989e5",
      "resourceName": "WordPress-1",
      "resourceType": "Instance",
      "createdAt": 1569625294.685,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationDetails": "SystemDiskSnapshot-1",
      "operationType": "CreateDiskSnapshot",
      "status": "Started",
      "statusChangedAt": 1569625294.685
    }
  ]
}
```

詳細については、[「Amazon Lightsail のスナップショット」](#)と[Amazon Lightsail のインスタンスルートボリュームのスナップショットの作成](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDiskSnapshot](#)」の「」を参照してください。

create-disk

次のコード例は、create-disk を使用する方法を示しています。

AWS CLI

ブロックストレージディスクを作成するには

次のcreate-disk例では、指定された AWS リージョンとアベイラビリティゾーンDisk-1に 32 GB のストレージ容量を持つブロックストレージディスクを作成します。

```
aws lightsail create-disk \  
  --disk-name Disk-1 \  
  --availability-zone us-west-2a \  
  --size-in-gb 32
```

出力:

```
{  
  "operations": [  
    {  
      "id": "1c85e2ec-86ba-4697-b936-77f4d3dc013a",  
      "resourceName": "Disk-1",  
      "resourceType": "Disk",  
      "createdAt": 1569449220.36,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "CreateDisk",  
      "status": "Started",  
      "statusChangedAt": 1569449220.588  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDisk](#)」の「」を参照してください。

create-domain-entry

次のコード例は、create-domain-entry を使用する方法を示しています。

AWS CLI

ドメインエントリを作成するには (DNS レコード)

次のcreate-domain-entry例では、インスタンスの IP アドレスを指す指定されたドメインの頂点のDNSレコード (A) を作成します。

注 : Lightsail のドメイン関連のAPIオペレーションは、 us-east-1リージョンでのみ使用できます。CLI プロファイルが異なるリージョンを使用するように設定されている場合は、 --region us-east-1パラメータを含める必要があります。そうしないと、コマンドは失敗します。

```
aws lightsail create-domain-entry \  
  --region us-east-1 \  
  --domain-name example.com \  
  --domain-entry name=example.com,type=A,target=192.0.2.0
```

出力:

```
{  
  "operation": {  
    "id": "5be4494d-56f4-41fc-8730-693dcd0ef9e2",  
    "resourceName": "example.com",  
    "resourceType": "Domain",  
    "createdAt": 1569865296.519,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "global"  
    },  
    "isTerminal": true,  
    "operationType": "CreateDomainEntry",  
    "status": "Succeeded",  
    "statusChangedAt": 1569865296.519  
  }  
}
```

詳細については、[DNSAmazon Lightsail デベロッパーガイド](#)の[Amazon Lightsail でのドメイン DNSレコードを管理するDNSゾーンの作成](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDomainEntry](#)」の「」を参照してください。

create-domain

次の例は、create-domain を使用する方法を説明しています。

AWS CLI

ドメインを作成するには (DNS ゾーン)

次のcreate-domain例では、指定されたドメインのDNSゾーンを作成します。

注： Lightsail のドメイン関連のAPIオペレーションは、 us-east-1リージョンでのみ使用できます。CLI プロファイルが異なるリージョンを使用するように設定されている場合は、 --region us-east-1パラメータを含める必要があります。そうしないと、コマンドは失敗します。

```
aws lightsail create-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "operation": {  
    "id": "64e522c8-9ae1-4c05-9b65-3f237324dc34",  
    "resourceName": "example.com",  
    "resourceType": "Domain",  
    "createdAt": 1569864291.92,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "global"  
    },  
    "isTerminal": true,  
    "operationType": "CreateDomain",  
    "status": "Succeeded",  
    "statusChangedAt": 1569864292.109  
  }  
}
```


詳細については、[DNSAmazon Lightsail デベロッパーガイド](#)の[Amazon Lightsail でのドメイン DNSレコードを管理するDNSゾーンの作成](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDomain](#)」の「」を参照してください。

create-instance-snapshot

次のコード例は、create-instance-snapshot を使用する方法を示しています。

AWS CLI

インスタンスのスナップショットを作成するには

次のcreate-instance-snapshot例では、指定されたインスタンスからスナップショットを作成します。

```
aws lightsail create-instance-snapshot \  
  --instance-name WordPress-1 \  
  --instance-snapshot-name WordPress-Snapshot-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "4c3db559-9dd0-41e7-89c0-2cb88c19786f",  
      "resourceName": "WordPress-Snapshot-1",  
      "resourceType": "InstanceSnapshot",  
      "createdAt": 1569866438.48,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "WordPress-1",  
      "operationType": "CreateInstanceSnapshot",  
      "status": "Started",  
      "statusChangedAt": 1569866438.48  
    },  
    {  
      "id": "c04fdc45-2981-488c-88b5-d6d2fd759a6a",
```

```
    "resourceName": "WordPress-1",
    "resourceType": "Instance",
    "createdAt": 1569866438.48,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "WordPress-Snapshot-1",
    "operationType": "CreateInstanceSnapshot",
    "status": "Started",
    "statusChangedAt": 1569866438.48
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInstanceSnapshot](#)」の「」を参照してください。

create-instances-from-snapshot

次のコード例は、create-instances-from-snapshot を使用する方法を示しています。

AWS CLI

スナップショットからインスタンスを作成するには

次のcreate-instances-from-snapshot例では、\$12 USDバンドルを使用して、指定されたAWS リージョンとアベイラビリティゾーンで、指定されたインスタンススナップショットからインスタンスを作成します。

注： 指定するバンドルは、スナップショットの作成に使用される元のソースインスタンスのバンドルと同等か、それ以上である必要があります。

```
aws lightsail create-instances-from-snapshot \
  --instance-snapshot-name WordPress-1-1569866208 \
  --instance-names WordPress-2 \
  --availability-zone us-west-2a \
  --bundle-id small_3_0
```

出力:

```
{
  "operations": [
    {
      "id": "003f8271-b711-464d-b9b8-7f3806cb496e",
      "resourceName": "WordPress-2",
      "resourceType": "Instance",
      "createdAt": 1569865914.908,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "CreateInstancesFromSnapshot",
      "status": "Started",
      "statusChangedAt": 1569865914.908
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInstancesFromSnapshot](#)」の「」を参照してください。

create-instances

次の例は、create-instances を使用する方法を説明しています。

AWS CLI

例 1: 単一のインスタンスを作成するには

次のcreate-instances例では、WordPress ブループリントと \$5.00 USDバンドルを使用して、指定された AWS リージョンとアベイラビリティゾーンにインスタンスを作成します。

```
aws lightsail create-instances \
  --instance-names Instance-1 \
  --availability-zone us-west-2a \
  --blueprint-id wordpress \
  --bundle-id nano_3_0
```

出力:

```
{
```

```

"operations": [
  {
    "id": "9a77158f-7be3-4d6d-8054-cf5ae2b720cc",
    "resourceName": "Instance-1",
    "resourceType": "Instance",
    "createdAt": 1569447986.061,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "CreateInstance",
    "status": "Started",
    "statusChangedAt": 1569447986.061
  }
]
}

```

例 2: 一度に複数のインスタンスを作成するには

次のcreate-instances例では、WordPress 設計図と \$5.00 USDバンドルを使用して、指定された AWS リージョンとアベイラビリティゾーンに 3 つのインスタンスを作成します。

```

aws lightsail create-instances \
  --instance-names {"Instance1","Instance2","Instance3"} \
  --availability-zone us-west-2a \
  --blueprint-id wordpress \
  --bundle-id nano_3_0

```

出力:

```

{
  "operations": [
    {
      "id": "5492f015-9d2e-48c6-8eea-b516840e6903",
      "resourceName": "Instance1",
      "resourceType": "Instance",
      "createdAt": 1569448780.054,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
    },
  ],
}

```

```
    "isTerminal": false,
    "operationType": "CreateInstance",
    "status": "Started",
    "statusChangedAt": 1569448780.054
  },
  {
    "id": "c58b5f46-2676-44c8-b95c-3ad375898515",
    "resourceName": "Instance2",
    "resourceType": "Instance",
    "createdAt": 1569448780.054,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "CreateInstance",
    "status": "Started",
    "statusChangedAt": 1569448780.054
  },
  {
    "id": "a5ad8006-9bee-4499-9eb7-75e42e6f5882",
    "resourceName": "Instance3",
    "resourceType": "Instance",
    "createdAt": 1569448780.054,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "CreateInstance",
    "status": "Started",
    "statusChangedAt": 1569448780.054
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateInstances](#)」の「」を参照してください。

create-key-pair

次の例は、create-key-pair を使用する方法を説明しています。

AWS CLI

キーペアを作成するには

次のcreate-key-pair例では、インスタンスの認証と接続に使用できるキーペアを作成します。

```
aws lightsail create-key-pair \
  --key-pair-name MyPersonalKeyPair
```

出力はプライベートキー base64 値を提供します。この値は、作成したキーペアを使用するインスタンスへの認証に使用できます。注：プライベートキー base64 値は後で取得できないため、安全な場所にコピーして貼り付けます。

```
{
  "keyPair": {
    "name": "MyPersonalKeyPair",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:KeyPair/55025c71-198f-403b-b42f-a69433e724fb",
    "supportCode": "621291663362/MyPersonalKeyPair",
    "createdAt": 1569866556.567,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "KeyPair"
  },
  "publicKeyBase64": "ssh-rsa ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCV0xUEwX96amPERH7K1bVT1tTF190mNk6o7m5YVhk9x10dMbDRbFvhtXvw4jz
+BUHgedGUXno6uF7agqxZN01kPLJBIVTW26SSYBJ0tE
+y804UyVsjrBUqCaMXDhmfXpWuLMPwuXhwcKh7e8hwoTfkiX0E6Q1
+KqF/MiA3w6DCjEqvvdI07SiEZJFsuGNfYDDN3w60Re15MUhmn30Jdn4y/
A7Nwb3IxL4pPVE4rgFRKU8n1jp9kwRnLVMVB0WuGXk6n+H6M2f1 ",
  "privateKeyBase64": "-----BEGIN RSA PRIVATE KEY-----
EXAMPLETCCAfICCD6m7oRw0uX0jANBgqhkKiG9w0BAQUFADCBiDELMAkGA1UEBhMC
\nVVMxCzAJBgNVBAgTAldBMRawDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6\nnb24xFDASBgNVBAwTC01BTSBD
\nBgqhkKiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
\nMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMVVMxCzAJBgNVBAgTAldBMRawDgYD
\nVQQHEwdTZWF0dGx1MQ8wDQEXAMPLEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
\nb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAdBgqhkKiG9w0BCQEWEG5vb251QGft
\nYXpvbi5jb20wZGZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEXAMPLE4GmWIWJ
\n21uUSfwfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
\nrDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
```

```

\nIbb30hjZnzcvcQAaREXAMPLEMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4\nnUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
\nFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780EXAMPLELvJx79LjSTb
\nNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=\n-----END RSA PRIVATE KEY-----",
  "operation": {
    "id": "67f984db-9994-45fe-ad38-59bafcaf82ef",
    "resourceName": "MyPersonalKeyPair",
    "resourceType": "KeyPair",
    "createdAt": 1569866556.567,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "CreateKeyPair",
    "status": "Succeeded",
    "statusChangedAt": 1569866556.704
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateKeyPair](#)」の「」を参照してください。

create-load-balancer-tls-certificate

次のコード例は、create-load-balancer-tls-certificate を使用する方法を示しています。

AWS CLI

ロードバランサーのTLS証明書を作成するには

次のcreate-load-balancer-tls-certificate例では、指定されたロードバランサーにアタッチされたTLS証明書を作成します。作成された証明書は、指定されたドメインに適用されます。注：ロードバランサーには2つの証明書のみを作成できます。

```

aws lightsail create-load-balancer-tls-certificate \
  --certificate-alternative-names abc.example.com \
  --certificate-domain-name example.com \
  --certificate-name MySecondCertificate \
  --load-balancer-name MyFirstLoadBalancer

```

出力:

```
{
  "operations": [
    {
      "id": "be663aed-cb46-41e2-9b23-e2f747245bd4",
      "resourceName": "MySecondCertificate",
      "resourceType": "LoadBalancerTlsCertificate",
      "createdAt": 1569867364.971,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "MyFirstLoadBalancer",
      "operationType": "CreateLoadBalancerTlsCertificate",
      "status": "Succeeded",
      "statusChangedAt": 1569867365.219
    },
    {
      "id": "f3dfa930-969e-41cc-ac7d-337178716f6d",
      "resourceName": "MyFirstLoadBalancer",
      "resourceType": "LoadBalancer",
      "createdAt": 1569867364.971,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "MySecondCertificate",
      "operationType": "CreateLoadBalancerTlsCertificate",
      "status": "Succeeded",
      "statusChangedAt": 1569867365.219
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoadBalancerTlsCertificate](#)」の「」を参照してください。

create-load-balancer

次のコード例は、create-load-balancer を使用する方法を示しています。

AWS CLI

ロードバランサーを作成するには

次のcreate-load-balancer例では、TLS証明書を使用してロードバランサーを作成します。TLS 証明書は指定されたドメインに適用され、トラフィックをポート 80 のインスタンスにルーティングします。

```
aws lightsail create-load-balancer \  
  --certificate-alternative-names www.example.com test.example.com \  
  --certificate-domain-name example.com \  
  --certificate-name Certificate-1 \  
  --instance-port 80 \  
  --load-balancer-name LoadBalancer-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "cc7b920a-83d8-4762-a74e-9174fe1540be",  
      "resourceName": "LoadBalancer-1",  
      "resourceType": "LoadBalancer",  
      "createdAt": 1569867169.406,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "CreateLoadBalancer",  
      "status": "Started",  
      "statusChangedAt": 1569867169.406  
    },  
    {  
      "id": "658ed43b-f729-42f3-a8e4-3f8024d3c98d",  
      "resourceName": "LoadBalancer-1",  
      "resourceType": "LoadBalancerTlsCertificate",  
      "createdAt": 1569867170.193,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
    }  
  ]  
}
```

```
    "operationDetails": "LoadBalancer-1",
    "operationType": "CreateLoadBalancerTlsCertificate",
    "status": "Succeeded",
    "statusChangedAt": 1569867170.54
  },
  {
    "id": "4757a342-5181-4870-b1e0-227eebc35ab5",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1569867170.193,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "Certificate-1",
    "operationType": "CreateLoadBalancerTlsCertificate",
    "status": "Succeeded",
    "statusChangedAt": 1569867170.54
  }
]
}
```

詳細については、[「Lightsail デベロッパーガイド」の「Lightsail ロードバランサー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLoadBalancer](#)」の「」を参照してください。

create-relational-database-from-snapshot

次のコード例は、create-relational-database-from-snapshot を使用する方法を示しています。

AWS CLI

スナップショットからマネージドデータベースを作成するには

次のcreate-relational-database-from-snapshot例では、\$15 のUSD標準データベースバンドルを使用して、指定された AWS リージョンとアベイラビリティーゾーンの指定されたスナップショットからマネージドデータベースを作成します。注：指定するバンドルは、スナップ

ショットの作成に使用される元のソースデータベースのバンドルと同等か、それ以上である必要があります。

```
aws lightsail create-relational-database-from-snapshot \  
  --relational-database-snapshot-name Database-Oregon-1-1566839359 \  
  --relational-database-name Database-1 \  
  --availability-zone us-west-2a \  
  --relational-database-bundle-id micro_1_0 \  
  --no-publicly-accessible
```

出力:

```
{  
  "operations": [  
    {  
      "id": "ad6d9193-9d5c-4ea1-97ae-8fe6de600b4c",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1569867916.938,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "CreateRelationalDatabaseFromSnapshot",  
      "status": "Started",  
      "statusChangedAt": 1569867918.643  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateRelationalDatabaseFromSnapshot](#)」の「」を参照してください。

create-relational-database-snapshot

次のコード例は、create-relational-database-snapshot を使用する方法を示しています。

AWS CLI

マネージドデータベースのスナップショットを作成するには

次のcreate-relational-database-snapshot例では、指定されたマネージドデータベースのスナップショットを作成します。

```
aws lightsail create-relational-database-snapshot \  
  --relational-database-name Database1 \  
  --relational-database-snapshot-name RelationalDatabaseSnapshot1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "853667fb-ea91-4c02-8d20-8fc5fd43b9eb",  
      "resourceName": "RelationalDatabaseSnapshot1",  
      "resourceType": "RelationalDatabaseSnapshot",  
      "createdAt": 1569868074.645,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "Database1",  
      "operationType": "CreateRelationalDatabaseSnapshot",  
      "status": "Started",  
      "statusChangedAt": 1569868074.645  
    },  
    {  
      "id": "fbafa521-3cac-4be8-9773-1c143780b239",  
      "resourceName": "Database1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1569868074.645,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "RelationalDatabaseSnapshot1",  
      "operationType": "CreateRelationalDatabaseSnapshot",  
      "status": "Started",  
      "statusChangedAt": 1569868074.645  
    }  
  ]  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRelationalDatabaseSnapshot](#)」の「」を参照してください。

create-relational-database

次のコード例は、create-relational-database を使用する方法を示しています。

AWS CLI

マネージドデータベースを作成するには

次のcreate-relational-database例では、MySQL 5.6 データベースエンジン (mysql_5_6) と \$15 のUSD標準データベースバンドル (micro_1_0) を使用して、指定された AWS リージョンとアベイラビリティゾーンにマネージドデータベースを作成します。マネージドデータベースにはマスターユーザー名があらかじめ入力されており、パブリックにアクセスすることはできません。

```
aws lightsail create-relational-database \  
  --relational-database-name Database-1 \  
  --availability-zone us-west-2a \  
  --relational-database-blueprint-id mysql_5_6 \  
  --relational-database-bundle-id micro_1_0 \  
  --master-database-name dbmaster \  
  --master-username user \  
  --no-publicly-accessible
```

出力:

```
{  
  "operations": [  
    {  
      "id": "b52bedee-73ed-4798-8d2a-9c12df89adcd",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1569450017.244,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      }  
    },  
  ],  
}
```

```
        "isTerminal": false,
        "operationType": "CreateRelationalDatabase",
        "status": "Started",
        "statusChangedAt": 1569450018.637
    }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateRelationalDatabase](#)」の「」を参照してください。

delete-auto-snapshot

次のコード例は、delete-auto-snapshot を使用する方法を示しています。

AWS CLI

自動スナップショットを削除するには

次のdelete-auto-snapshot例では、2019-10-10インスタンスの自動スナップショットを削除しますWordPress-1。

```
aws lightsail delete-auto-snapshot \
  --resource-name WordPress-1 \
  --date 2019-10-10
```

出力:

```
{
  "operations": [
    {
      "id": "31c36e09-3d52-46d5-b6d8-7EXAMPLE534a",
      "resourceName": "WordPress-1",
      "resourceType": "Instance",
      "createdAt": 1571088141.501,
      "location": {
        "availabilityZone": "us-west-2",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "DeleteAutoSnapshot-2019-10-10",
    }
  ]
}
```

```
        "operationType": "DeleteAutoSnapshot",
        "status": "Succeeded"
    }
]
}
```

詳細については、[Amazon Lightsail でのインスタンスまたはディスクの自動スナップショットの削除](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAutoSnapshot](#)」の「」を参照してください。

delete-disk-snapshot

次のコード例は、delete-disk-snapshot を使用する方法を示しています。

AWS CLI

ブロックストレージディスクのスナップショットを削除するには

次のdelete-disk-snapshot例では、ブロックストレージディスクの指定されたスナップショットを削除します。

```
aws lightsail delete-disk-snapshot \
  --disk-snapshot-name DiskSnapshot-1
```

出力:

```
{
  "operations": [
    {
      "id": "d1e5766d-b81e-4595-ad5d-02afbcccfd5d",
      "resourceName": "DiskSnapshot-1",
      "resourceType": "DiskSnapshot",
      "createdAt": 1569873552.79,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "DeleteDiskSnapshot",
```

```
        "status": "Succeeded",
        "statusChangedAt": 1569873552.79
      }
    ]
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDiskSnapshot](#)」の「」を参照してください。

delete-disk

次のコード例は、delete-disk を使用する方法を示しています。

AWS CLI

ブロックストレージディスクを削除するには

次のdelete-disk例では、指定されたブロックストレージディスクを削除します。

```
aws lightsail delete-disk \  
  --disk-name Disk-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "6378c70f-4d75-4f7a-ab66-730fca0bb2fc",  
      "resourceName": "Disk-1",  
      "resourceType": "Disk",  
      "createdAt": 1569872887.864,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationType": "DeleteDisk",  
      "status": "Succeeded",  
      "statusChangedAt": 1569872887.864  
    }  
  ]  
}
```



```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDisk](#)」の「」を参照してください。

delete-domain-entry

次のコード例は、delete-domain-entry を使用する方法を示しています。

AWS CLI

ドメインエントリを削除するには (DNS レコード)

次のdelete-domain-entry例では、既存のドメインから指定されたドメインエントリを削除します。

注 : Lightsail のドメイン関連のAPIオペレーションは、us-east-1リージョンでのみ使用できます。CLI プロファイルが別のリージョンを使用するように設定されている場合は、--region us-east-1パラメータを含める必要があります。そうしないと、コマンドは失敗します。

```
aws lightsail delete-domain-entry \  
  --region us-east-1 \  
  --domain-name example.com \  
  --domain-entry name=123.example.com,target=192.0.2.0,type=A
```

出力:

```
{  
  "operation": {  
    "id": "06eacd01-d785-420e-8daa-823150c7dca1",  
    "resourceName": "example.com ",  
    "resourceType": "Domain",  
    "createdAt": 1569874157.005,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "global"  
    },  
    "isTerminal": true,  
    "operationType": "DeleteDomainEntry",  
    "status": "Succeeded",  
    "statusChangedAt": 1569874157.005  
  }  
}
```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDomainEntry](#)」の「」を参照してください。

delete-domain

次の例は、delete-domain を使用する方法を説明しています。

AWS CLI

ドメインを削除するには (DNS ゾーン)

次のdelete-domain例では、指定されたドメインとドメイン内のすべてのエントリ (DNS レコード) を削除します。

注 : Lightsail のドメイン関連のAPIオペレーションは、 us-east-1リージョンでのみ使用できます。CLI プロファイルが別のリージョンを使用するように設定されている場合は、 --region us-east-1パラメータを含める必要があります。そうしないと、コマンドは失敗します。

```
aws lightsail delete-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "operation": {  
    "id": "fcef5265-5af1-4a46-a3d7-90b5e18b9b32",  
    "resourceName": "example.com",  
    "resourceType": "Domain",  
    "createdAt": 1569873788.13,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "global"  
    },  
    "isTerminal": true,  
    "operationType": "DeleteDomain",  
    "status": "Succeeded",  
    "statusChangedAt": 1569873788.13  
  }  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDomain](#)」の「」を参照してください。

delete-instance-snapshot

次のコード例は、delete-instance-snapshot を使用する方法を示しています。

AWS CLI

title

次のdelete-instance-snapshot例では、インスタンスの指定されたスナップショットを削除します。

```
aws lightsail delete-instance-snapshot \  
  --instance-snapshot-name WordPress-1-Snapshot-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "14dad182-976a-46c6-bfd4-9480482bf0ea",  
      "resourceName": "WordPress-1-Snapshot-1",  
      "resourceType": "InstanceSnapshot",  
      "createdAt": 1569874524.562,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationType": "DeleteInstanceSnapshot",  
      "status": "Succeeded",  
      "statusChangedAt": 1569874524.562  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInstanceSnapshot](#)」の「」を参照してください。

delete-instance

次の例は、delete-instance を使用する方法を説明しています。

AWS CLI

インスタンスを削除するには

次のdelete-instance例では、指定されたインスタンスを削除します。

```
aws lightsail delete-instance \  
  --instance-name WordPress-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "d77345a3-8f80-4d2e-b47d-aaa622718df2",  
      "resourceName": "Disk-1",  
      "resourceType": "Disk",  
      "createdAt": 1569874357.469,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "WordPress-1",  
      "operationType": "DetachDisk",  
      "status": "Started",  
      "statusChangedAt": 1569874357.469  
    },  
    {  
      "id": "708fa606-2bfd-4e48-a2c1-0b856585b5b1",  
      "resourceName": "WordPress-1",  
      "resourceType": "Instance",  
      "createdAt": 1569874357.465,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationDetails": "Disk-1",
```

```
    "operationType": "DetachDisk",
    "status": "Started",
    "statusChangedAt": 1569874357.465
  },
  {
    "id": "3187e823-8acb-405d-b098-fad5ceb17bec",
    "resourceName": "WordPress-1",
    "resourceType": "Instance",
    "createdAt": 1569874357.829,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "DeleteInstance",
    "status": "Succeeded",
    "statusChangedAt": 1569874357.829
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInstance](#)」の「」を参照してください。

delete-key-pair

次のコード例は、delete-key-pair を使用する方法を示しています。

AWS CLI

キーペアを削除するには

次のdelete-key-pair例では、指定されたキーペアを削除します。

```
aws lightsail delete-key-pair \
  --key-pair-name MyPersonalKeyPair
```

出力:

```
{
  "operation": {
    "id": "81621463-df38-4810-b866-6e801a15abbf",
```

```
    "resourceName": "MyPersonalKeyPair",
    "resourceType": "KeyPair",
    "createdAt": 1569874626.466,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "DeleteKeyPair",
    "status": "Succeeded",
    "statusChangedAt": 1569874626.685
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteKeyPair](#)」の「」を参照してください。

delete-known-host-keys

次のコード例は、delete-known-host-keys を使用する方法を示しています。

AWS CLI

インスタンスから既知のホストキーを削除するには

次のdelete-known-host-keys例では、指定されたインスタンスから既知のホストキーを削除します。

```
aws lightsail delete-known-host-keys \
  --instance-name Instance-1
```

出力:

```
{
  "operations": [
    {
      "id": "c61afe9c-45a4-41e6-a97e-d212364da3f5",
      "resourceName": "Instance-1",
      "resourceType": "Instance",
      "createdAt": 1569874760.201,
      "location": {
```

```
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "DeleteKnownHostKeys",
    "status": "Succeeded",
    "statusChangedAt": 1569874760.201
}
]
```

詳細については、[Amazon Lightsail ブラウザベースSSHまたはRDPクライアントとの接続問題のトラブルシューティング](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteKnownHostKeys](#)」の「」を参照してください。

delete-load-balancer-tls-certificate

次のコード例は、delete-load-balancer-tls-certificate を使用する方法を示しています。

AWS CLI

ロードバランサーのTLS証明書を削除するには

次のdelete-load-balancer-tls-certificate例では、指定されたロードバランサーから指定TLS証明書を削除します。

```
aws lightsail delete-load-balancer-tls-certificate \
  --load-balancer-name MyFirstLoadBalancer \
  --certificate-name MyFirstCertificate
```

出力:

```
{
  "operations": [
    {
      "id": "50bec274-e45e-4caa-8a69-b763ef636583",
      "resourceName": "MyFirstCertificate",
      "resourceType": "LoadBalancerTlsCertificate",
      "createdAt": 1569874989.48,
```

```
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "DeleteLoadBalancerTlsCertificate",
    "status": "Started",
    "statusChangedAt": 1569874989.48
  },
  {
    "id": "78c58cdc-a59a-4b27-8213-500638634a8f",
    "resourceName": "MyFirstLoadBalancer",
    "resourceType": "LoadBalancer",
    "createdAt": 1569874989.48,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationType": "DeleteLoadBalancerTlsCertificate",
    "status": "Started",
    "statusChangedAt": 1569874989.48
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLoadBalancerTlsCertificate](#)」の「」を参照してください。

delete-load-balancer

次の例は、delete-load-balancer を使用する方法を説明しています。

AWS CLI

ロードバランサーを削除するには

次のdelete-load-balancer例では、指定されたロードバランサーと関連するTLS証明書を削除します。

```
aws lightsail delete-load-balancer \  
  --load-balancer-name MyFirstLoadBalancer
```


出力:

```
{
  "operations": [
    {
      "id": "a8c968c7-72a3-4680-a714-af8f03eea535",
      "resourceName": "MyFirstLoadBalancer",
      "resourceType": "LoadBalancer",
      "createdAt": 1569875092.125,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "DeleteLoadBalancer",
      "status": "Succeeded",
      "statusChangedAt": 1569875092.125
    },
    {
      "id": "f91a29fc-8ce3-4e69-a227-ea70ca890bf5",
      "resourceName": "MySecondCertificate",
      "resourceType": "LoadBalancerTlsCertificate",
      "createdAt": 1569875091.938,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "DeleteLoadBalancerTlsCertificate",
      "status": "Started",
      "statusChangedAt": 1569875091.938
    },
    {
      "id": "cf64c060-154b-4eb4-ba57-84e2e41563d6",
      "resourceName": "MyFirstLoadBalancer",
      "resourceType": "LoadBalancer",
      "createdAt": 1569875091.94,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "DeleteLoadBalancerTlsCertificate",
      "status": "Started",
```

```
        "statusChangedAt": 1569875091.94
      }
    ]
  }
```

詳細については、ガイドの「タイトル」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteLoadBalancer](#)」の「」を参照してください。

delete-relational-database-snapshot

次のコード例は、`delete-relational-database-snapshot` を使用する方法を示しています。

AWS CLI

マネージドデータベースのスナップショットを削除するには

次の`delete-relational-database-snapshot`例では、マネージドデータベースの指定されたスナップショットを削除します。

```
aws lightsail delete-relational-database-snapshot \  
  --relational-database-snapshot-name Database-Oregon-1-1566839359
```

出力:

```
{  
  "operations": [  
    {  
      "id": "b99acae8-735b-4823-922f-30af580e3729",  
      "resourceName": "Database-Oregon-1-1566839359",  
      "resourceType": "RelationalDatabaseSnapshot",  
      "createdAt": 1569875293.58,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationType": "DeleteRelationalDatabaseSnapshot",  
      "status": "Succeeded",  
      "statusChangedAt": 1569875293.58  
    }  
  ]  
}
```

```
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRelationalDatabaseSnapshot](#)」の「」を参照してください。

delete-relational-database

次の例は、delete-relational-database を使用する方法を説明しています。

AWS CLI

マネージドデータベースを削除するには

次のdelete-relational-database例では、指定されたマネージドデータベースを削除します。

```
aws lightsail delete-relational-database \  
  --relational-database-name Database-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "3b0c41c1-053d-46f0-92a3-14f76141dc86",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1569875210.999,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "DeleteRelationalDatabase",  
      "status": "Started",  
      "statusChangedAt": 1569875210.999  
    },  
    {  
      "id": "01ddeae8-a87a-4a4b-a1f3-092c71bf9180",  
      "resourceName": "Database-1",
```

```
    "resourceType": "RelationalDatabase",
    "createdAt": 1569875211.029,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "Database-1-FinalSnapshot-1569875210793",
    "operationType": "CreateRelationalDatabaseSnapshot",
    "status": "Started",
    "statusChangedAt": 1569875211.029
  },
  {
    "id": "74d73681-30e8-4532-974e-1f23cd3f9f73",
    "resourceName": "Database-1-FinalSnapshot-1569875210793",
    "resourceType": "RelationalDatabaseSnapshot",
    "createdAt": 1569875211.029,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "Database-1",
    "operationType": "CreateRelationalDatabaseSnapshot",
    "status": "Started",
    "statusChangedAt": 1569875211.029
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRelationalDatabase](#)」の「」を参照してください。

detach-static-ip

次のコード例は、detach-static-ip を使用する方法を示しています。

AWS CLI

インスタンスから静的 IP をデタッチするには

次のdetach-static-ip例では、アタッチされたインスタンスStaticIp-1から静的 IP をデタッチします。

```
aws lightsail detach-static-ip \  
  --static-ip-name StaticIp-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "2a43d8a3-9f2d-4fe7-bdd0-eEXAMPLE3cf3",  
      "resourceName": "StaticIp-1",  
      "resourceType": "StaticIp",  
      "createdAt": 1571088261.999,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationDetails": "MEAN-1",  
      "operationType": "DetachStaticIp",  
      "status": "Succeeded",  
      "statusChangedAt": 1571088261.999  
    },  
    {  
      "id": "41a7d40c-74e8-4d2e-a837-cEXAMPLEf747",  
      "resourceName": "MEAN-1",  
      "resourceType": "Instance",  
      "createdAt": 1571088262.022,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationDetails": "StaticIp-1",  
      "operationType": "DetachStaticIp",  
      "status": "Succeeded",  
      "statusChangedAt": 1571088262.022  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DetachStaticIp](#)」の「」を参照してください。

get-active-names

次のコード例は、get-active-names を使用する方法を示しています。

AWS CLI

アクティブなリソース名を取得するには

次のget-active-names例では、設定された AWS リージョンのアクティブなリソース名を返します。

```
aws lightsail get-active-names
```

出力:

```
{
  "activeNames": [
    "WordPress-1",
    "StaticIp-1",
    "MEAN-1",
    "Plesk_Hosting_Stack_on_Ubuntu-1"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetActiveNames](#)」の「」を参照してください。

get-auto-snapshots

次のコード例は、get-auto-snapshots を使用する方法を示しています。

AWS CLI

インスタンスで使用可能な自動スナップショットを取得するには

次のget-auto-snapshots例では、インスタンスで使用可能な自動スナップショットを返しますWordPress-1。

```
aws lightsail get-auto-snapshots \
  --resource-name WordPress-1
```

出力:

```
{
  "resourceName": "WordPress-1",
  "resourceType": "Instance",
  "autoSnapshots": [
    {
      "date": "2019-10-14",
      "createdAt": 1571033872.0,
      "status": "Success",
      "fromAttachedDisks": []
    },
    {
      "date": "2019-10-13",
      "createdAt": 1570947473.0,
      "status": "Success",
      "fromAttachedDisks": []
    },
    {
      "date": "2019-10-12",
      "createdAt": 1570861072.0,
      "status": "Success",
      "fromAttachedDisks": []
    },
    {
      "date": "2019-10-11",
      "createdAt": 1570774672.0,
      "status": "Success",
      "fromAttachedDisks": []
    }
  ]
}
```

詳細については、[Amazon Lightsail でのインスタンスまたはディスクの自動スナップショットの保存](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAutoSnapshots](#)」の「」を参照してください。

get-blueprints

次のコード例は、get-blueprints を使用する方法を示しています。

AWS CLI

新しいインスタンスのブループリントを取得するには

次のget-blueprints例では、Amazon Lightsail で新しいインスタンスを作成するために使用できるすべてのブループリントの詳細を表示します。

```
aws lightsail get-blueprints
```

出力:

```
{
  "blueprints": [
    {
      "blueprintId": "wordpress",
      "name": "WordPress",
      "group": "wordpress",
      "type": "app",
      "description": "Bitnami, the leaders in application packaging, and Automattic, the experts behind WordPress, have teamed up to offer this official WordPress image. This image is a pre-configured, ready-to-run image for running WordPress on Amazon Lightsail. WordPress is the world's most popular content management platform. Whether it's for an enterprise or small business website, or a personal or corporate blog, content authors can easily create content using its new Gutenberg editor, and developers can extend the base platform with additional features. Popular plugins like Jetpack, Akismet, All in One SEO Pack, WP Mail, Google Analytics for WordPress, and Amazon Polly are all pre-installed in this image. Let's Encrypt SSL certificates are supported through an auto-configuration script.",
      "isActive": true,
      "minPower": 0,
      "version": "6.5.3-0",
      "versionCode": "1",
      "productUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U",
      "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U#pdp-usage",
      "platform": "LINUX_UNIX"
    },
    {
      "blueprintId": "lamp_8_bitnami",
      "name": "LAMP (PHP 8)",
      "group": "lamp_8",
      "type": "app",
```



```

        "description": "LAMP with PHP 8.X packaged by Bitnami enables you
to quickly start building your websites and applications by providing a coding
framework. As a developer, it provides standalone project directories to store your
applications. This blueprint is configured for production environments. It includes
SSL auto-configuration with Let's Encrypt certificates, and the latest releases of
PHP, Apache, and MariaDB on Linux. This application also includes phpMyAdmin, PHP
main modules and Composer.",
        "isActive": true,
        "minPower": 0,
        "version": "8.2.18-4",
        "versionCode": "1",
        "productUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dву",
        "licenseUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dву#pdp-usage",
        "platform": "LINUX_UNIX"
    },
    {
        "blueprintId": "nodejs",
        "name": "Node.js",
        "group": "node",
        "type": "app",
        "description": "Node.js packaged by Bitnami is a pre-configured, ready
to run image for Node.js on Amazon EC2. It includes the latest version of Node.js,
Apache, Python and Redis. The image supports multiple Node.js applications, each
with its own virtual host and project directory. It is configured for production
use and is secure by default, as all ports except HTTP, HTTPS and SSH ports are
closed. Let's Encrypt SSL certificates are supported through an auto-configuration
script. Developers benefit from instant access to a secure, update and consistent
Node.js environment without having to manually install and configure multiple
components and libraries.",
        "isActive": true,
        "minPower": 0,
        "version": "18.20.2-0",
        "versionCode": "1",
        "productUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAK0",
        "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAK0#pdp-
usage",
        "platform": "LINUX_UNIX"
    },
    ...
}
]

```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBlueprints](#)」の「」を参照してください。

get-bundles

次の例は、get-bundles を使用する方法を説明しています。

AWS CLI

新しいインスタンスのバンドルを取得するには

次のget-bundles例では、Amazon Lightsail で新しいインスタンスを作成するために使用できるすべてのバンドルの詳細を表示します。

```
aws lightsail get-bundles
```

出力:

```
{
  "bundles": [
    {
      "price": 5.0,
      "cpuCount": 2,
      "diskSizeInGb": 20,
      "bundleId": "nano_3_0",
      "instanceType": "nano",
      "isActive": true,
      "name": "Nano",
      "power": 298,
      "ramSizeInGb": 0.5,
      "transferPerMonthInGb": 1024,
      "supportedPlatforms": [
        "LINUX_UNIX"
      ]
    },
    {
      "price": 7.0,
      "cpuCount": 2,
      "diskSizeInGb": 40,
```

```
    "bundleId": "micro_3_0",
    "instanceType": "micro",
    "isActive": true,
    "name": "Micro",
    "power": 500,
    "ramSizeInGb": 1.0,
    "transferPerMonthInGb": 2048,
    "supportedPlatforms": [
      "LINUX_UNIX"
    ]
  },
  {
    "price": 12.0,
    "cpuCount": 2,
    "diskSizeInGb": 60,
    "bundleId": "small_3_0",
    "instanceType": "small",
    "isActive": true,
    "name": "Small",
    "power": 1000,
    "ramSizeInGb": 2.0,
    "transferPerMonthInGb": 3072,
    "supportedPlatforms": [
      "LINUX_UNIX"
    ]
  },
  ...
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBundles](#)」の「」を参照してください。

get-cloud-formation-stack-records

次の例は、get-cloud-formation-stack-records を使用する方法を説明しています。

AWS CLI

CloudFormation スタックレコードと関連するスタックを取得するには

次の`get-cloud-formation-stack-records`例では、エクスポートされた Amazon Lightsail スタックシヨットから Amazon EC2リソースを作成するために使用される CloudFormation スタックレコードと、それらの関連スタックの詳細を表示します。

```
aws lightsail get-cloud-formation-stack-records
```

出力:

```
{
  "cloudFormationStackRecords": [
    {
      "name": "CloudFormationStackRecord-588a4243-
e2d1-490d-8200-3a7513ecebdf",
      "arn": "arn:aws:lightsail:us-
west-2:111122223333:CloudFormationStackRecord/28d646ab-27bc-48d9-a422-1EXAMPLE6d37",
      "createdAt": 1565301666.586,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "CloudFormationStackRecord",
      "state": "Succeeded",
      "sourceInfo": [
        {
          "resourceType": "ExportSnapshotRecord",
          "name": "ExportSnapshotRecord-
e02f23d7-0453-4aa9-9c95-91aa01a141dd",
          "arn": "arn:aws:lightsail:us-
west-2:111122223333:ExportSnapshotRecord/f12b8792-f3ea-4d6f-b547-2EXAMPLE8796"
        }
      ],
      "destinationInfo": {
        "id": "arn:aws:cloudformation:us-west-2:111122223333:stack/
Lightsail-Stack-588a4243-e2d1-490d-8200-3EXAMPLEebdf/063203b0-
ba28-11e9-838b-0EXAMPLE8b00",
        "service": "Aws::CloudFormation::Stack"
      }
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetCloudFormationStackRecords](#)」の「」を参照してください。

get-disk-snapshot

次の例は、get-disk-snapshot を使用する方法を説明しています。

AWS CLI

ディスクスナップショットに関する情報を取得するには

次のget-disk-snapshot例では、ディスクスナップショットの詳細を表示しますDisk-1-1566839161。

```
aws lightsail get-disk-snapshot \  
--disk-snapshot-name Disk-1-1566839161
```

出力:

```
{  
  "diskSnapshot": {  
    "name": "Disk-1-1566839161",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/  
e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",  
    "supportCode": "6EXAMPLE3362/snap-0EXAMPLE06100d09",  
    "createdAt": 1566839163.749,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "DiskSnapshot",  
    "tags": [],  
    "sizeInGb": 8,  
    "state": "completed",  
    "progress": "100%",  
    "fromDiskName": "Disk-1",  
    "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/  
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",  
    "isFromAutoSnapshot": false  
  }  
}
```

詳細については、ガイドの「タイトル」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDiskSnapshot](#)」の「」を参照してください。

get-disk-snapshots

次のコード例は、get-disk-snapshots を使用する方法を示しています。

AWS CLI

すべてのディスクスナップショットに関する情報を取得するには

次のget-disk-snapshots例では、設定された AWS リージョンのすべてのディスクスナップショットの詳細を表示します。

```
aws lightsail get-disk-snapshots
```

出力:

```
{
  "diskSnapshots": [
    {
      "name": "Disk-2-1571090588",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/32e889a9-38d4-4687-9f21-eEXAMPLE7839",
      "supportCode": "6EXAMPLE3362/snap-0EXAMPLE1ca192a4",
      "createdAt": 1571090591.226,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "DiskSnapshot",
      "tags": [],
      "sizeInGb": 8,
      "state": "completed",
      "progress": "100%",
      "fromDiskName": "Disk-2",
      "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",
      "isFromAutoSnapshot": false
    },
    {
```

```
    "name": "Disk-1-1566839161",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:DiskSnapshot/
e2d0fa53-8ee0-41a0-8e56-0EXAMPLE1051",
    "supportCode": "6EXAMPLE3362/snap-0EXAMPLEe06100d09",
    "createdAt": 1566839163.749,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "DiskSnapshot",
    "tags": [],
    "sizeInGb": 8,
    "state": "completed",
    "progress": "100%",
    "fromDiskName": "Disk-1",
    "fromDiskArn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
    "isFromAutoSnapshot": false
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDiskSnapshots](#)」の「」を参照してください。

get-disk

次の例は、get-disk を使用する方法を説明しています。

AWS CLI

ブロックストレージディスクに関する情報を取得するには

次のget-disk例では、ディスクの詳細を表示しますDisk-1。

```
aws lightsail get-disk \
  --disk-name Disk-1
```

出力:

```
{
  "disk": {
```

```
    "name": "Disk-1",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
    "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
    "createdAt": 1566585439.587,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "Disk",
    "tags": [],
    "sizeInGb": 8,
    "isSystemDisk": false,
    "iops": 100,
    "path": "/dev/xvdf",
    "state": "in-use",
    "attachedTo": "WordPress_Multisite-1",
    "isAttached": true,
    "attachmentState": "attached"
  }
}
```

詳細については、ガイドの「タイトル」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDisk](#)」の「」を参照してください。

get-disks

次のコード例は、get-disks を使用する方法を示しています。

AWS CLI

すべてのブロックストレージディスクに関する情報を取得するには

次のget-disks例では、設定された AWS リージョン内のすべてのディスクの詳細を表示します。

```
aws lightsail get-disks
```

出力:

```
{
```



```
"disks": [  
  {  
    "name": "Disk-2",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/6a343ff8-6341-422d-86e2-bEXAMPLE16c2",  
    "supportCode": "6EXAMPLE3362/vol-0EXAMPLE929602087",  
    "createdAt": 1571090461.634,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "Disk",  
    "tags": [],  
    "sizeInGb": 8,  
    "isSystemDisk": false,  
    "iops": 100,  
    "state": "available",  
    "isAttached": false,  
    "attachmentState": "detached"  
  },  
  {  
    "name": "Disk-1",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",  
    "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",  
    "createdAt": 1566585439.587,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "Disk",  
    "tags": [],  
    "sizeInGb": 8,  
    "isSystemDisk": false,  
    "iops": 100,  
    "path": "/dev/xvdf",  
    "state": "in-use",  
    "attachedTo": "WordPress_Multisite-1",  
    "isAttached": true,  
    "attachmentState": "attached"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDisks](#)」の「」を参照してください。

get-domain

次のコード例は、get-domain を使用する方法を示しています。

AWS CLI

ドメインに関する情報を取得するには

次のget-domain例では、ドメインの詳細を表示しますexample.com。

注：Lightsail のドメイン関連のAPIオペレーションは、us-east-1 AWS リージョンでのみ使用できます。CLI プロファイルが別のリージョンを使用するように設定されている場合は、「--region us-east-1」パラメータを含める必要があります。そうしないと、コマンドは失敗します。

```
aws lightsail get-domain \  
  --domain-name example.com \  
  --region us-east-1
```

出力:

```
{  
  "domain": {  
    "name": "example.com",  
    "arn":  
    "arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEeb304",  
    "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",  
    "createdAt": 1570728588.6,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "global"  
    },  
    "resourceType": "Domain",  
    "tags": [],  
    "domainEntries": [  
      {  
        "id": "-1682899164",  
        "name": "example.com",  
        "target": "192.0.2.0",  
        "isAlias": false,  
        "arn": "arn:aws:lightsail:global:111122223333:DomainEntry/28cda903-3f15-44b2-9baf-3EXAMPLEeb304:example.com",  
        "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",  
        "createdAt": 1570728588.6,  
        "location": {  
          "availabilityZone": "all",  
          "regionName": "global"  
        },  
        "resourceType": "DomainEntry",  
        "tags": []  
      }  
    ]  
  }  
}
```

```
    "type": "A"
  },
  {
    "id": "1703104243",
    "name": "example.com",
    "target": "ns-137.awsdns-17.com",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "-1038331153",
    "name": "example.com",
    "target": "ns-1710.awsdns-21.co.uk",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "-2107289565",
    "name": "example.com",
    "target": "ns-692.awsdns-22.net",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "1582095705",
    "name": "example.com",
    "target": "ns-1436.awsdns-51.org",
    "isAlias": false,
    "type": "NS"
  },
  {
    "id": "-1769796132",
    "name": "example.com",
    "target": "ns-1710.awsdns-21.co.uk. awsdns-hostmaster.amazon.com. 1
7200 900 1209600 86400",
    "isAlias": false,
    "type": "SOA"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDomain](#)」の「」を参照してください。

get-domains

次の例は、get-domains を使用する方法を説明しています。

AWS CLI

すべてのドメインに関する情報を取得するには

次のget-domains例では、設定された AWS リージョン内のすべてのドメインの詳細を表示します。

注：Lightsail のドメイン関連のAPIオペレーションは、us-east-1 AWS リージョンでのみ使用できます。CLI プロファイルが別のリージョンを使用するように設定されている場合は、`--region us-east-1`パラメータを含める必要があります。そうしないと、コマンドは失敗します。

```
aws lightsail get-domains \  
  --region us-east-1
```

出力:

```
{  
  "domains": [  
    {  
      "name": "example.com",  
      "arn":  
"arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",  
      "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",  
      "createdAt": 1570728588.6,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "global"  
      },  
      "resourceType": "Domain",  
      "tags": [],  
      "domainEntries": [  
        {  
          "id": "-1682899164",  
          "name": "example.com",
```

```
        "target": "192.0.2.0",
        "isAlias": false,
        "type": "A"
    },
    {
        "id": "1703104243",
        "name": "example.com",
        "target": "ns-137.awsdns-17.com",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "-1038331153",
        "name": "example.com",
        "target": "ns-4567.awsdns-21.co.uk",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "-2107289565",
        "name": "example.com",
        "target": "ns-333.awsdns-22.net",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "1582095705",
        "name": "example.com",
        "target": "ns-1111.awsdns-51.org",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "-1769796132",
        "name": "example.com",
        "target": "ns-1234.awsdns-21.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
        "isAlias": false,
        "type": "SOA"
    },
    {
        "id": "1029454894",
        "name": "_dead6a124ede046a0319eb44a4eb3cbc.example.com",
```

```
        "target": "_be133b0a0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws",
        "isAlias": false,
        "type": "CNAME"
    }
]
},
{
    "name": "example.net",
    "arn": "arn:aws:lightsail:global:111122223333:Domain/9c9f0d70-
c92e-4753-86c2-6EXAMPLE029d",
    "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLE5TPKMV",
    "createdAt": 1556661071.384,
    "location": {
        "availabilityZone": "all",
        "regionName": "global"
    },
    "resourceType": "Domain",
    "tags": [],
    "domainEntries": [
        {
            "id": "-766320943",
            "name": "example.net",
            "target": "192.0.2.2",
            "isAlias": false,
            "type": "A"
        },
        {
            "id": "-453913825",
            "name": "example.net",
            "target": "ns-123.awsdns-10.net",
            "isAlias": false,
            "type": "NS"
        },
        {
            "id": "1553601564",
            "name": "example.net",
            "target": "ns-4444.awsdns-47.co.uk",
            "isAlias": false,
            "type": "NS"
        },
        {
            "id": "1653797661",
            "name": "example.net",
```

```
        "target": "ns-7890.awsdns-61.org",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "706414698",
        "name": "example.net",
        "target": "ns-123.awsdns-44.com",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "337271745",
        "name": "example.net",
        "target": "ns-4444.awsdns-47.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
        "isAlias": false,
        "type": "SOA"
    },
    {
        "id": "-1785431096",
        "name": "www.example.net",
        "target": "192.0.2.2",
        "isAlias": false,
        "type": "A"
    }
]
},
{
    "name": "example.org",
    "arn": "arn:aws:lightsail:global:111122223333:Domain/
f0f13ba3-3df0-4fdc-8ebb-1EXAMPLEf26e",
    "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEAF038",
    "createdAt": 1556661199.106,
    "location": {
        "availabilityZone": "all",
        "regionName": "global"
    },
    "resourceType": "Domain",
    "tags": [],
    "domainEntries": [
        {
            "id": "2065301345",
            "name": "example.org",
```

```
        "target": "192.0.2.4",
        "isAlias": false,
        "type": "A"
    },
    {
        "id": "-447198516",
        "name": "example.org",
        "target": "ns-123.awsdns-45.com",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "136463022",
        "name": "example.org",
        "target": "ns-9999.awsdns-15.co.uk",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "1395941679",
        "name": "example.org",
        "target": "ns-555.awsdns-01.net",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "872052569",
        "name": "example.org",
        "target": "ns-6543.awsdns-38.org",
        "isAlias": false,
        "type": "NS"
    },
    {
        "id": "1001949377",
        "name": "example.org",
        "target": "ns-1234.awsdns-15.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
        "isAlias": false,
        "type": "SOA"
    },
    {
        "id": "1046191192",
        "name": "www.example.org",
        "target": "192.0.2.4",
```



```
        "isAlias": false,  
        "type": "A"  
      }  
    ]  
  }  
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDomains](#)」の「」を参照してください。

get-export-snapshot-record

次のコード例は、get-export-snapshot-record を使用する方法を示しています。

AWS CLI

Amazon にエクスポートされたスナップショットのレコードを取得するには EC2

次のget-export-snapshot-record例では、Amazon Lightsail インスタンスまたは Amazon にエクスポートされたディスクスナップショットの詳細を表示しますEC2。

```
aws lightsail get-export-snapshot-records
```

出力:

```
{  
  "exportSnapshotRecords": [  
    {  
      "name": "ExportSnapshotRecord-d2da10ce-0b3c-4ae1-ab3a-2EXAMPLEa586",  
      "arn": "arn:aws:lightsail:us-west-2:111122223333:ExportSnapshotRecord/076c7060-b0cc-4162-98f0-2EXAMPLEe28e",  
      "createdAt": 1543534665.678,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "resourceType": "ExportSnapshotRecord",  
      "state": "Succeeded",  
      "sourceInfo": {  
        "resourceType": "InstanceSnapshot",  
        "createdAt": 1540339310.706,  
      }  
    }  
  ]  
}
```

```

        "name": "WordPress-512MB-Oregon-1-1540339219",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/5446f534-ed60-4c17-b4a5-bEXAMPLEf8b7",
        "fromResourceName": "WordPress-512MB-Oregon-1",
        "fromResourceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/4b8f1f24-e4d1-4cf3-88ff-cEXAMPLEa397",
        "instanceSnapshotInfo": {
            "fromBundleId": "nano_2_0",
            "fromBlueprintId": "wordpress_4_9_8",
            "fromDiskInfo": [
                {
                    "path": "/dev/sda1",
                    "sizeInGb": 20,
                    "isSystemDisk": true
                }
            ]
        }
    },
    "destinationInfo": {
        "id": "ami-0EXAMPLEc0d65058e",
        "service": "Aws::EC2::Image"
    }
},
{
    "name": "ExportSnapshotRecord-1c94e884-40ff-4fe1-9302-0EXAMPLE14c2",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:ExportSnapshotRecord/
fb392ce8-6567-4013-9bfd-3EXAMPLE5b4c",
    "createdAt": 1543432110.2,
    "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
    },
    "resourceType": "ExportSnapshotRecord",
    "state": "Succeeded",
    "sourceInfo": {
        "resourceType": "InstanceSnapshot",
        "createdAt": 1540833603.545,
        "name": "LAMP_PHP_5-512MB-Oregon-1-1540833565",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/82334399-b5f2-49ec-8382-0EXAMPLEe45f",
        "fromResourceName": "LAMP_PHP_5-512MB-Oregon-1",
        "fromResourceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/863b9f35-ab1e-4418-bdd2-1EXAMPLEbab2",
        "instanceSnapshotInfo": {

```

```

        "fromBundleId": "nano_2_0",
        "fromBlueprintId": "lamp_5_6_37_2",
        "fromDiskInfo": [
            {
                "path": "/dev/sda1",
                "sizeInGb": 20,
                "isSystemDisk": true
            }
        ]
    },
    "destinationInfo": {
        "id": "ami-0EXAMPLE7c5ec84e2",
        "service": "Aws::EC2::Image"
    }
}
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetExportSnapshotRecord](#)」の「」を参照してください。

get-instance-access-details

次の例は、get-instance-access-details を使用する方法を説明しています。

AWS CLI

インスタンスのホストキー情報を取得するには

次のget-instance-access-details例では、インスタンスのホストキー情報を表示しますWordPress_Multisite-1。

```
aws lightsail get-instance-access-details \
  --instance-name WordPress_Multisite-1
```

出力:

```
{
  "accessDetails": {
    "certKey": "ssh-rsa-cert-v01@openssh.com
AEXAMPLEEaC1yc2EtY2VydC12MDFAb3B1bnNzaC5jb20AAAAgNf076Dt3ppmPd0fPxZVMmS491aEAYYH9cHqAJ3fNML8
```

```
vEXAMPLE2eBWJyQvn7o1/  
i0+s966h5sx8qUD791PB7q5UESd5VZGFtytrykfQJnjiwqe7EV5agzvjb1Lj26Fb37EKda9HVfCOu8pWbvky7Tyn9w29  
+xMfQM9xVz0rXZmqx8uJidJpRgLCMTviofwQJU/  
K1EXAMPLEAAAAAAAAABAAAALS00MzMzMDU4MzA40Dg1MTY2NjM40np6UWlndHk4UElRSG9Stit0TG5QSEE9PQAAAAAsAA  
+LiB+ozNbUA0cdNL9Y67x7qPv/R7XhTc21+2A+8+GuVpK/Kz9dqDMKNAEXAMPLE+YYN  
+tiXm7Y80gziK+7iDB7xUuQ4vghmn4+qgz9mKwYgWvVe2+0XLuV7cnWPB7iU1HQg  
+E3LUKrv4ZFw9pj7X2dFdNkFmxwWgI1ISWKimEXAMPLEEhJrf1Rqc/  
QH6TpWCvPfcx8uvwVqdwTfke/SfA5BCzbGGI1UmIUadh8nHcb5FamQ1hK7kECy47K/x9FMn/  
KwmM7pCwJbSLDM07n9bnbvck6m8ZoB2N2YLMG5dW7BerEXAMPLEEobqfdtyYJHHe11EyyEJs1fWNU3D5JIGlgzcPAV  
+Z1bQyUCZXf0os1Sa+HE85f0/  
FRq9SVSBSHrmb0fr1PhgMzgSmqLeyh1br6wwWIDbREXAMPLEJZ49H7RdQxdKyYrZPWvRgcr0qI2EL0tAajnpQQ8UZo  
Aqter0xN5PhFL0J490WTacwCGRAjLhibAx7K1t/1ZXWo6c+ijq8c111327EXAMPLE/  
e89GC89KcmKCxfGQniDAUgF8UqofIbq3Z0UgiAAYCVXc1I4L68NhVXyoWuQXPBRQSEXAMPLEWm74tDL9tFN3c7tSe/  
0z0cTR+4sAAAIPAAAAB3NzaC1yc2EAAAIAQnG/  
L0DqiSnLrWhEox4aHqMgd0m0oLLAYx60QH9F0TM9EXAMPLEE961rzSCMon7ZgsWnNl00wZQgDG  
+rtJ4N0B7H0Vwns4ynUFbzNQ3qFGGeE31KwX1L41vV1iSy7sDk8aI0LmrKJi1LE1Qc1l8uboRlwoX0YEXAMPLEEaUCeX  
+10+WEXAMPLEg6Y4U4ZvE2B3xyRdpvysb5TGFNTk5qPslacnVkoL0GsZZXmPLGJnG40BpQLLtpj9sNMxAgZPCAUjhkqk  
+nx0904NUZ2pTWbVSUaV1gm6pug9xbwN01Im21t34JeLlKTqxcJ6zzS8W0c0KKpAm5c4hWkseMbyutS2jav/4hiS  
+BhrYgptzfwe5qRXEXAMPLEEHZQr3YfGzYoBJ/  
lLK3NHhx0ihhsfAYwMei0BFZT1F/7CT3IH4iitEkIgodI06/  
Mw6UDqMPoZYQCK11EA6LFhYCOZG9drWcoRa741M4kY9TP028Za8gDMh1WpkXLq9Gixon50HP8aM/  
sEXAMPLEr2+fnkw+1Bto05L6+VKoPlXaGqZ/fBYEXAMPLEAMQHjnlM1JYNvtEEPhp+TNzXHzuixWf/  
Ht04m0AVpXrzIDXaS102tXY=",  
    "ipAddress": "192.0.2.0",  
    "privateKey": "-----BEGIN RSA PRIVATE KEY-----  
\nEXAMPLEBAAKCAQEa+AD3qeU2toBy505v7wnRLVo/tngVickL5+6Jf4tPrPeuoeBm  
\nfK1A+/ZTwe6uVBEneVWRhbcra8pH0CZ44sKnuxFeWoM7425S49uhW9+xCnWvR1Xw  
\njrvKVm75Mu08p/cNvfWugrBuaPB65DspgxNn0fZWMVxpIpSq0SPWmSwQHv597d6C  
\nrEXAMPLEe08hJmqz2KFQ09X7fB21BruGgr9aXiNPmWmovYKqwFmrnFvR7odFmDecq  
\n5EXAMPLE9dyU1ZsrWhGby77eYrVaF10GNGQ8qy1HGUiscquZ9NDIL49n4mXbfsTH  
\n0EXAMPLE12ZqsfliYnSaUYCwjE74qH8ECVPytQIDAQABAoIBAHeZV9Z58JHAjifz  
\nCEXAMPLEEEqC3do0VDgXS1kKI92qNo4z2VcUEho878paCuVvXVHcCGgSnGeyIh2tN  
\nMEXAMPLESohR427BhH3YLA+3Z5SivnejbTgYPfLC37B8khTaYqkqMvdZiFVZK5qn  
\nIEXAMPLEM93oF9eSZjcLKB/jGHsfb0eCDMP8BshHE2beuqzVMoK1Dx0nvoP3+Fp  
\nAEXAMPLESg6pDpCo9YVUX8g1u3Ro9cP12LXHDy+oVEY5KhbZQJ7VU1I72W0vppWW  
\n0EXAMPLEkgYlq7p6qYtYcSgTEjz14gDiMfQ7SyHB3alkIoN0NQ9ZPaWHyJvymeud  
\noQTNuz0CgYEA/LFWNTEZrzdR1kJmyNRmAermU0B6utyNENChAlHGSHkB+11VSh  
\nbEXAMPLEQo9ooUeW5Ux03YwacZLoDT1mwxw1Ptc1+PNycZoLe1fE9UdARrdmGTob  
\n817CPLSXp3xuR8VqSp2fnIc7hfiQs/NrPX9gm/E0rB0we0RKyDSzWScGyEA+z/r  
\niob+nJZq0Ybn0SuP6oMULP4vnWniWj8MIhUJU53LwSAM8DeJd0NKDdkui0d52aAL  
\nVgn7nLo88rVWKhJwVc4tu/rNgZLCr3bP4+kL6zand0KQnMLyOzNA2Ys26aa5udH1\nqWl0WTt9WEm/  
h10ndC1kn0MectrvsG17b38y5sMCgYEA54NiRGgz8oCPW6GN/FZA  
\nKEXAMPLE5tw34GEH3UxlC9n3CeJDaQmzc0ATwX4nIwRZDEqWyYZcS0btg1jhGiBD\nYEXAMPLEkC8Z71L/  
agZEAaVCEog9FqfSqwB
```

```

+XTfoKh8qur74X1yCu9p6gof1q6k9\neEXAMPLEchJcNN0g4ETIfMkCgYBdVORRHE4mqvWp0dzA7v66FdEz2YSkjAXKk
\naEXAMPLE8Z/8yBSmuBv1Qv03XA12my462uB92uzzGAuW
+1yBc2Kn1sXqYTy0y1z0\ngEXAMPLEBogjw4MqHKL1bPKMHyQU8/
q24PaYgzHPzy13wLH6pTYf1XqLHdE2D6Vv\nyEXAMPLEgQC3i/
kVVhky/2XRwRV1C7J02Bg3QGTx38hpmDa5IuofKANjA+Wa3/zy\nbEXAMPLE6ytQgD9GN/YtBq+uh0
+2ZkvXPL+CWRi0ZRxpPwYDBBFU9Cw0AuWWG1L8\nwEXAMPLExMlcysRgcWB9RNgf3Au0pFd2i6XT/
riNsvvkpmJ+VooU8g==\n-----END RSA PRIVATE KEY-----\n",
    "protocol": "ssh",
    "instanceName": "WordPress_Multisite-1",
    "username": "bitnami",
    "hostKeys": [
      {
        "algorithm": "ssh-rsa",
        "publicKey":
"AEXAMPLEaC1yc2EAAAADAQABAAQCoer9ieZTjQ3pXCHczuAYZFj1F7t
+uBkXuqeGMRex78pCvmS+DiEXAMPLEuJ1Q8dcKhrQL4HpXbD9dosVCTaJnJwb4MQqsuSVFdHFzy3guP
+BKc1WqtXJEXAMPLEsBGqZZ1rIv6a9bTA0TCplZ8AD+hSRTaSXXqg6FT
+Qf16IktH0X1Ms7xIEXAMPLEmNtjCpzZiGXDHzytoMvUgwa8uHPp440g36EUu4VqQxoUHPJKoXvcQizyk3K8ym0hP0Tp
Ot6y9HwvykEXAMPLEAfbKjbr42+u6+0S1kr4d339q2U1sTDytJhhs8HUel1wTfGRfp",
        "witnessedAt": 1570744377.699,
        "fingerprintSHA1": "SHA1:GEXAMPLEMoYgUg0ucadqU9Bt3Lk",
        "fingerprintSHA256": "SHA256:IEXAMPLEcB5vgxnAUoJawbdZ
+MwELhIp6FUxuwq/LIU"
      },
      {
        "algorithm": "ssh-ed25519",
        "publicKey":
"AEXAMPLEaC11ZDI1NTE5AAAAIC1gwGPDfGa0NxEXAMPLEJX3UNap781QxHQmn8nzlrUv",
        "witnessedAt": 1570744377.697,
        "fingerprintSHA1": "SHA1:VEXAMPLE5ReqSmTgv03sSUw9toU",
        "fingerprintSHA256": "SHA256:0EXAMPLEdE6tI95k3TJpG
+qhJbAoknB0yz9nAEaDt3A"
      },
      {
        "algorithm": "ecdsa-sha2-nistp256",
        "publicKey":
"AEXAMPLEZHNhLXNoYTIItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABEXAMPLE9B4mZy8YSsZW7cixCDq5yHSAAxjJkDo5
+EnK1DCsYtUkxxEXAMPLE6V0WL2z63RTKa2AUPgd8irjxWI=",
        "witnessedAt": 1570744377.707,
        "fingerprintSHA1": "SHA1:UEXAMPLE0YCFxScf2G6tDg+7YG0",
        "fingerprintSHA256": "SHA256:wEXAMPLEQ9a/
iEXAMPLEhRufm6U9vFU4cpkMPHnBsNA"
      }
    ]
  ]

```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceAccessDetails](#)」の「」を参照してください。

get-instance-metric-data

次の例は、get-instance-metric-data を使用する方法を説明しています。

AWS CLI

インスタンスのメトリクスデータを取得するには

次のget-instance-metric-data例では、1571342400と1571428800の間の7200秒(2時間) CPUUtilizationごとの平均パーセントを返しますMEAN-1。

Unix タイムコンバーターを使用して、開始時刻と終了時刻を特定することをお勧めします。

```
aws lightsail get-instance-metric-data \  
  --instance-name MEAN-1 \  
  --metric-name CPUUtilization \  
  --period 7200 \  
  --start-time 1571342400 \  
  --end-time 1571428800 \  
  --unit Percent \  
  --statistics Average
```

出力:

```
{  
  "metricName": "CPUUtilization",  
  "metricData": [  
    {  
      "average": 0.26113718770120725,  
      "timestamp": 1571342400.0,  
      "unit": "Percent"  
    },  
    {  
      "average": 0.26861268928111953,  
      "timestamp": 1571392800.0,  
      "unit": "Percent"  
    }  
  ]  
}
```

```
  },
  {
    "average": 0.28187475104748777,
    "timestamp": 1571378400.0,
    "unit": "Percent"
  },
  {
    "average": 0.2651936960458352,
    "timestamp": 1571421600.0,
    "unit": "Percent"
  },
  {
    "average": 0.2561856213712188,
    "timestamp": 1571371200.0,
    "unit": "Percent"
  },
  {
    "average": 0.3021383254607764,
    "timestamp": 1571356800.0,
    "unit": "Percent"
  },
  {
    "average": 0.2618381649223539,
    "timestamp": 1571407200.0,
    "unit": "Percent"
  },
  {
    "average": 0.26331929394825787,
    "timestamp": 1571400000.0,
    "unit": "Percent"
  },
  {
    "average": 0.2576348407007818,
    "timestamp": 1571385600.0,
    "unit": "Percent"
  },
  {
    "average": 0.2513008454658378,
    "timestamp": 1571364000.0,
    "unit": "Percent"
  },
  {
    "average": 0.26329974562758346,
    "timestamp": 1571414400.0,
```

```
        "unit": "Percent"
      },
      {
        "average": 0.2667092536656445,
        "timestamp": 1571349600.0,
        "unit": "Percent"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceMetricData](#)」の「」を参照してください。

get-instance-port-states

次のコード例は、`get-instance-port-states` を使用する方法を示しています。

AWS CLI

インスタンスのファイアウォール情報を取得するには

次の`get-instance-port-states`例では、インスタンス用に設定されたファイアウォールポートを返しますMEAN-1。

```
aws lightsail get-instance-port-states \
  --instance-name MEAN-1
```

出力:

```
{
  "portStates": [
    {
      "fromPort": 80,
      "toPort": 80,
      "protocol": "tcp",
      "state": "open"
    },
    {
      "fromPort": 22,
      "toPort": 22,
      "protocol": "tcp",
      "state": "open"
    }
  ]
}
```



```
    },
    {
      "fromPort": 443,
      "toPort": 443,
      "protocol": "tcp",
      "state": "open"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstancePortStates](#)」の「」を参照してください。

get-instance-snapshot

次の例は、get-instance-snapshot を使用する方法を説明しています。

AWS CLI

指定されたインスタンススナップショットに関する情報を取得するには

次のget-instance-snapshot例では、指定されたインスタンススナップショットの詳細を表示します。

```
aws lightsail get-instance-snapshot \
  --instance-snapshot-name MEAN-1-1571419854
```

出力:

```
{
  "instanceSnapshot": {
    "name": "MEAN-1-1571419854",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
    "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
    "createdAt": 1571419891.927,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "InstanceSnapshot",
    "tags": [],
```

```
    "state": "available",
    "fromAttachedDisks": [],
    "fromInstanceName": "MEAN-1",
    "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
    "fromBlueprintId": "mean",
    "fromBundleId": "medium_3_0",
    "isFromAutoSnapshot": false,
    "sizeInGb": 80
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceSnapshot](#)」の「」を参照してください。

get-instance-snapshots

次の例は、get-instance-snapshots を使用する方法を説明しています。

AWS CLI

すべてのインスタンススナップショットに関する情報を取得するには

次のget-instance-snapshots例では、設定された AWS リージョン内のすべてのインスタンススナップショットの詳細を表示します。

```
aws lightsail get-instance-snapshots
```

出力:

```
{
  "instanceSnapshots": [
    {
      "name": "MEAN-1-1571421498",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
a20e6ebe-b0ee-4ae4-a750-3EXAMPLEcb0c",
      "supportCode": "6EXAMPLE3362/ami-0EXAMPLEEe33cabfa1",
      "createdAt": 1571421527.755,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
    },
  ],
}
```

```
    "resourceType": "InstanceSnapshot",
    "tags": [
      {
        "key": "no_delete"
      }
    ],
    "state": "available",
    "fromAttachedDisks": [],
    "fromInstanceName": "MEAN-1",
    "fromInstanceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/1761aa0a-6038-4f25-8b94-2EXAMPLE19fd",
    "fromBlueprintId": "wordpress",
    "fromBundleId": "micro_3_0",
    "isFromAutoSnapshot": false,
    "sizeInGb": 40
  },
  {
    "name": "MEAN-1-1571419854",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
    "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
    "createdAt": 1571419891.927,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "InstanceSnapshot",
    "tags": [],
    "state": "available",
    "fromAttachedDisks": [],
    "fromInstanceName": "MEAN-1",
    "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEebada",
    "fromBlueprintId": "mean",
    "fromBundleId": "medium_3_0",
    "isFromAutoSnapshot": false,
    "sizeInGb": 80
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceSnapshots](#)」の「」を参照してください。

get-instance-state

次の例は、get-instance-state を使用する方法を説明しています。

AWS CLI

インスタンスの状態に関する情報を取得するには

次のget-instance-state例では、指定されたインスタンスの状態を返します。

```
aws lightsail get-instance-state \  
  --instance-name MEAN-1
```

出力:

```
{  
  "state": {  
    "code": 16,  
    "name": "running"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstanceState](#)」の「」を参照してください。

get-instance

次のコード例は、get-instance を使用する方法を示しています。

AWS CLI

インスタンスに関する情報を取得するには

次のget-instance例では、インスタンスの詳細を表示しますMEAN-1。

```
aws lightsail get-instance \  
  --instance-name MEAN-1
```

出力:

```
{  
  "instance": {
```

```
    "name": "MEAN-1",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-EXAMPLE4bada",
    "supportCode": "6EXAMPLE3362/i-05EXAMPLE407c97d3",
    "createdAt": 1570635023.124,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "Instance",
    "tags": [],
    "blueprintId": "mean",
    "blueprintName": "MEAN",
    "bundleId": "medium_3_0",
    "isStaticIp": false,
    "privateIpAddress": "192.0.2.0",
    "publicIpAddress": "192.0.2.0",
    "hardware": {
      "cpuCount": 2,
      "disks": [
        {
          "createdAt": 1570635023.124,
          "sizeInGb": 80,
          "isSystemDisk": true,
          "iops": 240,
          "path": "/dev/xvda",
          "attachedTo": "MEAN-1",
          "attachmentState": "attached"
        }
      ],
      "ramSizeInGb": 4.0
    },
    "networking": {
      "monthlyTransfer": {
        "gbPerMonthAllocated": 4096
      },
      "ports": [
        {
          "fromPort": 80,
          "toPort": 80,
          "protocol": "tcp",
          "accessFrom": "Anywhere (0.0.0.0/0)",
          "accessType": "public",
          "commonName": ""
        }
      ]
    }
  }
}
```

```
        "accessDirection": "inbound"
      },
      {
        "fromPort": 22,
        "toPort": 22,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
      },
      {
        "fromPort": 443,
        "toPort": 443,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
      }
    ]
  },
  "state": {
    "code": 16,
    "name": "running"
  },
  "username": "bitnami",
  "sshKeyName": "MyKey"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstance](#)」の「」を参照してください。

get-instances

次のコード例は、get-instances を使用する方法を示しています。

AWS CLI

すべてのインスタンスに関する情報を取得するには

次のget-instances例では、設定された AWS リージョン内のすべてのインスタスの詳細を表示します。

```
aws lightsail get-instances
```

出力:

```
{
  "instances": [
    {
      "name": "Windows_Server_2022-1",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/0f44fbb9-8f55-4e47-a25e-EXAMPLE04763",
      "supportCode": "62EXAMPLE362/i-0bEXAMPLE71a686b9",
      "createdAt": 1571332358.665,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "resourceType": "Instance",
      "tags": [],
      "blueprintId": "windows_server_2022",
      "blueprintName": "Windows Server 2022",
      "bundleId": "large_win_3_0",
      "isStaticIp": false,
      "privateIpAddress": "192.0.2.0",
      "publicIpAddress": "192.0.2.0",
      "hardware": {
        "cpuCount": 1,
        "disks": [
          {
            "createdAt": 1571332358.665,
            "sizeInGb": 160,
            "isSystemDisk": true,
            "iops": 180,
            "path": "/dev/sda1",
            "attachedTo": "Windows_Server_2022-1",
            "attachmentState": "attached"
          },
          {
            "name": "my-disk-for-windows-server",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/4123a81c-484c-49ea-afea-5EXAMPLEda87",
```

```
    "supportCode": "6EXAMPLE3362/vol-0EXAMPLEb2b99ca3d",
    "createdAt": 1571355063.494,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "Disk",
    "tags": [],
    "sizeInGb": 128,
    "isSystemDisk": false,
    "iops": 384,
    "path": "/dev/xvdf",
    "state": "in-use",
    "attachedTo": "Windows_Server_2022-1",
    "isAttached": true,
    "attachmentState": "attached"
  }
],
"ramSizeInGb": 8.0
},
"networking": {
  "monthlyTransfer": {
    "gbPerMonthAllocated": 3072
  },
  "ports": [
    {
      "fromPort": 80,
      "toPort": 80,
      "protocol": "tcp",
      "accessFrom": "Anywhere (0.0.0.0/0)",
      "accessType": "public",
      "commonName": "",
      "accessDirection": "inbound"
    },
    {
      "fromPort": 22,
      "toPort": 22,
      "protocol": "tcp",
      "accessFrom": "Anywhere (0.0.0.0/0)",
      "accessType": "public",
      "commonName": "",
      "accessDirection": "inbound"
    }
  ],
  {
```



```

        "fromPort": 3389,
        "toPort": 3389,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound"
    }
]
},
"state": {
    "code": 16,
    "name": "running"
},
"username": "Administrator",
"sshKeyName": "LightsailDefaultKeyPair"
},
{
    "name": "MEAN-1",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-8EXAMPLEbada",
    "supportCode": "6EXAMPLE3362/i-0EXAMPLEa407c97d3",
    "createdAt": 1570635023.124,
    "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
    },
    "resourceType": "Instance",
    "tags": [],
    "blueprintId": "mean",
    "blueprintName": "MEAN",
    "bundleId": "medium_3_0",
    "isStaticIp": false,
    "privateIpAddress": "192.0.2.0",
    "publicIpAddress": "192.0.2.0",
    "hardware": {
        "cpuCount": 2,
        "disks": [
            {
                "name": "Disk-1",
                "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
                "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
                "createdAt": 1566585439.587,

```

```
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Disk",
        "tags": [
            {
                "key": "test"
            }
        ],
        "sizeInGb": 8,
        "isSystemDisk": false,
        "iops": 240,
        "path": "/dev/xvdf",
        "state": "in-use",
        "attachedTo": "MEAN-1",
        "isAttached": true,
        "attachmentState": "attached"
    },
    {
        "createdAt": 1570635023.124,
        "sizeInGb": 80,
        "isSystemDisk": true,
        "iops": 240,
        "path": "/dev/sda1",
        "attachedTo": "MEAN-1",
        "attachmentState": "attached"
    }
],
"ramSizeInGb": 4.0
},
"networking": {
    "monthlyTransfer": {
        "gbPerMonthAllocated": 4096
    },
    "ports": [
        {
            "fromPort": 80,
            "toPort": 80,
            "protocol": "tcp",
            "accessFrom": "Anywhere (0.0.0.0/0)",
            "accessType": "public",
            "commonName": "",
            "accessDirection": "inbound"
        }
    ]
}
```

```
    },
    {
      "fromPort": 22,
      "toPort": 22,
      "protocol": "tcp",
      "accessFrom": "Anywhere (0.0.0.0/0)",
      "accessType": "public",
      "commonName": "",
      "accessDirection": "inbound"
    },
    {
      "fromPort": 443,
      "toPort": 443,
      "protocol": "tcp",
      "accessFrom": "Anywhere (0.0.0.0/0)",
      "accessType": "public",
      "commonName": "",
      "accessDirection": "inbound"
    }
  ]
},
"state": {
  "code": 16,
  "name": "running"
},
"username": "bitnami",
"sshKeyName": "MyTestKey"
}
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInstance](#)」の「」を参照してください。

get-key-pair

次のコード例は、get-key-pair を使用する方法を示しています。

AWS CLI

キーペアに関する情報を取得するには

次のget-key-pair例では、指定されたキーペアの詳細を表示します。

```
aws lightsail get-key-pair \  
  --key-pair-name MyKey1
```

出力:

```
{  
  "keyPair": {  
    "name": "MyKey1",  
    "arn": "arn:aws:lightsail:us-  
west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",  
    "supportCode": "6EXAMPLE3362/MyKey1",  
    "createdAt": 1571255026.975,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "KeyPair",  
    "tags": [],  
    "fingerprint": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetKeyPair](#)」の「」を参照してください。

get-key-pairs

次の例は、get-key-pairs を使用する方法を説明しています。

AWS CLI

すべてのキーペアに関する情報を取得するには

次のget-key-pairs例では、設定された AWS リージョンのすべてのキーペアの詳細を表示します。

```
aws lightsail get-key-pairs
```

出力:

```
{
```

```
"keyPairs": [
  {
    "name": "MyKey1",
    "arn": "arn:aws:lightsail:us-
west-2:111122223333:KeyPair/19a4efdf-3054-43d6-91fd-eEXAMPLE21bf",
    "supportCode": "6EXAMPLE3362/MyKey1",
    "createdAt": 1571255026.975,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "KeyPair",
    "tags": [],
    "fingerprint":
"00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:gg:hh:ii:jj"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetKeyPairs](#)」の「」を参照してください。

get-load-balancer-tls-certificates

次のコード例は、`get-load-balancer-tls-certificates` を使用する方法を示しています。

AWS CLI

ロードバランサーのTLS証明書に関する情報を取得するには

次の`get-load-balancer-tls-certificates`例では、指定されたロードバランサーのTLS証明書の詳細を表示します。

```
aws lightsail get-load-balancer-tls-certificates \
  --load-balancer-name LoadBalancer-1
```

出力:

```
{
  "tlsCertificates": [
    {
```

```

    "name": "example-com",
    "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancerTlsCertificate/d7bf4643-6a02-4cd4-b3c4-
fEXAMPLE9b4d",
    "supportCode": "6EXAMPLE3362/arn:aws:acm:us-
west-2:333322221111:certificate/9af8e32c-a54e-4a67-8c63-cEXAMPLEb314",
    "createdAt": 1571678025.3,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "LoadBalancerTlsCertificate",
    "loadBalancerName": "LoadBalancer-1",
    "isAttached": false,
    "status": "ISSUED",
    "domainName": "example.com",
    "domainValidationRecords": [
      {
        "name": "_dEXAMPLE4ede046a0319eb44a4eb3cbc.example.com.",
        "type": "CNAME",
        "value": "_bEXAMPLE0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws.",
        "validationStatus": "SUCCESS",
        "domainName": "example.com"
      }
    ],
    "issuedAt": 1571678070.0,
    "issuer": "Amazon",
    "keyAlgorithm": "RSA-2048",
    "notAfter": 1605960000.0,
    "notBefore": 1571616000.0,
    "serial": "00:11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff",
    "signatureAlgorithm": "SHA256WITHRSA",
    "subject": "CN=example.com",
    "subjectAlternativeNames": [
      "example.com"
    ]
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLoadBalancerTlsCertificates](#)」の「」を参照してください。

get-load-balancer

次のコード例は、`get-load-balancer` を使用する方法を示しています。

AWS CLI

ロードバランサーに関する情報を取得するには

次の`get-load-balancer`例では、指定されたロードバランサーの詳細を表示します。

```
aws lightsail get-load-balancer \  
  --load-balancer-name LoadBalancer-1
```

出力:

```
{  
  "loadBalancer": {  
    "name": "LoadBalancer-1",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",  
    "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-west-2:333322221111:loadbalancer/app/bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",  
    "createdAt": 1571677906.723,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "LoadBalancer",  
    "tags": [],  
    "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-west-2.elb.amazonaws.com",  
    "state": "active",  
    "protocol": "HTTP",  
    "publicPorts": [  
      80  
    ],  
    "healthCheckPath": "/",  
    "instancePort": 80,  
    "instanceHealthSummary": [  
      {  
        "instanceName": "MEAN-3",  
        "instanceHealth": "healthy"  
      }  
    ],  
  }  
}
```

```
{
  {
    "instanceName": "MEAN-1",
    "instanceHealth": "healthy"
  },
  {
    "instanceName": "MEAN-2",
    "instanceHealth": "healthy"
  }
],
"tlsCertificateSummaries": [
  {
    "name": "example-com",
    "isAttached": false
  }
],
"configurationOptions": {
  "SessionStickinessEnabled": "false",
  "SessionStickiness_LB_CookieDurationSeconds": "86400"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLoadBalancer](#)」の「」を参照してください。

get-load-balancers

次のコード例は、get-load-balancers を使用する方法を示しています。

AWS CLI

すべてのロードバランサーに関する情報を取得するには

次のget-load-balancers例では、設定された AWS リージョン内のすべてのロードバランサーの詳細を表示します。

```
aws lightsail get-load-balancers
```

出力:

```
{
  "loadBalancers": [
```



```
{
  "name": "LoadBalancer-1",
  "arn": "arn:aws:lightsail:us-
west-2:111122223333:LoadBalancer/40486b2b-1ad0-4152-83e4-cEXAMPLE6f4b",
  "supportCode": "6EXAMPLE3362/arn:aws:elasticloadbalancing:us-
west-2:333322221111:loadbalancer/app/
bEXAMPLE128cb59d86f946a9395dd304/1EXAMPLE8dd9d77e",
  "createdAt": 1571677906.723,
  "location": {
    "availabilityZone": "all",
    "regionName": "us-west-2"
  },
  "resourceType": "LoadBalancer",
  "tags": [],
  "dnsName": "bEXAMPLE128cb59d86f946a9395dd304-1486911371.us-
west-2.elb.amazonaws.com",
  "state": "active",
  "protocol": "HTTP",
  "publicPorts": [
    80
  ],
  "healthCheckPath": "/",
  "instancePort": 80,
  "instanceHealthSummary": [
    {
      "instanceName": "MEAN-3",
      "instanceHealth": "healthy"
    },
    {
      "instanceName": "MEAN-1",
      "instanceHealth": "healthy"
    },
    {
      "instanceName": "MEAN-2",
      "instanceHealth": "healthy"
    }
  ],
  "tlsCertificateSummaries": [
    {
      "name": "example-com",
      "isAttached": false
    }
  ],
  "configurationOptions": {
```

```
        "SessionStickinessEnabled": "false",
        "SessionStickiness_LB_CookieDurationSeconds": "86400"
    }
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLoadBalancers](#)」の「」を参照してください。

get-operation

次のコード例は、get-operation を使用する方法を示しています。

AWS CLI

単一のオペレーションに関する情報を取得するには

次のget-operation例では、指定されたオペレーションの詳細を表示します。

```
aws lightsail get-operation \  
--operation-id e5700e8a-daf2-4b49-bc01-3EXAMPLE910a
```

出力:

```
{  
  "operation": {  
    "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",  
    "resourceName": "Instance-1",  
    "resourceType": "Instance",  
    "createdAt": 1571679872.404,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": true,  
    "operationType": "CreateInstance",  
    "status": "Succeeded",  
    "statusChangedAt": 1571679890.304  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOperation](#)」の「」を参照してください。

get-operations-for-resource

次の例は、get-operations-for-resource を使用する方法を説明しています。

AWS CLI

リソースのすべてのオペレーションを取得するには

次のget-operations-for-resource例では、指定されたリソースのすべてのオペレーションの詳細を表示します。

```
aws lightsail get-operations-for-resource \  
  --resource-name LoadBalancer-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",  
      "resourceName": "LoadBalancer-1",  
      "resourceType": "LoadBalancer",  
      "createdAt": 1571678786.071,  
      "location": {  
        "availabilityZone": "all",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": true,  
      "operationDetails": "MEAN-1",  
      "operationType": "DetachInstancesFromLoadBalancer",  
      "status": "Succeeded",  
      "statusChangedAt": 1571679087.57  
    },  
    {  
      "id": "2d742a18-0e7f-48c8-9705-3EXAMPLEf98a",  
      "resourceName": "LoadBalancer-1",  
      "resourceType": "LoadBalancer",  
      "createdAt": 1571678782.784,  
      "location": {
```

```
        "availabilityZone": "all",
        "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "MEAN-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571678798.465
},
{
    "id": "6c700fcc-4246-40ab-952b-1EXAMPLEdac2",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571678775.297,
    "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "MEAN-3",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571678842.806
},
...
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOperationsForResource](#)」の「」を参照してください。

get-operations

次の例は、get-operations を使用する方法を説明しています。

AWS CLI

すべてのオペレーションに関する情報を取得するには

次のget-operations例では、設定された AWS リージョンのすべてのオペレーションの詳細を表示します。

aws lightsail get-operations

出力:

```
{
  "operations": [
    {
      "id": "e5700e8a-daf2-4b49-bc01-3EXAMPLE910a",
      "resourceName": "Instance-1",
      "resourceType": "Instance",
      "createdAt": 1571679872.404,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationType": "CreateInstance",
      "status": "Succeeded",
      "statusChangedAt": 1571679890.304
    },
    {
      "id": "701a3339-930e-4914-a9f9-7EXAMPLE68d7",
      "resourceName": "WordPress-1",
      "resourceType": "Instance",
      "createdAt": 1571678786.072,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "LoadBalancer-1",
      "operationType": "DetachInstancesFromLoadBalancer",
      "status": "Succeeded",
      "statusChangedAt": 1571679086.399
    },
    {
      "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
      "resourceName": "LoadBalancer-1",
      "resourceType": "LoadBalancer",
      "createdAt": 1571678786.071,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      }
    }
  ]
}
```

```
    },
    "isTerminal": true,
    "operationDetails": "WordPress-1",
    "operationType": "DetachInstancesFromLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571679087.57
  },
  ...
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOperations](#)」の「」を参照してください。

get-regions

次の例は、get-regions を使用する方法を説明しています。

AWS CLI

Amazon Lightsail のすべての AWS リージョンを取得するには

次のget-regions例では、Amazon Lightsail のすべての AWS リージョンの詳細を表示します。

```
aws lightsail get-regions
```

出力:

```
{
  "regions": [
    {
      "continentCode": "NA",
      "description": "This region is recommended to serve users in the eastern
United States",
      "displayName": "Virginia",
      "name": "us-east-1",
      "availabilityZones": [],
      "relationalDatabaseAvailabilityZones": []
    },
    {
      "continentCode": "NA",
```

```
        "description": "This region is recommended to serve users in the eastern
United States",
        "displayName": "Ohio",
        "name": "us-east-2",
        "availabilityZones": [],
        "relationalDatabaseAvailabilityZones": []
    },
    {
        "continentCode": "NA",
        "description": "This region is recommended to serve users in the
northwestern United States, Alaska, and western Canada",
        "displayName": "Oregon",
        "name": "us-west-2",
        "availabilityZones": [],
        "relationalDatabaseAvailabilityZones": []
    },
    ...
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRegions](#)」の「」を参照してください。

get-relational-database-blueprints

次の例は、`get-relational-database-blueprints` を使用する方法を説明しています。

AWS CLI

新しいリレーショナルデータベースのブループリントを取得するには

次の`get-relational-database-blueprints`例では、Amazon Lightsail で新しいリレーショナルデータベースを作成するために使用できるすべてのリレーショナルデータベース設計図の詳細を表示します。

```
aws lightsail get-relational-database-blueprints
```

出力:

```
{
  "blueprints": [
```

```
{
  "blueprintId": "mysql_5_6",
  "engine": "mysql",
  "engineVersion": "5.6.44",
  "engineDescription": "MySQL Community Edition",
  "engineVersionDescription": "MySQL 5.6.44",
  "isEngineDefault": false
},
{
  "blueprintId": "mysql_5_7",
  "engine": "mysql",
  "engineVersion": "5.7.26",
  "engineDescription": "MySQL Community Edition",
  "engineVersionDescription": "MySQL 5.7.26",
  "isEngineDefault": true
},
{
  "blueprintId": "mysql_8_0",
  "engine": "mysql",
  "engineVersion": "8.0.16",
  "engineDescription": "MySQL Community Edition",
  "engineVersionDescription": "MySQL 8.0.16",
  "isEngineDefault": false
},
{
  "blueprintId": "postgres_9_6",
  "engine": "postgres",
  "engineVersion": "9.6.15",
  "engineDescription": "PostgreSQL",
  "engineVersionDescription": "PostgreSQL 9.6.15-R1",
  "isEngineDefault": false
},
{
  "blueprintId": "postgres_10",
  "engine": "postgres",
  "engineVersion": "10.10",
  "engineDescription": "PostgreSQL",
  "engineVersionDescription": "PostgreSQL 10.10-R1",
  "isEngineDefault": false
},
{
  "blueprintId": "postgres_11",
  "engine": "postgres",
  "engineVersion": "11.5",
```



```
        "engineDescription": "PostgreSQL",
        "engineVersionDescription": "PostgreSQL 11.5-R1",
        "isEngineDefault": true
    }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabaseBlueprints](#)」の「」を参照してください。

get-relational-database-bundles

次のコード例は、get-relational-database-bundles を使用する方法を示しています。

AWS CLI

新しいリレーショナルデータベースのバンドルを取得するには

次のget-relational-database-bundles例では、Amazon Lightsail で新しいリレーショナルデータベースを作成するために使用できるすべてのリレーショナルデータベースバンドルの詳細を表示します。--include-inactive フラグが コマンドで指定されていないため、レスポンスには非アクティブなバンドルが含まれていないことに注意してください。非アクティブなバンドルを使用して新しいリレーショナルデータベースを作成することはできません。

```
aws lightsail get-relational-database-bundles
```

出力:

```
{
  "bundles": [
    {
      "bundleId": "micro_2_0",
      "name": "Micro",
      "price": 15.0,
      "ramSizeInGb": 1.0,
      "diskSizeInGb": 40,
      "transferPerMonthInGb": 100,
      "cpuCount": 2,
      "isEncrypted": true,
      "isActive": true
    },
    {
```

```
    "bundleId": "micro_ha_2_0",
    "name": "Micro with High Availability",
    "price": 30.0,
    "ramSizeInGb": 1.0,
    "diskSizeInGb": 40,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "small_2_0",
    "name": "Small",
    "price": 30.0,
    "ramSizeInGb": 2.0,
    "diskSizeInGb": 80,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "small_ha_2_0",
    "name": "Small with High Availability",
    "price": 60.0,
    "ramSizeInGb": 2.0,
    "diskSizeInGb": 80,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "medium_2_0",
    "name": "Medium",
    "price": 60.0,
    "ramSizeInGb": 4.0,
    "diskSizeInGb": 120,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
```

```
    "bundleId": "medium_ha_2_0",
    "name": "Medium with High Availability",
    "price": 120.0,
    "ramSizeInGb": 4.0,
    "diskSizeInGb": 120,
    "transferPerMonthInGb": 100,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "large_2_0",
    "name": "Large",
    "price": 115.0,
    "ramSizeInGb": 8.0,
    "diskSizeInGb": 240,
    "transferPerMonthInGb": 200,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  },
  {
    "bundleId": "large_ha_2_0",
    "name": "Large with High Availability",
    "price": 230.0,
    "ramSizeInGb": 8.0,
    "diskSizeInGb": 240,
    "transferPerMonthInGb": 200,
    "cpuCount": 2,
    "isEncrypted": true,
    "isActive": true
  }
]
}
```

詳細については、[Amazon Lightsail デベロッパーガイド](#)の「[Amazon Lightsail でのデータベースの作成](#)」を参照してください。Amazon Lightsail

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabaseBundles](#)」の「」を参照してください。

get-relational-database-events

次の例は、get-relational-database-events を使用する方法を説明しています。

AWS CLI

リレーショナルデータベースのイベントを取得するには

次のget-relational-database-events例では、指定されたリレーショナルデータベースの過去 17 時間 (1020 分) のイベントの詳細を表示します。

```
aws lightsail get-relational-database-events \  
  --relational-database-name Database-1 \  
  --duration-in-minutes 1020
```

出力:

```
{  
  "relationalDatabaseEvents": [  
    {  
      "resource": "Database-1",  
      "createdAt": 1571654146.553,  
      "message": "Backing up Relational Database",  
      "eventCategories": [  
        "backup"  
      ]  
    },  
    {  
      "resource": "Database-1",  
      "createdAt": 1571654249.98,  
      "message": "Finished Relational Database backup",  
      "eventCategories": [  
        "backup"  
      ]  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabaseEvents](#)」の「」を参照してください。

get-relational-database-log-events

次の例は、get-relational-database-log-events を使用する方法を説明しています。

AWS CLI

リレーショナルデータベースのログイベントを取得するには

次のget-relational-database-log-events例では、リレーショナルデータベース 1571597176 の 1570733176 と の間の指定されたログの詳細を表示します Database1。返される情報は、 から開始するように設定されています head。

Unix タイムコンバーターを使用して、開始時刻と終了時刻を特定することをお勧めします。

```
aws lightsail get-relational-database-log-events \  
  --relational-database-name Database1 \  
  --log-stream-name error \  
  --start-from-head \  
  --start-time 1570733176 \  
  --end-time 1571597176
```

出力:

```
{  
  "resourceLogEvents": [  
    {  
      "createdAt": 1570820267.0,  
      "message": "2019-10-11 18:57:47 20969 [Warning] IP address '192.0.2.0'  
could not be resolved: Name or service not known"  
    },  
    {  
      "createdAt": 1570860974.0,  
      "message": "2019-10-12 06:16:14 20969 [Warning] IP address '8192.0.2.0'  
could not be resolved: Temporary failure in name resolution"  
    },  
    {  
      "createdAt": 1570860977.0,  
      "message": "2019-10-12 06:16:17 20969 [Warning] IP address '192.0.2.0'  
could not be resolved: Temporary failure in name resolution"  
    },  
    {  
      "createdAt": 1570860979.0,
```

```

    "message": "2019-10-12 06:16:19 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
  },
  {
    "createdAt": 1570860981.0,
    "message": "2019-10-12 06:16:21 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
  },
  {
    "createdAt": 1570860982.0,
    "message": "2019-10-12 06:16:22 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
  },
  {
    "createdAt": 1570860984.0,
    "message": "2019-10-12 06:16:24 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
  },
  {
    "createdAt": 1570860986.0,
    "message": "2019-10-12 06:16:26 20969 [Warning] IP address '192.0.2.0'
could not be resolved: Temporary failure in name resolution"
  },
  ...
}
],
"nextBackwardToken":
"eEXAMPLEZXJUZXh0IjoiZnRwb3F3cUpRS1Q5NndMYThxelRUZlFhR3J6c2dKWEEvM2kvajZMZzVWVWpqRDN0YjFXTj
"nextForwardToken":
"eEXAMPLEZXJUZXh0IjoiT09Lb0Z6ZFRJbHhaNEQ5N2tPbkkwRmwwNUxPZjFTbFFwUk1Qbz1SaWgvMwVXbEk4aG56VH
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetRelationalDatabaseLogEvents](#)」の「」を参照してください。

get-relational-database-log-streams

次の例は、get-relational-database-log-streams を使用する方法を説明しています。

AWS CLI

リレーショナルデータベースのログストリームを取得するには

次のget-relational-database-log-streams例では、指定されたリレーショナルデータベースで使用できるすべてのログストリームを返します。

```
aws lightsail get-relational-database-log-streams \  
--relational-database-name Database1
```

出力:

```
{  
  "logStreams": [  
    "audit",  
    "error",  
    "general",  
    "slowquery"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabaseLogStreams](#)」の「」を参照してください。

get-relational-database-master-user-password

次のコード例は、get-relational-database-master-user-password を使用する方法を示しています。

AWS CLI

リレーショナルデータベースのマスターユーザーパスワードを取得するには

次のget-relational-database-master-user-password例では、指定されたリレーショナルデータベースのマスターユーザーパスワードに関する情報を返します。

```
aws lightsail get-relational-database-master-user-password \  
--relational-database-name Database-1
```

出力:

```
{  
  "masterUserPassword": "VEXAMPLEec.9qvx,_t<)Wkf)kwboM,>2",  
  "createdAt": 1571259453.959  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetRelationalDatabaseMasterUserPassword](#)」の「」を参照してください。

get-relational-database-metric-data

次のコード例は、get-relational-database-metric-data を使用する方法を示しています。

AWS CLI

リレーショナルデータベースのメトリクスデータを取得するには

次のget-relational-database-metric-data例では、1571597176リレーショナルデータベースの 1570733176と の間の 24 時間 (86400秒) DatabaseConnectionsにわたるメトリクスのカウント合計を返しますDatabase1。

Unix タイムコンバーターを使用して、開始時刻と終了時刻を特定することをお勧めします。

```
aws lightsail get-relational-database-metric-data \  
  --relational-database-name Database1 \  
  --metric-name DatabaseConnections \  
  --period 86400 \  
  --start-time 1570733176 \  
  --end-time 1571597176 \  
  --unit Count \  
  --statistics Sum
```

出力:

```
{  
  "metricName": "DatabaseConnections",  
  "metricData": [  
    {  
      "sum": 1.0,  
      "timestamp": 1571510760.0,  
      "unit": "Count"  
    },  
    {  
      "sum": 1.0,  
      "timestamp": 1570733160.0,  
      "unit": "Count"  
    },  
    {
```



```
    "sum": 1.0,  
    "timestamp": 1570992360.0,  
    "unit": "Count"  
  },  
  {  
    "sum": 0.0,  
    "timestamp": 1571251560.0,  
    "unit": "Count"  
  },  
  {  
    "sum": 721.0,  
    "timestamp": 1570819560.0,  
    "unit": "Count"  
  },  
  {  
    "sum": 1.0,  
    "timestamp": 1571078760.0,  
    "unit": "Count"  
  },  
  {  
    "sum": 2.0,  
    "timestamp": 1571337960.0,  
    "unit": "Count"  
  },  
  {  
    "sum": 684.0,  
    "timestamp": 1570905960.0,  
    "unit": "Count"  
  },  
  {  
    "sum": 0.0,  
    "timestamp": 1571165160.0,  
    "unit": "Count"  
  },  
  {  
    "sum": 1.0,  
    "timestamp": 1571424360.0,  
    "unit": "Count"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetRelationalDatabaseMetricData](#)」の「」を参照してください。

get-relational-database-parameters

次のコード例は、get-relational-database-parameters を使用する方法を示しています。

AWS CLI

リレーショナルデータベースのパラメータを取得するには

次のget-relational-database-parameters例では、指定されたリレーショナルデータベースで使用できるすべてのパラメータに関する情報を返します。

```
aws lightsail get-relational-database-parameters \  
--relational-database-name Database-1
```

出力:

```
{  
  "parameters": [  
    {  
      "allowedValues": "0,1",  
      "applyMethod": "pending-reboot",  
      "applyType": "dynamic",  
      "dataType": "boolean",  
      "description": "Automatically set all granted roles as active after the  
user has authenticated successfully.",  
      "isModifiable": true,  
      "parameterName": "activate_all_roles_on_login",  
      "parameterValue": "0"  
    },  
    {  
      "allowedValues": "0,1",  
      "applyMethod": "pending-reboot",  
      "applyType": "static",  
      "dataType": "boolean",  
      "description": "Controls whether user-defined functions that have only  
an xxx symbol for the main function can be loaded",  
      "isModifiable": false,  
      "parameterName": "allow-suspicious-udfs"  
    },  
  ],  
}
```

```
{
  "allowedValues": "0,1",
  "applyMethod": "pending-reboot",
  "applyType": "dynamic",
  "dataType": "boolean",
  "description": "Sets the autocommit mode",
  "isModifiable": true,
  "parameterName": "autocommit"
},
{
  "allowedValues": "0,1",
  "applyMethod": "pending-reboot",
  "applyType": "static",
  "dataType": "boolean",
  "description": "Controls whether the server autogenerates SSL key and
certificate files in the data directory, if they do not already exist.",
  "isModifiable": false,
  "parameterName": "auto_generate_certs"
},
...
}
]
```

詳細については、[「Lightsail 開発ガイド」の「Amazon Lightsail でのデータベースパラメータの更新」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabaseParameters](#)」の「」を参照してください。

get-relational-database-snapshot

次の例は、get-relational-database-snapshot を使用する方法を説明しています。

AWS CLI

リレーショナルデータベーススナップショットに関する情報を取得するには

次のget-relational-database-snapshot例では、指定されたリレーショナルデータベーススナップショットの詳細を表示します。

```
aws lightsail get-relational-database-snapshot \
```

```
--relational-database-snapshot-name Database-1-1571350042
```

出力:

```
{
  "relationalDatabaseSnapshot": {
    "name": "Database-1-1571350042",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9EXAMPLEaee3643d2",
    "supportCode": "6EXAMPLE3362/1s-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
    "createdAt": 1571350046.238,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "RelationalDatabaseSnapshot",
    "tags": [],
    "engine": "mysql",
    "engineVersion": "8.0.16",
    "sizeInGb": 40,
    "state": "available",
    "fromRelationalDatabaseName": "Database-1",
    "fromRelationalDatabaseArn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
    "fromRelationalDatabaseBundleId": "micro_1_0",
    "fromRelationalDatabaseBlueprintId": "mysql_8_0"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabaseSnapshot](#)」の「」を参照してください。

get-relational-database-snapshots

次のコード例は、get-relational-database-snapshots を使用する方法を示しています。

AWS CLI

すべてのリレーショナルデータベーススナップショットに関する情報を取得するには

次のget-relational-database-snapshots例では、設定された AWS リージョンのすべてのリレーショナルデータベーススナップショットの詳細を表示します。

aws lightsail get-relational-database-snapshots

出力:

```
{
  "relationalDatabaseSnapshots": [
    {
      "name": "Database-1-1571350042",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9861-6EXAMPLE43d2",
      "supportCode": "6EXAMPLE3362/1s-8EXAMPLE2ba7ad041451946fafc2ad19cfbd9eb2",
      "createdAt": 1571350046.238,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "RelationalDatabaseSnapshot",
      "tags": [],
      "engine": "mysql",
      "engineVersion": "8.0.16",
      "sizeInGb": 40,
      "state": "available",
      "fromRelationalDatabaseName": "Database-1",
      "fromRelationalDatabaseArn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
      "fromRelationalDatabaseBundleId": "micro_1_0",
      "fromRelationalDatabaseBlueprintId": "mysql_8_0"
    },
    {
      "name": "Database1-Console",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabaseSnapshot/8b94136e-06ec-4b1a-a3fb-5EXAMPLEe1e9",
      "supportCode": "6EXAMPLE3362/1s-9EXAMPLE14b000d34c8d1c432734e137612d5b5c",
      "createdAt": 1571249981.025,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "RelationalDatabaseSnapshot",
      "tags": [
```

```
        {
            "key": "test"
        }
    ],
    "engine": "mysql",
    "engineVersion": "5.6.44",
    "sizeInGb": 40,
    "state": "available",
    "fromRelationalDatabaseName": "Database1",
    "fromRelationalDatabaseArn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/a6161cb7-4535-4f16-9dcf-8EXAMPLE3d4e",
    "fromRelationalDatabaseBundleId": "micro_1_0",
    "fromRelationalDatabaseBlueprintId": "mysql_5_6"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabaseSnapshots](#)」の「」を参照してください。

get-relational-database

次のコード例は、get-relational-database を使用する方法を示しています。

AWS CLI

リレーショナルデータベースに関する情報を取得するには

次のget-relational-database例では、指定されたリレーショナルデータベースの詳細を表示します。

```
aws lightsail get-relational-database \
  --relational-database-name Database-1
```

出力:

```
{
  "relationalDatabase": {
    "name": "Database-1",
    "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
```

```
"supportCode": "6EXAMPLE3362/1s-9EXAMPLE8ad863723b62cc8901a8aa6e794ae0d2",
"createdAt": 1571259453.795,
"location": {
  "availabilityZone": "us-west-2a",
  "regionName": "us-west-2"
},
"resourceType": "RelationalDatabase",
"tags": [],
"relationalDatabaseBlueprintId": "mysql_8_0",
"relationalDatabaseBundleId": "micro_1_0",
"masterDatabaseName": "dbmaster",
"hardware": {
  "cpuCount": 1,
  "diskSizeInGb": 40,
  "ramSizeInGb": 1.0
},
"state": "available",
"backupRetentionEnabled": false,
"pendingModifiedValues": {},
"engine": "mysql",
"engineVersion": "8.0.16",
"masterUsername": "dbmasteruser",
"parameterApplyStatus": "in-sync",
"preferredBackupWindow": "10:01-10:31",
"preferredMaintenanceWindow": "sat:11:14-sat:11:44",
"publiclyAccessible": true,
"masterEndpoint": {
  "port": 3306,
  "address": "1s-9EXAMPLE8ad863723b62ccEXAMPLEa6e794ae0d2.czowadgeezqi.us-
west-2.rds.amazonaws.com"
},
"pendingMaintenanceActions": []
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabase](#)」の「」を参照してください。

get-relational-databases

次のコード例は、get-relational-databases を使用する方法を示しています。

AWS CLI

すべてのリレーショナルデータベースに関する情報を取得するには

次の`get-relational-databases`例では、設定された AWS リージョン内のすべてのリレーショナルデータベースの詳細を表示します。

```
aws lightsail get-relational-databases
```

出力:

```
{
  "relationalDatabases": [
    {
      "name": "MySQL",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/8529020c-3ab9-4d51-92af-5EXAMPLE8979",
      "supportCode": "6EXAMPLE3362/1s-3EXAMPLEa995d8c3b06b4501356e5f2f28e1aeba",
      "createdAt": 1554306019.155,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "resourceType": "RelationalDatabase",
      "tags": [],
      "relationalDatabaseBlueprintId": "mysql_8_0",
      "relationalDatabaseBundleId": "micro_1_0",
      "masterDatabaseName": "dbmaster",
      "hardware": {
        "cpuCount": 1,
        "diskSizeInGb": 40,
        "ramSizeInGb": 1.0
      },
      "state": "available",
      "backupRetentionEnabled": true,
      "pendingModifiedValues": {},
      "engine": "mysql",
      "engineVersion": "8.0.15",
      "latestRestorableTime": 1571686200.0,
      "masterUsername": "dbmasteruser",
      "parameterApplyStatus": "in-sync",
      "preferredBackupWindow": "07:51-08:21",
```



```
    "preferredMaintenanceWindow": "tue:12:18-tue:12:48",
    "publiclyAccessible": true,
    "masterEndpoint": {
      "port": 3306,
      "address":
"ls-3EXAMPLEa995d8c3b06b4501356e5f2fEXAMPLEa.czowadgeezqi.us-
west-2.rds.amazonaws.com"
    },
    "pendingMaintenanceActions": []
  },
  {
    "name": "Postgres",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/
e9780b6b-d0ab-4af2-85f1-1EXAMPLEac68",
    "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEb4ffffb5cec056220c734713e14bd5fcd",
    "createdAt": 1554306000.814,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "resourceType": "RelationalDatabase",
    "tags": [],
    "relationalDatabaseBlueprintId": "postgres_11",
    "relationalDatabaseBundleId": "micro_1_0",
    "masterDatabaseName": "dbmaster",
    "hardware": {
      "cpuCount": 1,
      "diskSizeInGb": 40,
      "ramSizeInGb": 1.0
    },
    "state": "available",
    "backupRetentionEnabled": true,
    "pendingModifiedValues": {},
    "engine": "postgres",
    "engineVersion": "11.1",
    "latestRestorableTime": 1571686339.0,
    "masterUsername": "dbmasteruser",
    "parameterApplyStatus": "in-sync",
    "preferredBackupWindow": "06:19-06:49",
    "preferredMaintenanceWindow": "sun:10:19-sun:10:49",
    "publiclyAccessible": false,
    "masterEndpoint": {
      "port": 5432,
```

```
        "address":
          "ls-3EXAMPLEb4ffffb5cec056220c734713eEXAMPLEd.czowadgeezqi.us-
west-2.rds.amazonaws.com"
        },
        "pendingMaintenanceActions": []
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRelationalDatabases](#)」の「」を参照してください。

get-static-ip

次の例は、get-static-ip を使用する方法を説明しています。

AWS CLI

静的 IP に関する情報を取得するには

次のget-static-ip例では、指定された静的 IP の詳細を表示します。

```
aws lightsail get-static-ip \
  --static-ip-name StaticIp-1
```

出力:

```
{
  "staticIp": {
    "name": "StaticIp-1",
    "arn": "arn:aws:lightsail:us-
west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-82e2-2EXAMPLE23ad",
    "supportCode": "6EXAMPLE3362/192.0.2.0",
    "createdAt": 1571071325.076,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "resourceType": "StaticIp",
    "ipAddress": "192.0.2.0",
    "isAttached": false
  }
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStaticIp](#)」の「」を参照してください。

get-static-ips

次の例は、get-static-ips を使用する方法を説明しています。

AWS CLI

すべての静的情報を取得するには IPs

次のget-static-ips例では、IPs設定された AWS リージョンのすべての静的に関する詳細を表示します。

```
aws lightsail get-static-ips
```

出力:

```
{
  "staticIps": [
    {
      "name": "StaticIp-1",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:StaticIp/2257cd76-1f0e-4ac0-8EXAMPLE16f9423ad",
      "supportCode": "6EXAMPLE3362/192.0.2.0",
      "createdAt": 1571071325.076,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      },
      "resourceType": "StaticIp",
      "ipAddress": "192.0.2.0",
      "isAttached": false
    },
    {
      "name": "StaticIP-2",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:StaticIp/c61edb40-e5f0-4fd6-ae7c-8EXAMPLE19f8",
      "supportCode": "6EXAMPLE3362/192.0.2.2",
      "createdAt": 1568305385.681,
```

```
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "StaticIp",
        "ipAddress": "192.0.2.2",
        "attachedTo": "WordPress-1",
        "isAttached": true
    }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStaticIps](#)」の「」を参照してください。

is-vpc-peered

次のコード例は、`is-vpc-peered` を使用する方法を示しています。

AWS CLI

Amazon Lightsail 仮想プライベートクラウドがピアリングされているかどうかを確認するには

次の`is-vpc-peered`例では、指定された AWS リージョンの Amazon Lightsail 仮想プライベートクラウド (VPC) のピアリングステータスを返します。

```
aws lightsail is-vpc-peered \
  --region us-west-2
```

出力:

```
{
  "isPeered": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[IsVpcPeered](#)」の「」を参照してください。

open-instance-public-ports

次のコード例は、`open-instance-public-ports` を使用する方法を示しています。

AWS CLI

インスタンスのファイアウォールポートを開くには

次のopen-instance-public-ports例では、指定されたインスタンスでTCPポート 22 を開きます。

```
aws lightsail open-instance-public-ports \  
  --instance-name MEAN-2 \  
  --port-info fromPort=22,protocol=TCP,toPort=22
```

出力:

```
{  
  "operation": {  
    "id": "719744f0-a022-46f2-9f11-6EXAMPLE4642",  
    "resourceName": "MEAN-2",  
    "resourceType": "Instance",  
    "createdAt": 1571072906.849,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": true,  
    "operationDetails": "22/tcp",  
    "operationType": "OpenInstancePublicPorts",  
    "status": "Succeeded",  
    "statusChangedAt": 1571072906.849  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[OpenInstancePublicPorts](#)」の「」を参照してください。

peer-vpc

次のコード例は、peer-vpc を使用する方法を示しています。

AWS CLI

Amazon Lightsail 仮想プライベートクラウドをピアリングするには

次のpeer-vpc例では、指定した AWS リージョンの Amazon Lightsail 仮想プライベートクラウド (VPC) をピアリングします。

```
aws lightsail peer-vpc \  
  --region us-west-2
```

出力:

```
{  
  "operation": {  
    "id": "787e846a-54ac-497f-bce2-9EXAMPLE5d91",  
    "resourceName": "vpc-0EXAMPLEEa5261efb3",  
    "resourceType": "PeeredVpc",  
    "createdAt": 1571694233.104,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": true,  
    "operationDetails": "vpc-e2b3eb9b",  
    "operationType": "PeeredVpc",  
    "status": "Succeeded",  
    "statusChangedAt": 1571694233.104  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PeerVpc](#)」の「」を参照してください。

reboot-instance

次の例は、reboot-instance を使用する方法を説明しています。

AWS CLI

インスタンスを再起動するには

次のreboot-instance例では、指定されたインスタンスを再起動します。

```
aws lightsail reboot-instance \  
  --instance-name MEAN-1
```

出力:

```
{
  "operations": [
    {
      "id": "2b679f1c-8b71-4bb4-8e97-8EXAMPLEed93",
      "resourceName": "MEAN-1",
      "resourceType": "Instance",
      "createdAt": 1571694445.49,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": true,
      "operationDetails": "",
      "operationType": "RebootInstance",
      "status": "Succeeded",
      "statusChangedAt": 1571694445.49
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootInstance](#)」の「」を参照してください。

reboot-relational-database

次の例は、reboot-relational-database を使用する方法を説明しています。

AWS CLI

リレーショナルデータベースを再起動するには

次のreboot-relational-database例では、指定されたリレーショナルデータベースを再起動します。

```
aws lightsail reboot-relational-database \
  --relational-database-name Database-1
```

出力:

```
{
```

```

"operations": [
  {
    "id": "e4c980c0-3137-496c-9c91-1EXAMPLEdec2",
    "resourceName": "Database-1",
    "resourceType": "RelationalDatabase",
    "createdAt": 1571694532.91,
    "location": {
      "availabilityZone": "us-west-2a",
      "regionName": "us-west-2"
    },
    "isTerminal": false,
    "operationDetails": "",
    "operationType": "RebootRelationalDatabase",
    "status": "Started",
    "statusChangedAt": 1571694532.91
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootRelationalDatabase](#)」の「」を参照してください。

release-static-ip

次の例は、`release-static-ip` を使用する方法を説明しています。

AWS CLI

静的 IP を削除するには

次の`release-static-ip`例では、指定された静的 IP を削除します。

```

aws lightsail release-static-ip \
  --static-ip-name StaticIp-1

```

出力:

```

{
  "operations": [
    {
      "id": "e374c002-dc6d-4c7f-919f-2EXAMPLE13ce",
      "resourceName": "StaticIp-1",

```



```
    "resourceType": "StaticIp",
    "createdAt": 1571694962.003,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "ReleaseStaticIp",
    "status": "Succeeded",
    "statusChangedAt": 1571694962.003
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReleaseStaticIp](#)」の「」を参照してください。

start-instance

次のコード例は、start-instance を使用する方法を示しています。

AWS CLI

インスタンスを開始するには

次のstart-instance例では、指定されたインスタンスを起動します。

```
aws lightsail start-instance \  
  --instance-name WordPress-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "f88d2a93-7cea-4165-afce-2d688cb18f23",  
      "resourceName": "WordPress-1",  
      "resourceType": "Instance",  
      "createdAt": 1571695583.463,  
      "location": {  
        "availabilityZone": "us-west-2a",
```

```
        "regionName": "us-west-2"
      },
      "isTerminal": false,
      "operationType": "StartInstance",
      "status": "Started",
      "statusChangedAt": 1571695583.463
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartInstance](#)」の「」を参照してください。

start-relational-database

次のコード例は、start-relational-database を使用する方法を示しています。

AWS CLI

リレーショナルデータベースを開始するには

次のstart-relational-database例では、指定されたリレーショナルデータベースを開始します。

```
aws lightsail start-relational-database \
  --relational-database-name Database-1
```

出力:

```
{
  "operations": [
    {
      "id": "4d5294ec-a38a-4fda-9e37-aEXAMPLE0d24",
      "resourceName": "Database-1",
      "resourceType": "RelationalDatabase",
      "createdAt": 1571695998.822,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "isTerminal": false,
```

```
    "operationType": "StartRelationalDatabase",
    "status": "Started",
    "statusChangedAt": 1571695998.822
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartRelationalDatabase](#)」の「」を参照してください。

stop-instance

次の例は、stop-instance を使用する方法を説明しています。

AWS CLI

インスタンスを停止するには

次のstop-instance例では、指定されたインスタンスを停止します。

```
aws lightsail stop-instance \  
--instance-name WordPress-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "265357e2-2943-4d51-888a-1EXAMPLE7585",  
      "resourceName": "WordPress-1",  
      "resourceType": "Instance",  
      "createdAt": 1571695471.134,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "StopInstance",  
      "status": "Started",  
      "statusChangedAt": 1571695471.134  
    }  
  ]  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StopInstance](#)」の「」を参照してください。

stop-relational-database

次のコード例は、stop-relational-database を使用する方法を示しています。

AWS CLI

リレーショナルデータベースを停止するには

次のstop-relational-database例では、指定されたリレーショナルデータベースを停止します。

```
aws lightsail stop-relational-database \  
  --relational-database-name Database-1
```

出力:

```
{  
  "operations": [  
    {  
      "id": "cc559c19-4adb-41e4-b75b-5EXAMPLE4e61",  
      "resourceName": "Database-1",  
      "resourceType": "RelationalDatabase",  
      "createdAt": 1571695526.29,  
      "location": {  
        "availabilityZone": "us-west-2a",  
        "regionName": "us-west-2"  
      },  
      "isTerminal": false,  
      "operationType": "StopRelationalDatabase",  
      "status": "Started",  
      "statusChangedAt": 1571695526.29  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StopRelationalDatabase](#)」の「」を参照してください。

unpeer-vpc

次の例は、unpeer-vpc を使用する方法を説明しています。

AWS CLI

Amazon Lightsail 仮想プライベートクラウドのピアリングを解除するには

次のunpeer-vpc例では、指定された AWS リージョンの Amazon Lightsail 仮想プライベートクラウド (VPC) を解きます。

```
aws lightsail unpeer-vpc \  
  --region us-west-2
```

出力:

```
{  
  "operation": {  
    "id": "531aca64-7157-47ab-84c6-eEXAMPLEd898",  
    "resourceName": "vpc-0EXAMPLEEa5261efb3",  
    "resourceType": "PeeredVpc",  
    "createdAt": 1571694109.945,  
    "location": {  
      "availabilityZone": "all",  
      "regionName": "us-west-2"  
    },  
    "isTerminal": true,  
    "operationDetails": "vpc-e2b3eb9b",  
    "operationType": "UnpeeredVpc",  
    "status": "Succeeded",  
    "statusChangedAt": 1571694109.945  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UnpeerVpc](#)」の「」を参照してください。

を使用した Macie の例 AWS CLI

次のコード例は、Macie AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-buckets

次の例は、describe-buckets を使用する方法を説明しています。

AWS CLI

Amazon Macie がアカウントのモニタリングおよび分析を行う 1 つ以上の S3 バケットに関するデータをクエリするには

次のdescribe-buckets例では、名前が MY-S3 で始まり、現在の AWS リージョンにあるすべての S3 バケットのメタデータをクエリします。MY-S3

```
aws macie2 describe-buckets \  
  --criteria '{"bucketName":{"prefix":"my-S3"}}'
```

出力:

```
{  
  "buckets": [  
    {  
      "accountId": "123456789012",  
      "allowsUnencryptedObjectUploads": "FALSE",  
      "bucketArn": "arn:aws:s3:::MY-S3-DOC-EXAMPLE-BUCKET1",  
      "bucketCreatedAt": "2020-05-18T19:54:00+00:00",  
      "bucketName": "MY-S3-DOC-EXAMPLE-BUCKET1",  
      "classifiableObjectCount": 13,  
      "classifiableSizeInBytes": 1592088,  
      "jobDetails": {  
        "isDefinedInJob": "TRUE",
```

```
    "isMonitoredByJob": "TRUE",
    "lastJobId": "08c81dc4a2f3377fae45c9ddaexample",
    "lastJobRunTime": "2021-04-26T14:55:30.270000+00:00"
  },
  "lastAutomatedDiscoveryTime": "2022-12-10T19:11:25.364000+00:00",
  "lastUpdated": "2022-12-13T07:33:06.337000+00:00",
  "objectCount": 13,
  "objectCountByEncryptionType": {
    "customerManaged": 0,
    "kmsManaged": 2,
    "s3Managed": 7,
    "unencrypted": 4,
    "unknown": 0
  },
  "publicAccess": {
    "effectivePermission": "NOT_PUBLIC",
    "permissionConfiguration": {
      "accountLevelPermissions": {
        "blockPublicAccess": {
          "blockPublicAcls": true,
          "blockPublicPolicy": true,
          "ignorePublicAcls": true,
          "restrictPublicBuckets": true
        }
      },
      "bucketLevelPermissions": {
        "accessControlList": {
          "allowsPublicReadAccess": false,
          "allowsPublicWriteAccess": false
        },
        "blockPublicAccess": {
          "blockPublicAcls": true,
          "blockPublicPolicy": true,
          "ignorePublicAcls": true,
          "restrictPublicBuckets": true
        },
        "bucketPolicy": {
          "allowsPublicReadAccess": false,
          "allowsPublicWriteAccess": false
        }
      }
    }
  },
  "region": "us-west-2",
```

```
    "replicationDetails": {
      "replicated": false,
      "replicatedExternally": false,
      "replicationAccounts": []
    },
    "sensitivityScore": 78,
    "serverSideEncryption": {
      "kmsMasterKeyId": null,
      "type": "NONE"
    },
    "sharedAccess": "NOT_SHARED",
    "sizeInBytes": 4549746,
    "sizeInBytesCompressed": 0,
    "tags": [
      {
        "key": "Division",
        "value": "HR"
      },
      {
        "key": "Team",
        "value": "Recruiting"
      }
    ],
    "unclassifiableObjectCount": {
      "fileType": 0,
      "storageClass": 0,
      "total": 0
    },
    "unclassifiableObjectSizeInBytes": {
      "fileType": 0,
      "storageClass": 0,
      "total": 0
    },
    "versioning": true
  },
  {
    "accountId": "123456789012",
    "allowsUnencryptedObjectUploads": "TRUE",
    "bucketArn": "arn:aws:s3:::MY-S3-DOC-EXAMPLE-BUCKET2",
    "bucketCreatedAt": "2020-11-25T18:24:38+00:00",
    "bucketName": "MY-S3-DOC-EXAMPLE-BUCKET2",
    "classifiableObjectCount": 8,
    "classifiableSizeInBytes": 133810,
    "jobDetails": {
```



```
    "isDefinedInJob": "TRUE",
    "isMonitoredByJob": "FALSE",
    "lastJobId": "188d4f6044d621771ef7d65f2example",
    "lastJobRunTime": "2021-04-09T19:37:11.511000+00:00"
  },
  "lastAutomatedDiscoveryTime": "2022-12-12T19:11:25.364000+00:00",
  "lastUpdated": "2022-12-13T07:33:06.337000+00:00",
  "objectCount": 8,
  "objectCountByEncryptionType": {
    "customerManaged": 0,
    "kmsManaged": 0,
    "s3Managed": 8,
    "unencrypted": 0,
    "unknown": 0
  },
  "publicAccess": {
    "effectivePermission": "NOT_PUBLIC",
    "permissionConfiguration": {
      "accountLevelPermissions": {
        "blockPublicAccess": {
          "blockPublicAcls": true,
          "blockPublicPolicy": true,
          "ignorePublicAcls": true,
          "restrictPublicBuckets": true
        }
      },
      "bucketLevelPermissions": {
        "accessControlList": {
          "allowsPublicReadAccess": false,
          "allowsPublicWriteAccess": false
        },
        "blockPublicAccess": {
          "blockPublicAcls": true,
          "blockPublicPolicy": true,
          "ignorePublicAcls": true,
          "restrictPublicBuckets": true
        },
        "bucketPolicy": {
          "allowsPublicReadAccess": false,
          "allowsPublicWriteAccess": false
        }
      }
    }
  }
},
```

```
    "region": "us-west-2",
    "replicationDetails": {
      "replicated": false,
      "replicatedExternally": false,
      "replicationAccounts": []
    },
    "sensitivityScore": 95,
    "serverSideEncryption": {
      "kmsMasterKeyId": null,
      "type": "AES256"
    },
    "sharedAccess": "EXTERNAL",
    "sizeInBytes": 175978,
    "sizeInBytesCompressed": 0,
    "tags": [
      {
        "key": "Division",
        "value": "HR"
      },
      {
        "key": "Team",
        "value": "Recruiting"
      }
    ],
    "unclassifiableObjectCount": {
      "fileType": 3,
      "storageClass": 0,
      "total": 3
    },
    "unclassifiableObjectSizeInBytes": {
      "fileType": 2999826,
      "storageClass": 0,
      "total": 2999826
    },
    "versioning": true
  }
]
}
```

詳細については、[Amazon Macie ユーザーガイド」のS3 バケットインベントリのフィルタリング](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeBuckets](#)」の「」を参照してください。

を使用した Amazon Managed Grafana の例 AWS CLI

次のコード例は、Amazon Managed Grafana AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

list-workspaces

次の例は、list-workspaces を使用する方法を説明しています。

AWS CLI

ユーザー認証情報で指定されたリージョン内のアカウントのワークスペースを一覧表示するには

次のlist-workspaces例では、アカウントのリージョンの Grafana ワークスペースを一覧表示します。

```
aws grafana list-workspaces
```

出力:

```
{
  "workspaces": [
    {
      "authentication": {
        "providers": [
          "AWS_SSO"
        ]
      }
    }
  ]
}
```

```
    },
    "created": "2022-04-04T16:20:21.796000-07:00",
    "description": "to test tags",
    "endpoint": "g-949e7b44df.grafana-workspace.us-east-1.amazonaws.com",
    "grafanaVersion": "8.2",
    "id": "g-949e7b44df",
    "modified": "2022-04-04T16:20:21.796000-07:00",
    "name": "testtag2",
    "notificationDestinations": [
      "SNS"
    ],
    "status": "ACTIVE"
  },
  {
    "authentication": {
      "providers": [
        "AWS_SSO"
      ]
    },
    "created": "2022-04-20T10:22:15.115000-07:00",
    "description": "ww",
    "endpoint": "g-bffa51ed1b.grafana-workspace.us-east-1.amazonaws.com",
    "grafanaVersion": "8.2",
    "id": "g-bffa51ed1b",
    "modified": "2022-04-20T10:22:15.115000-07:00",
    "name": "ww",
    "notificationDestinations": [
      "SNS"
    ],
    "status": "ACTIVE"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListWorkspaces](#)」の「」を参照してください。

MediaConnect を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています MediaConnect。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-flow-outputs

次の例は、add-flow-outputs を使用する方法を説明しています。

AWS CLI

フローに出力を追加するには

次のadd-flow-outputs例では、指定されたフローに出力を追加します。

```
aws mediaconnect add-flow-outputs \  
--flow-arn arn:aws:mediaconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \  
--outputs Description='NYC  
stream',Destination=192.0.2.12,Name=NYC,Port=3333,Protocol=rtp-  
fec,SmoothingLatency=100 Description='LA  
stream',Destination=203.0.113.9,Name=LA,Port=4444,Protocol=rtp-  
fec,SmoothingLatency=100
```

出力:

```
{  
  "Outputs": [  
    {  
      "Port": 3333,  
      "OutputArn": "arn:aws:mediaconnect:us-  
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",  
      "Name": "NYC",  
      "Description": "NYC stream",  
      "Destination": "192.0.2.12",
```

```
    "Transport": {
      "Protocol": "rtp-fec",
      "SmoothingLatency": 100
    }
  },
  {
    "Port": 4444,
    "OutputArn": "arn:aws:mediacconnect:us-
east-1:111122223333:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
    "Name": "LA",
    "Description": "LA stream",
    "Destination": "203.0.113.9",
    "Transport": {
      "Protocol": "rtp-fec",
      "SmoothingLatency": 100
    }
  }
],
"FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[フローへの出力の追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddFlowOutputs](#)」の「」を参照してください。

create-flow

次のコード例は、create-flow を使用する方法を示しています。

AWS CLI

フローを作成するには

次のcreate-flow例では、指定された設定でフローを作成します。

```
aws mediacconnect create-flow \
  --availability-zone us-west-2c \
  --name ExampleFlow \
  --source Description='Example source,
backup',IngestPort=1055,Name=BackupSource,Protocol=rtp,WhitelistCidr=10.24.34.0/23
```

出力:

```
{
  "Flow": {
    "FlowArn": "arn:aws:mediacconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:ExampleFlow",
    "AvailabilityZone": "us-west-2c",
    "EgressIp": "54.245.71.21",
    "Source": {
      "IngestPort": 1055,
      "SourceArn": "arn:aws:mediacconnect:us-
east-1:123456789012:source:2-3aBC45dEF67hiJ89-c34de5fG678h:BackupSource",
      "Transport": {
        "Protocol": "rtp",
        "MaxBitrate": 80000000
      },
      "Description": "Example source, backup",
      "IngestIp": "54.245.71.21",
      "WhitelistCidr": "10.24.34.0/23",
      "Name": "mySource"
    },
    "Entitlements": [],
    "Name": "ExampleFlow",
    "Outputs": [],
    "Status": "STANDBY",
    "Description": "Example source, backup"
  }
}
```

詳細については、「AWS Elemental MediaConnect ユーザーガイド」の「[フローの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFlow](#)」の「」を参照してください。

delete-flow

次のコード例は、delete-flow を使用する方法を示しています。

AWS CLI

フローを削除するには

次のdelete-flow例では、指定されたフローを削除します。

```
aws mediaconnect delete-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

出力:

```
{  
  "FlowArn": "arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",  
  "Status": "DELETING"  
}
```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[フローの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFlow](#)」の「」を参照してください。

describe-flow

次の例は、describe-flow を使用する方法を説明しています。

AWS CLI

フローの詳細を表示するには

次のdescribe-flow例では、、アベイラビリティゾーン、ステータス、ソースARN、エンタイトルメント、出力など、指定されたフローの詳細を表示します。

```
aws mediaconnect describe-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

出力:

```
{  
  "Flow": {  
    "EgressIp": "54.201.4.39",  
    "AvailabilityZone": "us-west-2c",  
    "Status": "ACTIVE",
```



```
"FlowArn": "arn:aws:mediacconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
  "Entitlements": [
    {
      "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:123456789012:entitlement:1-AaBb11CcDd22EeFf-34DE5fG12AbC:MyEntitlement",
      "Description": "Assign to this account",
      "Name": "MyEntitlement",
      "Subscribers": [
        "444455556666"
      ]
    }
  ],
  "Description": "NYC awards show",
  "Name": "AwardsShow",
  "Outputs": [
    {
      "Port": 2355,
      "Name": "NYC",
      "Transport": {
        "SmoothingLatency": 0,
        "Protocol": "rtp-fec"
      },
      "OutputArn": "arn:aws:mediacconnect:us-
east-1:123456789012:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
      "Destination": "192.0.2.0"
    },
    {
      "Port": 3025,
      "Name": "LA",
      "Transport": {
        "SmoothingLatency": 0,
        "Protocol": "rtp-fec"
      },
      "OutputArn": "arn:aws:mediacconnect:us-
east-1:123456789012:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
      "Destination": "192.0.2.0"
    }
  ],
  "Source": {
    "IngestIp": "54.201.4.39",
    "SourceArn": "arn:aws:mediacconnect:us-
east-1:123456789012:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource",
    "Transport": {
```

```

        "MaxBitrate": 80000000,
        "Protocol": "rtp"
    },
    "IngestPort": 1069,
    "Description": "Saturday night show",
    "Name": "ShowSource",
    "WhitelistCidr": "10.24.34.0/23"
}
}
}

```

詳細については、「AWS Elemental MediaConnect ユーザーガイド」の「[フローの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFlow](#)」の「」を参照してください。

grant-flow-entitlements

次の例は、grant-flow-entitlements を使用する方法を説明しています。

AWS CLI

フローにエンタイトルメントを付与するには

次のgrant-flow-entitlements例では、指定された既存のフローに、コンテンツを別の AWS アカウントと共有するための権限を付与します。

```

aws mediaconnect grant-flow-entitlements \
  --flow-arn arn:aws:mediaconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --entitlements Description='For AnyCompany',Encryption={"Algorithm=aes128,KeyType=static-key,RoleArn=arn:aws:iam::111122223333:role/MediaConnect-ASM,SecretArn=arn:aws:secretsmanager:us-west-2:111122223333:secret:mySecret1"},Name=AnyCompany_Entitlement,Subscribers=444455556666 Description='For Example Corp',Name=ExampleCorp,Subscribers=777788889999

```

出力:

```

{
  "Entitlements": [

```

```
{
  "Name": "AnyCompany_Entitlement",
  "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
  "Subscribers": [
    "444455556666"
  ],
  "Description": "For AnyCompany",
  "Encryption": {
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1",
    "Algorithm": "aes128",
    "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
    "KeyType": "static-key"
  }
},
{
  "Name": "ExampleCorp",
  "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:111122223333:entitlement:1-3333cccc4444dddd-1111aaaa2222:ExampleCorp",
  "Subscribers": [
    "777788889999"
  ],
  "Description": "For Example Corp"
}
],
"FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[フローに対する権限の付与](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGrantFlowEntitlements](#)」の「」を参照してください。

list-entitlements

次のコード例は、list-entitlements を使用する方法を示しています。

AWS CLI

エンタイトルメントのリストを表示するには

次のlist-entitlements例では、アカウントに付与されたすべてのエンタイトルメントのリストを表示します。

```
aws mediaconnect list-entitlements
```

出力:

```
{
  "Entitlements": [
    {
      "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
      "EntitlementName": "MyEntitlement"
    }
  ]
}
```

詳細については、「AWS Elemental MediaConnect API Reference [ListEntitlements](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEntitlements](#)」の「」を参照してください。

list-flows

次の例は、list-flows を使用する方法を説明しています。

AWS CLI

フローのリストを表示するには

次のlist-flows例では、フローのリストを表示します。

```
aws mediaconnect list-flows
```

出力:

```
{
  "Flows": [
    {
      "Status": "STANDBY",
      "SourceType": "OWNED",
```

```

    "AvailabilityZone": "us-west-2a",
    "Description": "NYC awards show",
    "Name": "AwardsShow",
    "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
  },
  {
    "Status": "STANDBY",
    "SourceType": "OWNED",
    "AvailabilityZone": "us-west-2c",
    "Description": "LA basketball game",
    "Name": "BasketballGame",
    "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
  }
]
}

```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[フローのリストの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFlows](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

MediaConnect リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定された MediaConnect リソースに関連付けられたタグキーと値を表示します。

```

aws mediacconnect list-tags-for-resource \
  --resource-arn arn:aws:mediacconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame

```

出力:

```
{
```

```
"Tags": {
  "region": "west",
  "stage": "prod"
}
```

詳細については、「AWS Elemental MediaConnect API Reference」の[ListTagsForResource TagResource UntagResource](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

remove-flow-output

次のコード例は、remove-flow-output を使用する方法を示しています。

AWS CLI

フローから出力を削除するには

次のremove-flow-output例では、指定されたフローから出力を削除します。

```
aws mediaconnect remove-flow-output \
  --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --output-arn arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC
```

出力:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC"
}
```

詳細については、「AWS Elemental MediaConnect ユーザーガイド」の[「フローからの出力の削除](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[RemoveFlowOutput](#)」の「」を参照してください。

revoke-flow-entitlement

次の例は、revoke-flow-entitlement を使用する方法を説明しています。

AWS CLI

エンタイトルメントを取り消すには

次のrevoke-flow-entitlement例では、指定されたフローのエンタイトルメントを取り消します。

```
aws mediaconnect revoke-flow-entitlement \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \  
  --entitlement-arn arn:aws:mediaconnect:us-  
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
```

出力:

```
{  
  "FlowArn": "arn:aws:mediaconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",  
  "EntitlementArn": "arn:aws:mediaconnect:us-  
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement"  
}
```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[エンタイトルメントの取り消し](#)」を参照してください。

- API 詳細については、「[コマンドリファレンスRevokeFlowEntitlement](#)」の「」を参照してください。AWS CLI

start-flow

次の例は、start-flow を使用する方法を説明しています。

AWS CLI

フローを開始するには

次のstart-flow例では、指定されたフローを開始します。

```
aws mediaconnect start-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame
```

```
--flow-arn arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

このコマンドでは何も出力されません。出力:

```
{  
  "FlowArn": "arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",  
  "Status": "STARTING"  
}
```

詳細については、「AWS Elemental MediaConnect ユーザーガイド」の「[フローの開始](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartFlow](#)」の「」を参照してください。

stop-flow

次の例は、stop-flow を使用する方法を説明しています。

AWS CLI

フローを停止するには

次のstop-flow例では、指定されたフローを停止します。

```
aws mediacconnect stop-flow \  
  --flow-arn arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

出力:

```
{  
  "Status": "STOPPING",  
  "FlowArn": "arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"  
}
```

詳細については、「AWS Elemental MediaConnect ユーザーガイド」の「[フローの停止](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopFlow](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

MediaConnect リソースにタグを追加するには

次のtag-resource例では、指定された MediaConnect リソースにキー名と値を持つタグを追加します。

```
aws mediacconnect tag-resource \  
  --resource-arn arn:aws:mediacconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame  
  --tags region=west
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaConnect API Reference」の[ListTagsForResource TagResource UntagResource](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

MediaConnect リソースからタグを削除するには

次のuntag-resource例では、指定されたキー名と関連付けられた値のタグを MediaConnect リソースから削除します。

```
aws mediacconnect untag-resource \  
  --resource-arn arn:aws:mediacconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \  
  --tags region=west
```

```
--tag-keys region
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaConnect API Reference」の[ListTagsForResource TagResource UntagResource](#)「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-flow-entitlement

次の例は、update-flow-entitlement を使用する方法を説明しています。

AWS CLI

エンタイトルメントを更新するには

次のupdate-flow-entitlement例では、指定されたエンタイトルメントを新しい説明とサブスクライバーで更新します。

```
aws mediaconnect update-flow-entitlement \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \  
  --entitlement-arn arn:aws:mediaconnect:us-  
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement \  
  --description 'For AnyCompany Affiliate' \  
  --subscribers 777788889999
```

出力:

```
{  
  "FlowArn": "arn:aws:mediaconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",  
  "Entitlement": {  
    "Name": "AnyCompany_Entitlement",  
    "Description": "For AnyCompany Affiliate",  
    "EntitlementArn": "arn:aws:mediaconnect:us-  
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",  
    "Encryption": {  
      "KeyType": "static-key",  
      "Algorithm": "aes128",
```

```

        "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
    },
    "Subscribers": [
        "777788889999"
    ]
}
}

```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[エンタイトルメントの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateFlowEntitlement](#)」の「」を参照してください。

update-flow-output

次の例は、update-flow-output を使用する方法を説明しています。

AWS CLI

フローの出力を更新するには

次のupdate-flow-output例では、指定されたフローの出力を更新します。

```

aws mediacconnect update-flow-output \
  --flow-arn arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --output-arn arn:aws:mediacconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC \
  --port 3331

```

出力:

```

{
  "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "Output": {
    "Name": "NYC",
    "Port": 3331,
    "Description": "NYC stream",
    "Transport": {

```

```

        "Protocol": "rtp-fec",
        "SmoothingLatency": 100
    },
    "OutputArn": "arn:aws:mediacconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
    "Destination": "192.0.2.12"
}
}

```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[フローの出力の更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFlowOutput](#)」の「」を参照してください。

update-flow-source

次のコード例は、update-flow-source を使用する方法を示しています。

AWS CLI

既存のフローのソースを更新するには

次のupdate-flow-source例では、既存のフローのソースを更新します。

```

aws mediacconnect update-flow-source \
  --flow-arn arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow \
  --source-arn arn:aws:mediacconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource \
  --description 'Friday night show' \
  --ingest-port 3344 \
  --protocol rtp-fec \
  --whitelist-cidr 10.24.34.0/23

```

出力:

```

{
  "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
  "Source": {
    "IngestIp": "34.210.136.56",
    "WhitelistCidr": "10.24.34.0/23",

```

```
    "Transport": {
      "Protocol": "rtp-fec"
    },
    "IngestPort": 3344,
    "Name": "ShowSource",
    "Description": "Friday night show",
    "SourceArn": "arn:aws:mediaconnect:us-east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource"
  }
}
```

詳細については、「[AWS Elemental MediaConnect ユーザーガイド](#)」の「[フローのソースの更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFlowSource](#)」の「」を参照してください。

MediaConvert を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています MediaConvert。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

cancel-job

次の例は、cancel-job を使用する方法を説明しています。

AWS CLI

キューにあるジョブをキャンセルするには

次のcancel-job例では、ID が のジョブをキャンセルします1234567891234-abc123。サービスが処理を開始したジョブをキャンセルすることはできません。

```
aws mediaconvert cancel-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --id 1234567891234-abc123
```

アカウント固有のエンドポイントを取得するには、describe-endpoints を使用するか、エンドポイントを指定せずにコマンドを送信します。このサービスは、エラーとエンドポイントを返します。

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Jobs の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelJob](#)」の「」を参照してください。

create-job-template

次のコード例は、create-job-template を使用方法を示しています。

AWS CLI

ジョブテンプレートを作成するには

次のcreate-job-template例では、システムjob-template.jsonに存在する ファイルで指定されているトランスコード設定を使用してジョブテンプレートを作成します。

```
aws mediaconvert create-job-template \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --name JobTemplate1 \  
  --cli-input-json file://~/job-template.json
```

を使用してジョブテンプレートJSONファイルを作成し、ファイルget-job-templateを変更した場合は、JobTemplateオブジェクトを削除しますが、設定の子オブジェクトはその中に保持します。また、LastUpdated、Arn、Typeおよび のキーと値のペアも必ず削除してくださいCreatedAt。ファイルJSONまたはコマンドラインで、カテゴリ、説明、名前、キューを指定できます。

アカウント固有のエンドポイントを取得するには、`describe-endpoints` を使用するか、エンドポイントを指定せずにコマンドを送信します。このサービスは、エラーとエンドポイントを返します。

リクエストが成功すると、サービスは作成したジョブテンプレートのJSON仕様を返します。

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Job Templates の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateJobTemplate](#)」の「」を参照してください。

create-job

次のコード例は、`create-job` を使用する方法を示しています。

AWS CLI

ジョブを作成するには

次の `create-job` の例では、コマンドの送信元となるシステム上のファイル `job.json` に指定されている設定を使用してトランスコーディングジョブを作成します。このJSONジョブ仕様では、各設定を個別に指定したり、ジョブテンプレートを参照したり、出力プリセットを参照したりできます。

```
aws mediaconvert create-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --cli-input-json file://~/job.json
```

AWS Elemental MediaConvert コンソールを使用して、JSONジョブ設定を選択し、ジョブセクションの下部でジョブを表示JSONを選択して、ジョブ仕様を生成できます。

アカウント固有のエンドポイントを取得するには、`describe-endpoints` を使用するか、エンドポイントを指定せずにコマンドを送信します。このサービスは、エラーとエンドポイントを返します。

リクエストが成功すると、サービスはリクエストで送信したJSONジョブ仕様を返します。

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Jobs の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateJob](#)」の「」を参照してください。

create-preset

次の例は、create-preset を使用する方法を説明しています。

AWS CLI

カスタム出カプリセットを作成するには

次のcreate-preset例では、ファイルで指定された出力設定に基づいてカスタム出カプリセットを作成しますpresets.json。カテゴリ、説明、および名前は、JSON ファイルまたはコマンドラインで指定できます。

```
aws mediaconvert create-preset \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --cli-input-json file://~/presets.json
```

get-preset を使用してプリセットJSONファイルを作成し、出力ファイルを変更する場合は、LastUpdated、Arn、Typeおよびのキーと値のペアを削除してくださいCreatedAt。

アカウント固有のエンドポイントを取得するには、describe-endpoints を使用するか、エンドポイントを指定せずにコマンドを送信します。このサービスは、エラーとエンドポイントを返します。

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Output Presets の使用](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePreset](#)」の「」を参照してください。

create-queue

次のコード例は、create-queue を使用する方法を示しています。

AWS CLI

カスタムキューを作成するには

次のcreate-queue例では、カスタムトランスコーディングキューを作成します。


```
aws mediaconvert create-queue \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --name Queue1 \  
  --description "Keep this queue empty unless job is urgent."
```

アカウント固有のエンドポイントを取得するには、describe-endpoints を使用するか、エンドポイントを指定せずにコマンドを送信します。このサービスは、エラーとエンドポイントを返します。

出力:

```
{  
  "Queue": {  
    "Status": "ACTIVE",  
    "Name": "Queue1",  
    "LastUpdated": 1518034928,  
    "Arn": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",  
    "Type": "CUSTOM",  
    "CreatedAt": 1518034928,  
    "Description": "Keep this queue empty unless job is urgent."  
  }  
}
```

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Queues の使用](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateQueue](#)」の「」を参照してください。

delete-job-template

次の例は、delete-job-template を使用する方法を説明しています。

AWS CLI

ジョブテンプレートを削除するには

次のdelete-job-template例では、指定されたカスタムジョブテンプレートを削除します。

```
aws mediaconvert delete-job-template \  
  --name "DASH Streaming" \  
  --region region-name-1
```

```
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

このコマンドでは何も出力されません。を実行してaws mediaconvert list-job-templates、テンプレートが削除されたことを確認します。

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Job Templates の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteJobTemplate](#)」の「」を参照してください。

delete-preset

次の例は、delete-preset を使用する方法を説明しています。

AWS CLI

カスタムオンデマンドキューを削除するには

次のdelete-preset例では、指定されたカスタムプリセットを削除します。

```
aws mediaconvert delete-preset \  
  --name SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

このコマンドでは何も出力されません。を実行してaws mediaconvert list-presets、プリセットが削除されたことを確認します。

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Output Presets の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePreset](#)」の「」を参照してください。

delete-queue

次のコード例は、delete-queue を使用する方法を示しています。

AWS CLI

カスタムオンデマンドキューを削除するには

次のdelete-queue例では、指定されたカスタムオンデマンドキューを削除します。

デフォルトのキューを削除することはできません。有効な料金プランがある、または未処理のジョブを含む予約済みのキューを削除することはできません。

```
aws mediaconvert delete-queue \  
  --name Customer1 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

このコマンドでは何も出力されません。を実行してaws mediaconvert list-queues、キューが削除されたことを確認します。

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Queues の使用](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteQueue](#)」の「」を参照してください。

describe-endpoints

次のコード例は、describe-endpoints を使用する方法を示しています。

AWS CLI

アカウント固有のエンドポイントを取得するには

次のdescribe-endpoints例では、他のリクエストを サービスに送信する必要があるエンドポイントを取得します。

```
aws mediaconvert describe-endpoints
```

出力:

```
{  
  "Endpoints": [  
    {  
      "Url": "https://abcd1234.mediaconvert.region-name-1.amazonaws.com"  
    }  
  ]  
}
```

詳細については、「[AWS Elemental MediaConvert API Reference](#)」の [MediaConvert](#) 「[の使用の開始API方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEndpoints](#)」の「」を参照してください。

get-job-template

次の例は、get-job-template を使用する方法を説明しています。

AWS CLI

ジョブテンプレートの詳細を取得するには

次のget-job-template例では、指定されたカスタムジョブテンプレートJSONの定義を表示します。

```
aws mediaconvert get-job-template \  
  --name "DASH Streaming" \  
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com
```

出力:

```
{  
  "JobTemplate": {  
    "StatusUpdateInterval": "SECONDS_60",  
    "LastUpdated": 1568652998,  
    "Description": "Create a DASH streaming ABR stack",  
    "CreatedAt": 1568652998,  
    "Priority": 0,  
    "Name": "DASH Streaming",  
    "Settings": {  
      ...<truncatedforbrevity>...  
    },  
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH  
Streaming",  
    "Type": "CUSTOM"  
  }  
}
```

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Job Templates の使用](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[GetJobTemplate](#)」の「」を参照してください。

get-job

次のコード例は、get-job を使用する方法を示しています。

AWS CLI

特定のジョブの詳細を取得するには

次の例は、ID 1234567890987-1ab2c3 のジョブの情報をリクエストしていますが、エラーで終了します。

```
aws mediaconvert get-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --id 1234567890987-1ab2c3
```

アカウント固有のエンドポイントを取得するには、describe-endpoints を使用するか、エンドポイントを指定せずにコマンドを送信します。このサービスは、エラーとエンドポイントを返します。

リクエストが成功すると、サービスは次のように、ジョブ設定、返されたエラー、その他のジョブデータなどのジョブ情報を含むJSONファイルを返します。

```
{  
  "Job": {  
    "Status": "ERROR",  
    "Queue": "arn:aws:mediaconvert:region-name-1:012345678998:queues/Queue1",  
    "Settings": {  
      ...<truncated for brevity>...  
    },  
    "ErrorMessage": "Unable to open input file [s3://my-input-bucket/file-name.mp4]: [Failed probe/open: [Failed to read data: AssumeRole failed]]",  
    "ErrorCode": 1434,  
    "Role": "arn:aws:iam::012345678998:role/MediaConvertServiceRole",  
    "Arn": "arn:aws:mediaconvert:us-west-1:012345678998:jobs/1234567890987-1ab2c3",  
    "UserMetadata": {},  
    "Timing": {  
      "FinishTime": 1517442131,  
      "SubmitTime": 1517442103,  
      "StartTime": 1517442104  
    },  
    "Id": "1234567890987-1ab2c3",
```

```
    "CreatedAt": 1517442103
  }
}
```

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Jobs の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[GetJob](#)」の「」を参照してください。

get-preset

次のコード例は、get-preset を使用する方法を示しています。

AWS CLI

特定のプリセットの詳細を取得するには

次のget-preset例では、指定されたカスタムプリセットJSONの定義をリクエストします。

```
aws mediaconvert get-preset \  
  --name SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{  
  "Preset": {  
    "Description": "Creates basic MP4 file. No filtering or preprocessing.",  
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4",  
    "LastUpdated": 1568843141,  
    "Name": "SimpleMP4",  
    "Settings": {  
      "ContainerSettings": {  
        "Mp4Settings": {  
          "FreeSpaceBox": "EXCLUDE",  
          "CslgAtom": "INCLUDE",  
          "MoovPlacement": "PROGRESSIVE_DOWNLOAD"  
        },  
        "Container": "MP4"  
      },  
      "AudioDescriptions": [  
        {
```

```
    "LanguageCodeControl": "FOLLOW_INPUT",
    "AudioTypeControl": "FOLLOW_INPUT",
    "CodecSettings": {
      "AacSettings": {
        "RawFormat": "NONE",
        "CodecProfile": "LC",
        "AudioDescriptionBroadcasterMix": "NORMAL",
        "SampleRate": 48000,
        "Bitrate": 96000,
        "RateControlMode": "CBR",
        "Specification": "MPEG4",
        "CodingMode": "CODING_MODE_2_0"
      },
      "Codec": "AAC"
    }
  ],
  "VideoDescription": {
    "RespondToAfd": "NONE",
    "TimecodeInsertion": "DISABLED",
    "Sharpness": 50,
    "ColorMetadata": "INSERT",
    "CodecSettings": {
      "H264Settings": {
        "FramerateControl": "INITIALIZE_FROM_SOURCE",
        "SpatialAdaptiveQuantization": "ENABLED",
        "Softness": 0,
        "Telecine": "NONE",
        "CodecLevel": "AUTO",
        "QualityTuningLevel": "SINGLE_PASS",
        "UnregisteredSeiTimecode": "DISABLED",
        "Slices": 1,
        "Syntax": "DEFAULT",
        "GopClosedCadence": 1,
        "AdaptiveQuantization": "HIGH",
        "EntropyEncoding": "CABAC",
        "InterlaceMode": "PROGRESSIVE",
        "ParControl": "INITIALIZE_FROM_SOURCE",
        "NumberBFramesBetweenReferenceFrames": 2,
        "GopSizeUnits": "FRAMES",
        "RepeatPps": "DISABLED",
        "CodecProfile": "MAIN",
        "FieldEncoding": "PAFF",
        "GopSize": 90.0,
```

```

        "SlowPal": "DISABLED",
        "SceneChangeDetect": "ENABLED",
        "GopBReference": "DISABLED",
        "RateControlMode": "CBR",
        "FramerateConversionAlgorithm": "DUPLICATE_DROP",
        "FlickerAdaptiveQuantization": "DISABLED",
        "DynamicSubGop": "STATIC",
        "MinIInterval": 0,
        "TemporalAdaptiveQuantization": "ENABLED",
        "Bitrate": 400000,
        "NumberReferenceFrames": 3
    },
    "Codec": "H_264"
},
"AfdSignaling": "NONE",
"AntiAlias": "ENABLED",
"ScalingBehavior": "DEFAULT",
"DropFrameTimecode": "ENABLED"
}
},
"Type": "CUSTOM",
"CreatedAt": 1568841521
}
}

```

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Output Presets の使用](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPreset](#)」の「」を参照してください。

get-queue

次の例は、get-queue を使用する方法を説明しています。

AWS CLI

キューの詳細を取得するには

次のget-queue例では、指定されたカスタムキューの詳細を取得します。

```
aws mediaconvert get-queue \
  --name Customer1 \
```



```
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{
  "Queue": {
    "LastUpdated": 1526428502,
    "Type": "CUSTOM",
    "SubmittedJobsCount": 0,
    "Status": "ACTIVE",
    "PricingPlan": "ON_DEMAND",
    "CreatedAt": 1526428502,
    "ProgressingJobsCount": 0,
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
    "Name": "Customer1"
  }
}
```

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Queues の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[GetQueue](#)」の「」を参照してください。

list-job-templates

次のコード例は、list-job-templates を使用する方法を示しています。

AWS CLI

例 1: カスタムジョブテンプレートを一覧表示するには

次のlist-job-templates例では、現在のリージョンのすべてのカスタムジョブテンプレートを一覧表示します。システムジョブテンプレートを一覧表示するには、次の例を参照してください。

```
aws mediaconvert list-job-templates \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{
```

```
"JobTemplates": [  
  {  
    "Description": "Create a DASH streaming ABR stack",  
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH  
Streaming",  
    "Name": "DASH Streaming",  
    "LastUpdated": 1568653007,  
    "Priority": 0,  
    "Settings": {  
      ...<truncatedforbrevity>...  
    },  
    "Type": "CUSTOM",  
    "StatusUpdateInterval": "SECONDS_60",  
    "CreatedAt": 1568653007  
  },  
  {  
    "Description": "Create a high-res file",  
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/File",  
    "Name": "File",  
    "LastUpdated": 1568653007,  
    "Priority": 0,  
    "Settings": {  
      ...<truncatedforbrevity>...  
    },  
    "Type": "CUSTOM",  
    "StatusUpdateInterval": "SECONDS_60",  
    "CreatedAt": 1568653023  
  }  
]  
}
```

例 2: MediaConvert システムジョブテンプレートを一覧表示するには

次のlist-job-templates例では、すべてのシステムジョブテンプレートを一覧表示します。

```
aws mediaconvert list-job-templates \  
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com \  
  --list-by SYSTEM
```

出力:

```
{  
  "JobTemplates": [  
    {  
      "Description": "Create a DASH streaming ABR stack",  
      "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/DASH  
Streaming",  
      "Name": "DASH Streaming",  
      "LastUpdated": 1568653007,  
      "Priority": 0,  
      "Settings": {  
        ...<truncatedforbrevity>...  
      },  
      "Type": "CUSTOM",  
      "StatusUpdateInterval": "SECONDS_60",  
      "CreatedAt": 1568653007  
    },  
    {  
      "Description": "Create a high-res file",  
      "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:jobTemplates/File",  
      "Name": "File",  
      "LastUpdated": 1568653007,  
      "Priority": 0,  
      "Settings": {  
        ...<truncatedforbrevity>...  
      },  
      "Type": "CUSTOM",  
      "StatusUpdateInterval": "SECONDS_60",  
      "CreatedAt": 1568653023  
    }  
  ]  
}
```

```
{
  "CreatedAt": 1568321779,
  "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:jobTemplates/System-
Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
  "Name": "System-Generic_Mp4_Hev1_Avc_Aac_Sdr_Qvbr",
  "Description": "GENERIC, MP4, AVC + HEV1(HEVC,SDR), AAC, SDR, QVBR",
  "Category": "GENERIC",
  "Settings": {
    "AdAvailOffset": 0,
    "OutputGroups": [
      {
        "Outputs": [
          {
            "Extension": "mp4",
            "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5Mbps_Qvbr_Vq9",
            "NameModifier":
"_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1280x720p_30Hz_5000Kbps_Qvbr_Vq9"
          },
          {
            "Extension": "mp4",
            "Preset": "System-
Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10Mbps_Qvbr_Vq9",
            "NameModifier":
"_Generic_Hd_Mp4_Avc_Aac_16x9_Sdr_1920x1080p_30Hz_10000Kbps_Qvbr_Vq9"
          },
          {
            "Extension": "mp4",
            "Preset": "System-
Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_0.8Mbps_Qvbr_Vq7",
            "NameModifier":
"_Generic_Sd_Mp4_Avc_Aac_16x9_Sdr_640x360p_30Hz_800Kbps_Qvbr_Vq7"
          },
          {
            "Extension": "mp4",
            "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4Mbps_Qvbr_Vq9",
            "NameModifier":
"_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1280x720p_30Hz_4000Kbps_Qvbr_Vq9"
          },
          {
            "Extension": "mp4",
            "Preset": "System-
Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8Mbps_Qvbr_Vq9",
```

```

        "NameModifier":
        "_Generic_Hd_Mp4_Hev1_Aac_16x9_Sdr_1920x1080p_30Hz_8000Kbps_Qvbr_Vq9"
        },
        {
            "Extension": "mp4",
            "Preset": "System-
Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12Mbps_Qvbr_Vq9",
            "NameModifier":
            "_Generic_Uhd_Mp4_Hev1_Aac_16x9_Sdr_3840x2160p_30Hz_12000Kbps_Qvbr_Vq9"
            }
        ],
        "OutputGroupSettings": {
            "FileGroupSettings": {

            },
            "Type": "FILE_GROUP_SETTINGS"
        },
        "Name": "File Group"
    }
    ]
},
"Type": "SYSTEM",
"LastUpdated": 1568321779
},
...<truncatedforbrevity>...
]
}

```

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Job Templates の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobTemplates](#)」の「」を参照してください。

list-jobs

次のコード例は、list-jobs を使用する方法を示しています。

AWS CLI

リージョン内のすべてのジョブの詳細を取得するには

次の例は、指定されたリージョンのすべてのジョブの情報をリクエストします。

```
aws mediaconvert list-jobs \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1
```

アカウント固有のエンドポイントを取得するには、`describe-endpoints` を使用するか、エンドポイントを指定せずにコマンドを送信します。このサービスは、エラーとエンドポイントを返します。

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Jobs の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobs](#)」の「」を参照してください。

list-presets

次のコード例は、`list-presets` を使用する方法を示しています。

AWS CLI

例 1: カスタム出力プリセットを一覧表示するには

次の`list-presets`例では、カスタム出力プリセットを一覧表示します。システムプリセットを一覧表示するには、次の例を参照してください。

```
aws mediaconvert list-presets \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{  
  "Presets": [  
    {  
      "Name": "SimpleMP4",  
      "CreatedAt": 1568841521,  
      "Settings": {  
        .....  
      },  
      "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",  
      "Type": "CUSTOM",  
      "LastUpdated": 1568843141,  
      "Description": "Creates basic MP4 file. No filtering or preprocessing."  
    }  
  ]  
}
```

```
    },
    {
      "Name": "SimpleTS",
      "CreatedAt": 1568843113,
      "Settings": {
        ... truncated for brevity ...
      },
      "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleTS",
      "Type": "CUSTOM",
      "LastUpdated": 1568843113,
      "Description": "Create a basic transport stream."
    }
  ]
}
```

例 2: システム出力プリセットを一覧表示するには

次のlist-presets例では、使用可能な MediaConvert システムプリセットを一覧表示します。カスタムプリセットを一覧表示するには、前の例を参照してください。

```
aws mediaconvert list-presets \
  --list-by SYSTEM \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{
  "Presets": [
    {
      "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/System-Avc_16x9_1080p_29_97fps_8500kbps",
      "Name": "System-Avc_16x9_1080p_29_97fps_8500kbps",
      "CreatedAt": 1568321789,
      "Description": "Wifi, 1920x1080, 16:9, 29.97fps, 8500kbps",
      "LastUpdated": 1568321789,
      "Type": "SYSTEM",
      "Category": "HLS",
      "Settings": {
        ...<output settings removed for brevity>...
      }
    },
    ...<list of presets shortened for brevity>...
  ]
}
```

```
{
  "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:presets/System-
Xdcam_HD_1080i_29_97fps_35mpbs",
  "Name": "System-Xdcam_HD_1080i_29_97fps_35mpbs",
  "CreatedAt": 1568321790,
  "Description": "XDCAM MPEG HD, 1920x1080i, 29.97fps, 35mpbs",
  "LastUpdated": 1568321790,
  "Type": "SYSTEM",
  "Category": "MXF",
  "Settings": {
    ...<output settings removed for brevity>...
  }
}
]
```

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Output Presets の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPresets](#)」の「」を参照してください。

list-queues

次のコード例は、list-queues を使用する方法を示しています。

AWS CLI

キューを一覧表示するには

次のlist-queues例では、すべての MediaConvert キューを一覧表示します。

```
aws mediaconvert list-queues \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{
  "Queues": [
    {
      "PricingPlan": "ON_DEMAND",
      "Type": "SYSTEM",
```

```
    "Status": "ACTIVE",
    "CreatedAt": 1503451595,
    "Name": "Default",
    "SubmittedJobsCount": 0,
    "ProgressingJobsCount": 0,
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Default",
    "LastUpdated": 1534549158
  },
  {
    "PricingPlan": "ON_DEMAND",
    "Type": "CUSTOM",
    "Status": "ACTIVE",
    "CreatedAt": 1537460025,
    "Name": "Customer1",
    "SubmittedJobsCount": 0,
    "Description": "Jobs we run for our cusotmer.",
    "ProgressingJobsCount": 0,
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/Customer1",
    "LastUpdated": 1537460025
  },
  {
    "ProgressingJobsCount": 0,
    "Status": "ACTIVE",
    "Name": "transcode-library",
    "SubmittedJobsCount": 0,
    "LastUpdated": 1564066204,
    "ReservationPlan": {
      "Status": "ACTIVE",
      "ReservedSlots": 1,
      "PurchasedAt": 1564066203,
      "Commitment": "ONE_YEAR",
      "ExpiresAt": 1595688603,
      "RenewalType": "EXPIRE"
    },
    "PricingPlan": "RESERVED",
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:queues/transcode-
library",
    "Type": "CUSTOM",
    "CreatedAt": 1564066204
  }
]
}
```


詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Queues の使用](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[ListQueues](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

MediaConvert キュー、ジョブテンプレート、または出力プリセットのタグを一覧表示するには次のlist-tags-for-resource例では、指定された出力プリセットのタグを一覧表示します。

```
aws mediaconvert list-tags-for-resource \  
  --arn arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{  
  "ResourceTags": {  
    "Tags": {  
      "customer": "zippyVideo"  
    },  
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4"  
  }  
}
```

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Queues、ジョブテンプレート、および出力プリセットのタグ付け](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

update-job-template

次の例は、update-job-template を使用する方法を説明しています。

AWS CLI

ジョブテンプレートを変更するには

次のupdate-job-template例では、指定されたカスタムジョブテンプレートJSONの定義を、指定されたファイル内のJSON定義に置き換えます。

```
aws mediaconvert update-job-template --name File1 --endpoint-url https://
abcd1234.mediaconvert.us-west-2.amazonaws.com --cli-input-json file://~/job-template-
update.json
```

job-template-update.json の内容:

```
{
  "Description": "A simple job template that generates a single file output.",
  "Queue": "arn:aws:mediaconvert:us-east-1:012345678998:queues/Default",
  "Name": "SimpleFile",
  "Settings": {
    "OutputGroups": [
      {
        "Name": "File Group",
        "Outputs": [
          {
            "ContainerSettings": {
              "Container": "MP4",
              "Mp4Settings": {
                "CslgAtom": "INCLUDE",
                "FreeSpaceBox": "EXCLUDE",
                "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
              }
            }
          },
          {
            "VideoDescription": {
              "ScalingBehavior": "DEFAULT",
              "TimecodeInsertion": "DISABLED",
              "AntiAlias": "ENABLED",
              "Sharpness": 50,
              "CodecSettings": {
                "Codec": "H_264",
                "H264Settings": {
                  "InterlaceMode": "PROGRESSIVE",
                  "NumberReferenceFrames": 3,
                  "Syntax": "DEFAULT",
                  "Softness": 0,
                  "GopClosedCadence": 1,

```

```
    "GopSize": 90,
    "Slices": 1,
    "GopReference": "DISABLED",
    "SlowPal": "DISABLED",
    "SpatialAdaptiveQuantization": "ENABLED",
    "TemporalAdaptiveQuantization": "ENABLED",
    "FlickerAdaptiveQuantization": "DISABLED",
    "EntropyEncoding": "CABAC",
    "Bitrate": 400000,
    "FramerateControl": "INITIALIZE_FROM_SOURCE",
    "RateControlMode": "CBR",
    "CodecProfile": "MAIN",
    "Telecine": "NONE",
    "MinIInterval": 0,
    "AdaptiveQuantization": "HIGH",
    "CodecLevel": "AUTO",
    "FieldEncoding": "PAFF",
    "SceneChangeDetect": "ENABLED",
    "QualityTuningLevel": "SINGLE_PASS",
    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
    "UnregisteredSeiTimecode": "DISABLED",
    "GopSizeUnits": "FRAMES",
    "ParControl": "INITIALIZE_FROM_SOURCE",
    "NumberBFramesBetweenReferenceFrames": 2,
    "RepeatPps": "DISABLED",
    "DynamicSubGop": "STATIC"
  }
},
"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
"ColorMetadata": "INSERT"
},
"AudioDescriptions": [
  {
    "AudioTypeControl": "FOLLOW_INPUT",
    "CodecSettings": {
      "Codec": "AAC",
      "AacSettings": {
        "AudioDescriptionBroadcasterMix": "NORMAL",
        "Bitrate": 96000,
        "RateControlMode": "CBR",
        "CodecProfile": "LC",
        "CodingMode": "CODING_MODE_2_0",
```

```
        "RawFormat": "NONE",
        "SampleRate": 48000,
        "Specification": "MPEG4"
      }
    },
    "LanguageCodeControl": "FOLLOW_INPUT"
  }
]
}
],
"OutputGroupSettings": {
  "Type": "FILE_GROUP_SETTINGS",
  "FileGroupSettings": {}
}
},
"AdAvailOffset": 0
},
"StatusUpdateInterval": "SECONDS_60",
"Priority": 0
}
```

リクエストでエラーが発生した場合でも、システムはリクエストとともに送信したJSONペイロードを返します。したがって、JSON返される は必ずしもジョブテンプレートの新しい定義ではありません。

JSON ペイロードが長くなる可能性があるため、上にスクロールしてエラーメッセージを表示する必要がある場合があります。

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Job Templates の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateJobTemplate](#)」の「」を参照してください。

update-preset

次の例は、update-preset を使用する方法を説明しています。

AWS CLI

プリセットを変更するには

次のupdate-preset例では、指定されたプリセットの説明を置き換えます。

```
aws mediaconvert update-preset \  
--name Customer1 \  
--description "New description text."  
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

このコマンドでは何も出力されません。出力:

```
{  
  "Preset": {  
    "Arn": "arn:aws:mediaconvert:us-east-1:003235472598:presets/SimpleMP4",  
    "Settings": {  
      ...<output settings removed for brevity>...  
    },  
    "Type": "CUSTOM",  
    "LastUpdated": 1568938411,  
    "Description": "New description text.",  
    "Name": "SimpleMP4",  
    "CreatedAt": 1568938240  
  }  
}
```

詳細については、[「AWS Elemental ユーザーガイド」の「Elemental MediaConvert Output Presets の使用」](#)を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePreset](#)」の「」を参照してください。

update-queue

次の例は、update-queue を使用する方法を説明しています。

AWS CLI

キューを変更するには

次のupdate-queue例では、ステータスを に変更することで、指定されたキューを一時停止しますPAUSED。

```
aws mediaconvert update-queue \  

```

```
--name Customer1 \  
--status PAUSED  
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

出力:

```
{  
  "Queue": {  
    "LastUpdated": 1568839845,  
    "Status": "PAUSED",  
    "ProgressingJobsCount": 0,  
    "CreatedAt": 1526428516,  
    "Arn": "arn:aws:mediaconvert:us-west-1:123456789012:queues/Customer1",  
    "Name": "Customer1",  
    "SubmittedJobsCount": 0,  
    "PricingPlan": "ON_DEMAND",  
    "Type": "CUSTOM"  
  }  
}
```

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental MediaConvert Queues の使用](#)」を参照してください。AWS MediaConvert

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateQueue](#)」の「」を参照してください。

MediaLive を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています MediaLive。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-channel

次の例は、create-channel を使用する方法を説明しています。

AWS CLI

チャンネルを作成するには

次のcreate-channel例では、指定したいパラメータを含むJSONファイルを渡してチャンネルを作成します。

この例のチャンネルは、ビデオ、オーディオ、埋め込みキャプションを含むソースに接続するHLSPULL入力を取り込みます。チャンネルは、Akamai サーバーを送信先として1つのHLS出力グループを作成します。出力グループには、H.265 ビデオとAACオーディオ用の出力と、ウェブ字幕用の出力の2つの出力があります。1 VTTつは英語のみです。

この例のチャンネルJSONのには、HLSPULL入力を使用し、Akamai を宛先とするHLS出力グループを生成するチャンネルに必要な最小パラメータが含まれています。JSONには、次の主要なセクションが含まれています。

InputAttachments。オーディオのソースを1つ、字幕のソースを1つ指定します。ビデオセクタを指定しません。つまり、はソースで見つけた最初のビデオを MediaLive 抽出します。にはDestinations、このチャンネル内の単一の出力グループの2つのIPアドレス(URLs)が含まれます。これらのアドレスにはパスワードが必要です。EncoderSettingsには、のソースを使用するオーディオ出力アセットがチャンネルに1つ含まれ、AAC形式のオーディオInputAttachmentsを生成するAudioDescriptionsことを指定します。にはCaptionDescriptions、のソースを使用するキャプション出力アセットが1つ含まれInputAttachments、にはウェブVTT形式のキャプションが生成されます。にはVideoDescriptions、チャンネルに指定された解像度のビデオ出力アセットが1つ含まれていることを指定します。OutputGroupsには出力グループを指定します。この例では、という名前のグループが1つありますAkamai。接続はHLSを使用して行われますPUT。出力グループには2つの出力が含まれます。1つの出力は、ビデオアセット(という名前Video_high)とオーディオアセット(という名前)用ですAudio_EN。1つの出力は、字幕アセット(という名前)用ですWebVTT_EN。

この例では、一部のパラメータに値が含まれないか、ネストされた空のパラメータが含まれています。例えば、Video_and_audio出力 OutputSettingsには、空のパラメータ M3u8Settingsで終わるいくつかのネストされたパラメータが含まれています。このパラメータを含める必要があ

りますが、1つ、複数、またはすべての子を省略できます。つまり、子はデフォルト値を取るか null になります。

このサンプルチャンネルに適用されるパラメータのうち、このファイルで指定されていないパラメータはすべて、デフォルト値を使用するか、null に設定されるか、によって生成された一意の値を取得します MediaLive。

```
aws medialive create-channel \  
  --cli-input-json file://channel-in-hls-out-hls-akamai.json
```

channel-in-hls-out-hls-akamai.json の内容:

```
{  
  "Name": "News_West",  
  "RoleArn": "arn:aws:iam::111122223333:role/MediaLiveAccessRole",  
  "InputAttachments": [  
    {  
      "InputAttachmentName": "local_news",  
      "InputId": "1234567",  
      "InputSettings": {  
        "AudioSelectors": [  
          {  
            "Name": "English-Audio",  
            "SelectorSettings": {  
              "AudioLanguageSelection": {  
                "LanguageCode": "EN"  
              }  
            }  
          }  
        ],  
        "CaptionSelectors": [  
          {  
            "LanguageCode": "ENE",  
            "Name": "English_embedded"  
          }  
        ]  
      }  
    }  
  ],  
  "Destinations": [  
    {  
      "Id": "akamai-server-west",  
      "Settings": [  

```



```
        {
            "PasswordParam": "/medialive/examplecorp1",
            "Url": "http://203.0.113.55/news/news_west",
            "Username": "examplecorp"
        },
        {
            "PasswordParam": "/medialive/examplecorp2",
            "Url": "http://203.0.113.82/news/news_west",
            "Username": "examplecorp"
        }
    ]
},
"EncoderSettings": {
    "AudioDescriptions": [
        {
            "AudioSelectorName": "English-Audio",
            "CodecSettings": {
                "AacSettings": {}
            },
            "Name": "Audio_EN"
        }
    ],
    "CaptionDescriptions": [
        {
            "CaptionSelectorName": "English_embedded",
            "DestinationSettings": {
                "WebvttDestinationSettings": {}
            },
            "Name": "WebVTT_EN"
        }
    ],
    "VideoDescriptions": [
        {
            "Height": 720,
            "Name": "Video_high",
            "Width": 1280
        }
    ],
    "OutputGroups": [
        {
            "Name": "Akamai",
            "OutputGroupSettings": {
                "HlsGroupSettings": {
```

```
        "Destination": {
            "DestinationRefId": "akamai-server-west"
        },
        "HlsCdnSettings": {
            "HlsBasicPutSettings": {}
        }
    },
    "Outputs": [
        {
            "AudioDescriptionNames": [
                "Audio_EN"
            ],
            "OutputName": "Video_and_audio",
            "OutputSettings": {
                "HlsOutputSettings": {
                    "HlsSettings": {
                        "StandardHlsSettings": {
                            "M3u8Settings": {}
                        }
                    },
                    "NameModifier": "_1"
                }
            },
            "VideoDescriptionName": "Video_high"
        },
        {
            "CaptionDescriptionNames": [
                "WebVTT_EN"
            ],
            "OutputName": "Captions-WebVTT",
            "OutputSettings": {
                "HlsOutputSettings": {
                    "HlsSettings": {
                        "StandardHlsSettings": {
                            "M3u8Settings": {}
                        }
                    },
                    "NameModifier": "_2"
                }
            }
        }
    ]
}
```

```
    ],  
    "TimecodeConfig": {  
        "Source": "EMBEDDED"  
    }  
}  
}
```

出力:

出力は、JSONファイルの内容と以下の値を繰り返します。すべてのパラメータはアルファベット順に並べられます。

ARN チャンネル用。の最後の部分は一意的チャンネル ID ARNです。EgressEndpoints はPUSH 入力にのみ使用されるため、このサンプルチャンネルでは空白です。適用すると、MediaLive そのコンテンツのアドレスがプッシュされます。OutputGroups、Outputs。これらは、含めなかったが、このチャンネルに関連するものを含め、出力グループと出力のすべてのパラメータを表示します。パラメータは空(このチャンネル設定でパラメータまたは機能が無効化されていることを示す場合があります)であるか、適用されるデフォルト値が表示される場合があります。LogLevel はデフォルト () に設定されますDISABLED。Tags はデフォルト (null)PipelinesRunningCount に設定されます。はチャンネルの現在のステータスStateを表示します。

詳細については、「AWS Elemental MediaLive ユーザーガイド」の「[Scratch からのチャンネルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateChannel](#)」の「」を参照してください。

create-input

次の例は、create-input を使用する方法を説明しています。

AWS CLI

入力を作成するには

次のcreate-input例では、このタイプのHLS PULL入りに適用されるパラメータを含むJSON ファイルを渡すことで入力を作成します。この入力例JSONのは、取り込みの冗長性をサポートするために、入力への2つのソース(アドレス)を指定します。これらのアドレスにはパスワードが必要です。

```
aws medialive create-input \  
--cli-input-json file://input-hls-pull-news.json
```

input-hls-pull-news.json の内容:

```
{  
  "Name": "local_news",  
  "RequestId": "cli000059",  
  "Sources": [  
    {  
      "Url": "https://203.0.113.13/newschannel/anytownusa.m3u8",  
      "Username": "examplecorp",  
      "PasswordParam": "/medialive/examplecorp1"  
    },  
    {  
      "Url": "https://198.51.100.54/fillervideos/oceanwaves.mp4",  
      "Username": "examplecorp",  
      "PasswordParam": "examplecorp2"  
    }  
  ],  
  "Type": "URL_PULL"  
}
```

出力:

出力は、JSONファイルの内容と以下の値を繰り返します。すべてのパラメータはアルファベット順に並べられます。

Arn は入力用です。の最後の部分はAttached Channels一意の入力 ID ARNです。これは、新しく作成された入力では常に空です。この例では空です。Destinationsこれは PUSH入力でのみ使用されるためです。Idこの例では の ID と同じですARN。この例では空です。MediaConnectFlowsこれはタイプ の入力でのみ使用されます MediaConnect。この例ではSecurityGroups空です。この入力PUSHの入力 の入力でのみ使用されます。State Tagsは空です (このパラメータのデフォルト)。

詳細については、「[AWS Elemental MediaLive ユーザーガイド](#)」の「[入力の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateInput](#)」の「」を参照してください。

MediaPackage を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています MediaPackage。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-channel

次のコード例は、create-channel を使用する方法を示しています。

AWS CLI

チャンネルを作成するには

次のcreate-channelコマンドは、現在のアカウントに という名前sportschannelのチャンネルを作成します。

```
aws mediapackage create-channel --id sportschannel
```

出力:

```
{
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",
  "HlsIngest": {
    "IngestEndpoints": [
      {
        "Id": "6d345804ec3f46c9b454a91d4a80d0e0",
        "Password": "generatedwebdavpassword1",
```

```

        "Url": "https://f31c86aed53b815a.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",
        "Username": "generatedwebdavusername1"
    },
    {
        "Id": "2daa32878af24803b24183727211b8ff",
        "Password": "generatedwebdavpassword2",
        "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-
west-2.amazonaws.com/in/
v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",
        "Username": "generatedwebdavusername2"
    }
]
},
"Id": "sportschannel",
"Tags": {
    "region": "west"
}
}

```

詳細については、[「Elemental ユーザーガイド」の「チャンネルの作成」](#)を参照してください。

AWS MediaPackage

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateChannel](#)」の「」を参照してください。

create-origin-endpoint

次のコード例は、create-origin-endpoint を使用する方法を示しています。

AWS CLI

オリジンエンドポイントを作成するには

次のcreate-origin-endpointコマンドは、JSON ファイルで提供されるパッケージ設定と指定されたエンドポイント設定cmaf sportsを使用して、という名前のオリジンエンドポイントを作成します。

```

aws mediapackage create-origin-endpoint \
  --channel-id sportschannel \
  --id cmaf sports \

```

```
--cmf-package file:///file/path/cmafpkg.json --description "cmf output of sports" \  
--id cmf_sports \  
--manifest-name sports_channel \  
--startover-window-seconds 300 \  
--tags region=west,media=sports \  
--time-delay-seconds 10
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",  
  "ChannelId": "sportschannel",  
  "CmafPackage": {  
    "HlsManifests": [  
      {  
        "AdMarkers": "PASSTHROUGH",  
        "Id": "cmf_sports_endpoint",  
        "IncludeIframeOnlyStream": true,  
        "ManifestName": "index",  
        "PlaylistType": "EVENT",  
        "PlaylistWindowSeconds": 300,  
        "ProgramDateTimeIntervalSeconds": 300,  
        "Url": "https://c4af3793bf76b33c.mediapackage.us-west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/index.m3u8"  
      }  
    ],  
    "SegmentDurationSeconds": 2,  
    "SegmentPrefix": "sportschannel"  
  },  
  "Description": "cmf output of sports",  
  "Id": "cmf_sports",  
  "ManifestName": "sports_channel",  
  "StartoverWindowSeconds": 300,  
  "Tags": {  
    "region": "west",  
    "media": "sports"  
  },  
  "TimeDelaySeconds": 10,  
  "Url": "",  
  "Whitelist": []  
}
```

```
}
```

詳細については、「AWS Elemental MediaPackage ユーザーガイド」の「[エンドポイントの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateOriginEndpoint](#)」の「」を参照してください。

delete-channel

次の例は、delete-channel を使用する方法を説明しています。

AWS CLI

チャンネルを削除するには

次のdelete-channelコマンドは、 という名前のチャンネルを削除しますtest。

```
aws mediapackage delete-channel \  
  --id test
```

このコマンドでは何も出力されません。

詳細については、「AWS Elemental MediaPackage ユーザーガイド」の「[チャンネルの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteChannel](#)」の「」を参照してください。

delete-origin-endpoint

次のコード例は、delete-origin-endpoint を使用する方法を示しています。

AWS CLI

オリジンエンドポイントを削除するには

次のdelete-origin-endpointコマンドは、 という名前のオリジンエンドポイントを削除しますtester2。

```
aws mediapackage delete-origin-endpoint \  
  --id tester2
```


詳細については、「[Elemental ユーザーガイド](#)」の「[エンドポイントの削除](#)」を参照してください。AWS MediaPackage

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteOriginEndpoint](#)」の「」を参照してください。

describe-channel

次のコード例は、describe-channel を使用する方法を示しています。

AWS CLI

チャンネルを記述するには

次のdescribe-channelコマンドは、`test` という名前のチャンネルの詳細をすべて表示します。

```
aws mediapackage describe-channel \
  --id test
```

出力:

```
{
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:channels/584797f1740548c389a273585dd22a63",
  "HlsIngest": {
    "IngestEndpoints": [
      {
        "Id": "584797f1740548c389a273585dd22a63",
        "Password": "webdavgeneratedpassword1",
        "Url": "https://9be9c4405c474882.mediapackage.us-west-2.amazonaws.com/in/v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
        "Username": "webdavgeneratedusername1"
      },
      {
        "Id": "7d187c8616fd455f88aaa5a9fcf74442",
        "Password": "webdavgeneratedpassword2",
        "Url": "https://7bf454c57220328d.mediapackage.us-west-2.amazonaws.com/in/v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
        "Username": "webdavgeneratedusername2"
      }
    ]
  }
}
```

```
    ]
  },
  "Id": "test",
  "Tags": {}
}
```

詳細については、「AWS Elemental MediaPackage ユーザーガイド」の「チャンネルの詳細の表示 <<https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html>>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeChannel](#)」の「」を参照してください。

describe-origin-endpoint

次のコード例は、describe-origin-endpoint を使用する方法を示しています。

AWS CLI

オリジンエンドポイントを記述するには

次のdescribe-origin-endpointコマンドは、という名前のオリジンエンドポイントのすべての詳細を表示しますcmaf_sports。

```
aws mediapackage describe-origin-endpoint \
  --id cmaf_sports
```

出力:

```
{
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",
  "ChannelId": "sportschannel",
  "CmafPackage": {
    "HlsManifests": [
      {
        "AdMarkers": "NONE",
        "Id": "cmaf_sports_endpoint",
        "IncludeIframeOnlyStream": false,
        "PlaylistType": "EVENT",
        "PlaylistWindowSeconds": 60,
        "ProgramDateTimeIntervalSeconds": 0,

```

```
        "Url": "https://c4af3793bf76b33c.mediapackage.us-  
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/  
index.m3u8"  
      }  
    ],  
    "SegmentDurationSeconds": 2,  
    "SegmentPrefix": "sportschannel"  
  },  
  "Id": "cmaf_sports",  
  "ManifestName": "index",  
  "StartoverWindowSeconds": 0,  
  "Tags": {  
    "region": "west",  
    "media": "sports"  
  },  
  "TimeDelaySeconds": 0,  
  "Url": "",  
  "Whitelist": []  
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[単一のエンドポイントの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOriginEndpoint](#)」の「」を参照してください。

list-channels

次のコード例は、list-channels を使用する方法を示しています。

AWS CLI

すべてのチャンネルを一覧表示するには

次のlist-channelsコマンドは、現在の AWS アカウントに設定されているすべてのチャンネルを一覧表示します。

```
aws mediapackage list-channels
```

出力:

```
{
```

```
"Channels": [
  {
    "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
    "HlsIngest": {
      "IngestEndpoints": [
        {
          "Id": "584797f1740548c389a273585dd22a63",
          "Password": "webdavgeneratedpassword1",
          "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
          "Username": "webdavgeneratedusername1"
        },
        {
          "Id": "7d187c8616fd455f88aaa5a9fcf74442",
          "Password": "webdavgeneratedpassword2",
          "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
          "Username": "webdavgeneratedusername2"
        }
      ]
    },
    "Id": "test",
    "Tags": {}
  }
]
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[チャンネルの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListChannels](#)」の「」を参照してください。

list-origin-endpoints

次の例は、list-origin-endpoints を使用する方法を説明しています。

AWS CLI

チャンネルのすべてのオリジンエンドポイントを一覧表示するには

次の `list-origin-endpoints` コマンドは、`test` という名前のチャンネルに設定されているすべてのオリジンエンドポイントを一覧表示します。

```
aws mediapackage list-origin-endpoints \  
  --channel-id test
```

出力:

```
{  
  "OriginEndpoints": [  
    {  
      "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/247cff871f2845d3805129be22f2c0a2",  
      "ChannelId": "test",  
      "DashPackage": {  
        "ManifestLayout": "FULL",  
        "ManifestWindowSeconds": 60,  
        "MinBufferTimeSeconds": 30,  
        "MinUpdatePeriodSeconds": 15,  
        "PeriodTriggers": [],  
        "Profile": "NONE",  
        "SegmentDurationSeconds": 2,  
        "SegmentTemplateFormat": "NUMBER_WITH_TIMELINE",  
        "StreamSelection": {  
          "MaxVideoBitsPerSecond": 2147483647,  
          "MinVideoBitsPerSecond": 0,  
          "StreamOrder": "ORIGINAL"  
        },  
        "SuggestedPresentationDelaySeconds": 25  
      },  
      "Id": "tester2",  
      "ManifestName": "index",  
      "StartoverWindowSeconds": 0,  
      "Tags": {},  
      "TimeDelaySeconds": 0,  
      "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/out/v1/247cff871f2845d3805129be22f2c0a2/index.mpd",  
      "Whitelist": []  
    },  
    {  
      "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/869e237f851549e9bcf10e3bc2830839",  
      "ChannelId": "test",
```

```

    "HlsPackage": {
      "AdMarkers": "NONE",
      "IncludeIframeOnlyStream": false,
      "PlaylistType": "EVENT",
      "PlaylistWindowSeconds": 60,
      "ProgramDateTimeIntervalSeconds": 0,
      "SegmentDurationSeconds": 6,
      "StreamSelection": {
        "MaxVideoBitsPerSecond": 2147483647,
        "MinVideoBitsPerSecond": 0,
        "StreamOrder": "ORIGINAL"
      },
      "UseAudioRenditionGroup": false
    },
    "Id": "tester",
    "ManifestName": "index",
    "StartoverWindowSeconds": 0,
    "Tags": {},
    "TimeDelaySeconds": 0,
    "Url": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/
out/v1/869e237f851549e9bcf10e3bc2830839/index.m3u8",
    "Whitelist": []
  }
]
}

```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[チャンネルに関連付けられたすべてのエンドポイントの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListOriginEndpoints](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースに割り当てられたタグを一覧表示するには

次のlist-tags-for-resourceコマンドは、指定されたリソースに割り当てられたタグを一覧表示します。

```
aws mediapackage list-tags-for-resource \  
  --resource-arn arn:aws:mediapackage:us-  
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0
```

出力:

```
{  
  "Tags": {  
    "region": "west"  
  }  
}
```

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental でのリソースのタグ付け MediaPackage](#)」を参照してください。AWS MediaPackage

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

rotate-ingest-endpoint-credentials

次の例は、rotate-ingest-endpoint-credentials を使用する方法を説明しています。

AWS CLI

取り込み認証情報をローテーションするには

次のrotate-ingest-endpoint-credentialsコマンドは、指定された取り込みエンドポイントのウェブDAVユーザー名とパスワードをローテーションします。

```
aws mediapackage rotate-ingest-endpoint-credentials \  
  --id test \  
  --ingest-endpoint-id 584797f1740548c389a273585dd22a63
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage:us-  
west-2:111222333:channels/584797f1740548c389a273585dd22a63",  
  "HlsIngest": {  
    "IngestEndpoints": [  
      {
```

```

        "Id": "584797f1740548c389a273585dd22a63",
        "Password": "webdavregeneratedpassword1",
        "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
        "Username": "webdavregeneratedusername1"
    },
    {
        "Id": "7d187c8616fd455f88aaa5a9fcf74442",
        "Password": "webdavgeneratedpassword2",
        "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
        "Username": "webdavgeneratedusername2"
    }
]
},
"Id": "test",
"Tags": {}
}

```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[入力での認証情報のローテーションURL](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスRotateIngestEndpointCredentials](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを追加

次のtag-resourceコマンドは、指定されたリソースにregion=westキーと値のペアを追加します。

```

aws mediapackage tag-resource \
  --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
  --tags region=west

```


このコマンドでは何も出力されません。

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental でのリソースのタグ付け MediaPackage](#)」を参照してください。AWS MediaPackage

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグを削除するには

次のuntag-resourceコマンドは、指定されたチャネルregionから キーを持つ タグを削除します。

```
aws mediapackage untag-resource \  
  --resource-arn arn:aws:mediapackage:us-  
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \  
  --tag-keys region
```

詳細については、「[AWS Elemental ユーザーガイド](#)」の「[Elemental でのリソースのタグ付け MediaPackage](#)」を参照してください。AWS MediaPackage

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-channel

次のコード例は、update-channel を使用する方法を示しています。

AWS CLI

チャンネルを更新するには

次のupdate-channelコマンドは、`という名前のチャンネルを更新sportschannelして、説明を含めます24x7 sports。`

```
aws mediapackage update-channel \  
  --channel-name sportschannel \  
  --description 24x7 sports
```

```
--id sportschannel \  
--description "24x7 sports"
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage:us-  
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0",  
  "Description": "24x7 sports",  
  "HlsIngest": {  
    "IngestEndpoints": [  
      {  
        "Id": "6d345804ec3f46c9b454a91d4a80d0e0",  
        "Password": "generatedwebdavpassword1",  
        "Url": "https://f31c86aed53b815a.mediapackage.us-  
west-2.amazonaws.com/in/  
v2/6d345804ec3f46c9b454a91d4a80d0e0/6d345804ec3f46c9b454a91d4a80d0e0/channel",  
        "Username": "generatedwebdavusername1"  
      },  
      {  
        "Id": "2daa32878af24803b24183727211b8ff",  
        "Password": "generatedwebdavpassword2",  
        "Url": "https://6ebbe7e04c4b0afa.mediapackage.us-  
west-2.amazonaws.com/in/  
v2/6d345804ec3f46c9b454a91d4a80d0e0/2daa32878af24803b24183727211b8ff/channel",  
        "Username": "generatedwebdavusername2"  
      }  
    ]  
  },  
  "Id": "sportschannel",  
  "Tags": {}  
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[チャンネルの編集](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス UpdateChannel](#)」の「」を参照してください。

update-origin-endpoint

次のコード例は、update-origin-endpoint を使用する方法を示しています。

AWS CLI

オリジンエンドポイントを更新するには

次のupdate-origin-endpointコマンドは、という名前のオリジンエンドポイントを更新しますcmf_sports。遅延時間を0秒に変更します。

```
aws mediapackage update-origin-endpoint \  
  --id cmf_sports \  
  --time-delay-seconds 0
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",  
  "ChannelId": "sportschannel",  
  "CmafPackage": {  
    "HlsManifests": [  
      {  
        "AdMarkers": "NONE",  
        "Id": "cmf_sports_endpoint",  
        "IncludeIframeOnlyStream": false,  
        "PlaylistType": "EVENT",  
        "PlaylistWindowSeconds": 60,  
        "ProgramDateTimeIntervalSeconds": 0,  
        "Url": "https://c4af3793bf76b33c.mediapackage.us-west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmf_sports_endpoint/index.m3u8"  
      }  
    ],  
    "SegmentDurationSeconds": 2,  
    "SegmentPrefix": "sportschannel"  
  },  
  "Id": "cmf_sports",  
  "ManifestName": "index",  
  "StartoverWindowSeconds": 0,  
  "Tags": {  
    "region": "west",  
    "media": "sports"  
  },  
  "TimeDelaySeconds": 0,  
  "Url": "",
```

```
"Whitelist": []
}
```

詳細については、「AWS Elemental MediaPackage ユーザーガイド」の「[エンドポイントの編集](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateOriginEndpoint](#)」の「」を参照してください。

MediaPackage VOD を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています MediaPackage VOD。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-asset

次のコード例は、create-asset を使用する方法を示しています。

AWS CLI

アセットを作成するには

次のcreate-asset例では、現在のAWSアカウントに という名前Chicken_Assetのアセットを作成します。アセットは30sec_chicken.smilにファイルを取り込みます MediaPackage。

```
aws mediapackage-vod create-asset \  
  --id chicken_asset \  
  --
```

```
--packaging-group-id hls_chicken_gp \  
--source-role-arn arn:aws:iam::111122223333:role/EMP_Vod \  
--source-arn arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:assets/chicken_asset",  
  "Id": "chicken_asset",  
  "PackagingGroupId": "hls_chicken_gp",  
  "SourceArn": "arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",  
  "SourceRoleArn": "arn:aws:iam::111122223333:role/EMP_Vod",  
  "EgressEndpoints": [  
    {  
      "PackagingConfigurationId": "New_config_1",  
      "Url": "https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-  
vod.us-west-2.amazonaws.com/out/  
v1/6644b55df1744261ab3732a8e5cdaf07/904b06a58c7645e08d57d40d064216ac/  
f5b2e633ff4942228095d164c10074f3/index.m3u8"  
    },  
    {  
      "PackagingConfigurationId": "new_hls",  
      "Url": "https://c75ea2668ab49d02bca7ae10ef31c59e.egress.mediapackage-  
vod.us-west-2.amazonaws.com/out/v1/6644b55df1744261ab3732a8e5cdaf07/  
fe8f1f00a80e424cb4f8da4095835e9e/7370ec57432343af816332356d2bd5c6/string.m3u8"  
    }  
  ]  
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[アセットの取り込み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateAsset](#)」の「」を参照してください。

create-packaging-configuration

次の例は、create-packaging-configuration を使用する方法を説明しています。

AWS CLI

パッケージ設定を作成するには

次のcreate-packaging-configuration例では、という名前new_hlsのパッケージンググループにという名前のパッケージング設定を作成しますhls_chicken。この例では、という名前のディスク上のファイルを使用して詳細hls_pc.jsonを提供します。

```
aws mediapackage-vod create-packaging-configuration \  
  --id new_hls \  
  --packaging-group-id hls_chicken \  
  --hls-package file://hls_pc.json
```

hls_pc.json の内容:

```
{  
  "HlsManifests": [  
    {  
      "AdMarkers": "NONE",  
      "IncludeIframeOnlyStream": false,  
      "ManifestName": "string",  
      "ProgramDateTimeIntervalSeconds": 60,  
      "RepeatExtXKey": true,  
      "StreamSelection": {  
        "MaxVideoBitsPerSecond": 1000,  
        "MinVideoBitsPerSecond": 0,  
        "StreamOrder": "ORIGINAL"  
      }  
    }  
  ],  
  "SegmentDurationSeconds": 6,  
  "UseAudioRenditionGroup": false  
}
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/  
new_hls",  
  "Id": "new_hls",  
  "PackagingGroupId": "hls_chicken",  
  "HlsManifests": {  
    "SegmentDurationSeconds": 6,  
    "UseAudioRenditionGroup": false,  
    "HlsMarkers": [  
      {
```

```
    "AdMarkers": "NONE",
    "IncludeIframeOnlyStream": false,
    "ManifestName": "string",
    "ProgramDateTimeIntervalSeconds": 60,
    "RepeatExtXKey": true,
    "StreamSelection": {
      "MaxVideoBitsPerSecond": 1000,
      "MinVideoBitsPerSecond": 0,
      "StreamOrder": "ORIGINAL"
    }
  ]
}
```

詳細については、「AWS Elemental MediaPackage ユーザーガイド」の「[パッケージング設定の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePackagingConfiguration](#)」の「」を参照してください。

create-packaging-group

次のコード例は、create-packaging-group を使用する方法を示しています。

AWS CLI

パッケージンググループを作成するには

次のcreate-packaging-group例では、現在の AWS アカウントで設定されているすべてのパッケージンググループを一覧表示します。

```
aws mediapackage-vod create-packaging-group \
  --id hls_chicken
```

出力:

```
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/hls_chicken",
  "Id": "hls_chicken"
}
```

詳細については、「AWS Elemental MediaPackage ユーザーガイド」の「[パッケージンググループの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePackagingGroup](#)」の「」を参照してください。

delete-asset

次のコード例は、delete-asset を使用する方法を示しています。

AWS CLI

アセットを削除するには

次のdelete-asset例では、 という名前のアセットを削除します30sec_chicken。

```
aws mediapackage-vod delete-asset \  
  --id 30sec_chicken
```

このコマンドでは何も出力されません。

詳細については、「Elemental ユーザーガイド」の「[アセットの削除](#)」を参照してください。

AWS MediaPackage

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAsset](#)」の「」を参照してください。

delete-packaging-configuration

次のコード例は、delete-packaging-configuration を使用する方法を示しています。

AWS CLI

パッケージング設定を削除するには

次のdelete-packaging-configuration例では、 という名前のパッケージング設定を削除しますCMAF。

```
aws mediapackage-vod delete-packaging-configuration \  
  --id CMAF
```

このコマンドでは何も出力されません。

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[パッケージング設定の削除](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeletePackagingConfiguration](#)」の「」を参照してください。

delete-packaging-group

次の例は、delete-packaging-group を使用する方法を説明しています。

AWS CLI

パッケージンググループを削除するには

次のdelete-packaging-group例では、`dash_widevine` という名前のパッケージンググループを削除します。

```
aws mediapackage-vod delete-packaging-group \  
  --id Dash_widevine
```

このコマンドでは何も出力されません。

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[パッケージンググループの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeletePackagingGroup](#)」の「」を参照してください。

describe-asset

次のコード例は、describe-asset を使用する方法を示しています。

AWS CLI

アセットを記述するには

次のdescribe-asset例では、`30sec_chicken` という名前のアセットの詳細をすべて表示します。

```
aws mediapackage-vod describe-asset \  
  --id 30sec_chicken
```

出力:

```
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:assets/30sec_chicken",
  "Id": "30sec_chicken",
  "PackagingGroupId": "Packaging_group_1",
  "SourceArn": "arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
  "SourceRoleArn": "arn:aws:iam::111122223333:role/EMP_Vod",
  "EgressEndpoints": [
    {
      "PackagingConfigurationId": "DASH",
      "Url": "https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/66c25aff456d463aae0855172b3beb27/4ddfda6da17c4c279a1b8401cb
index.mpd"
    },
    {
      "PackagingConfigurationId": "HLS",
      "Url": "https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/6e5bf286a3414254a2bf0d22ae148d7e/06b5875b4d004c3cbdc4da2dc4
index.m3u8"
    },
    {
      "PackagingConfigurationId": "CMAF",
      "Url": "https://a5f46a44118ba3e3724ef39ef532e701.egress.mediapackage-
vod.us-west-2.amazonaws.com/out/v1/
aad7962c569946119c2d5a691be5663c/628fb5d8d89e4702958b020af27fde0e/05eb062214064238ad6330a443
index.m3u8"
    }
  ]
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[アセットの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAsset](#)」の「」を参照してください。

describe-packaging-configuration

次の例は、describe-packaging-configuration を使用する方法を説明しています。

AWS CLI

パッケージング設定を記述するには

次のdescribe-packaging-configuration例では、 という名前のパッケージング設定の詳細をすべて表示しますDASH。

```
aws mediapackage-vod describe-packaging-configuration \  
  --id DASH
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-configurations/  
DASH",  
  "Id": "DASH",  
  "PackagingGroupId": "Packaging_group_1",  
  "DashPackage": [  
    {  
      "SegmentDurationSeconds": "2"  
    },  
    {  
      "DashManifests": {  
        "ManifestName": "index",  
        "MinBufferTimeSeconds": "30",  
        "Profile": "NONE"  
      }  
    }  
  ]  
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[パッケージング設定の詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribePackagingConfiguration](#)」の「」を参照してください。

describe-packaging-group

次のコード例は、describe-packaging-group を使用する方法を示しています。

AWS CLI

パッケージンググループを記述するには

次のdescribe-packaging-group例では、`Packaging_group_1`という名前のパッケージンググループの詳細をすべて表示しますPackaging_group_1。

```
aws mediapackage-vod describe-packaging-group \  
  --id Packaging_group_1
```

出力:

```
{  
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/  
Packaging_group_1",  
  "Id": "Packaging_group_1"  
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[パッケージンググループの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePackagingGroup](#)」の「」を参照してください。

list-assets

次の例は、list-assets を使用する方法を説明しています。

AWS CLI

すべてのアセットを一覧表示するには

次のlist-assets例では、現在の AWS アカウントに設定されているすべてのアセットを一覧表示します。

```
aws mediapackage-vod list-assets
```

出力:

```
{  
  "Assets": [  
    ...  
  ]  
}
```

```
    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:assets/30sec_chicken",
    "Id": "30sec_chicken",
    "PackagingGroupId": "Packaging_group_1",
    "SourceArn": "arn:aws:s3::111122223333:video-bucket/A/30sec_chicken.smil",
    "SourceRoleArn": "arn:aws:iam::111122223333:role/EMP_Vod"
  ]
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[アセットの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssets](#)」の「」を参照してください。

list-packaging-configurations

次の例は、list-packaging-configurations を使用する方法を説明しています。

AWS CLI

すべてのパッケージング設定を一覧表示するには

次のlist-packaging-configurations例では、という名前のパッケージンググループに設定されているすべてのパッケージング設定を一覧表示しますPackaging_group_1。

```
aws mediapackage-vod list-packaging-configurations \
  --packaging-group-id Packaging_group_1
```

出力:

```
{
  "PackagingConfigurations": [
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/CMAF",
      "Id": "CMAF",
      "PackagingGroupId": "Packaging_group_1",
      "CmafPackage": [
        {
          "SegmentDurationSeconds": "2"
        },
        {
```

```

        "HlsManifests":{
            "AdMarkers":"NONE",
            "RepeatExtXKey":"False",
            "ManifestName":"index",
            "ProgramDateTimeIntervalSeconds":"0",
            "IncludeIframeOnlyStream":"False"
        }
    ]
},
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/DASH",
    "Id":"DASH",
    "PackagingGroupId":"Packaging_group_1",
    "DashPackage":[
        {
            "SegmentDurationSeconds":"2"
        },
        {
            "DashManifests":{
                "ManifestName":"index",
                "MinBufferTimeSeconds":"30",
                "Profile":"NONE"
            }
        }
    ]
},
{
    "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/HLS",
    "Id":"HLS",
    "PackagingGroupId":"Packaging_group_1",
    "HlsPackage":[
        {
            "SegmentDurationSeconds":"6",
            "UseAudioRenditionGroup":"False"
        },
        {
            "HlsManifests":{
                "AdMarkers":"NONE",
                "RepeatExtXKey":"False",
                "ManifestName":"index",
                "ProgramDateTimeIntervalSeconds":"0",

```

```
        "IncludeIframeOnlyStream":"False"
      }
    }
  ]
},
{
  "Arn":"arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-
configurations/New_config_0_copy",
  "Id":"New_config_0_copy",
  "PackagingGroupId":"Packaging_group_1",
  "HlsPackage":[
    {
      "SegmentDurationSeconds":"6",
      "UseAudioRenditionGroup":"False"
    },
    {
      "Encryption":{
        "EncryptionMethod":"AWS_128",
        "SpekeKeyProvider":{
          "RoleArn":"arn:aws:iam:111122223333::role/SPEKERole",
          "Url":"https://lfgubdvs97.execute-api.us-
west-2.amazonaws.com/EkeStage/copyProtection/",
          "SystemIds":[
            "81376844-f976-481e-a84e-cc25d39b0b33"
          ]
        }
      }
    },
    {
      "HlsManifests":{
        "AdMarkers":"NONE",
        "RepeatExtXKey":"False",
        "ManifestName":"index",
        "ProgramDateTimeIntervalSeconds":"0",
        "IncludeIframeOnlyStream":"False"
      }
    }
  ]
}
]
}
```

詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[パッケージング設定の詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListPackagingConfigurations](#)」の「」を参照してください。

list-packaging-groups

次のコード例は、list-packaging-groups を使用する方法を示しています。

AWS CLI

すべてのパッケージンググループを一覧表示するには

次のlist-packaging-groups例では、現在の AWS アカウントで設定されているすべてのパッケージンググループを一覧表示します。

```
aws mediapackage-vod list-packaging-groups
```

出力:

```
{
  "PackagingGroups": [
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/Dash_widevine",
      "Id": "Dash_widevine"
    },
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/Encrypted_HLS",
      "Id": "Encrypted_HLS"
    },
    {
      "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/Packaging_group_1",
      "Id": "Packaging_group_1"
    }
  ]
}
```


詳細については、「[AWS Elemental MediaPackage ユーザーガイド](#)」の「[パッケージンググループの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListPackagingGroups](#)」の「」を参照してください。

MediaStore を使用したデータプレーンの例 AWS CLI

次のコード例は、MediaStore Data Plane AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-object

次のコード例は、delete-object を使用する方法を示しています。

AWS CLI

オブジェクトを削除するには

次のdelete-object例では、指定されたオブジェクトを削除します。

```
aws mediastore-data delete-object \  
  --endpoint=https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path=/folder_name/README.md
```

このコマンドでは何も出力されません。

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の「[オブジェクトの削除](#)」を参照してください。


```
aws mediastore-data get-object \  
  --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball/setup.jpg setup.jpg
```

出力:

```
{  
  "ContentType": "image/jpeg",  
  "StatusCode": 200,  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f5555555555555555da6d3",  
  "ContentLength": "3860266",  
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT"  
}
```

例 2: オブジェクトの一部をダウンロードするには

次のget-object例では、オブジェクトの指定された部分をダウンロードします。

```
aws mediastore-data get-object \  
  --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball/setup.jpg setup.jpg \  
  --range "bytes=0-100"
```

出力:

```
{  
  "StatusCode": 206,  
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",  
  "ContentType": "image/jpeg",  
  "ContentRange": "bytes 0-100/3860266",  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f5555555555555555da6d3",  
  "ContentLength": "101"  
}
```

詳細については、「AWS Elemental MediaStore ユーザーガイド」の[「オブジェクトのダウンロード」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetObject](#)」の「」を参照してください。

list-items

次の例は、list-items を使用する方法を説明しています。

AWS CLI

例 1: コンテナに保存されている項目 (オブジェクトとフォルダ) のリストを表示するには

次のlist-items例では、指定されたコンテナに保存されている項目 (オブジェクトとフォルダ) のリストを表示します。

```
aws mediastore-data list-items \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com
```

出力:

```
{  
  "Items": [  
    {  
      "Type": "OBJECT",  
      "ContentLength": 3784,  
      "Name": "setup.jpg",  
      "ETag":  
      "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3",  
      "ContentType": "image/jpeg",  
      "LastModified": 1563571859.379  
    },  
    {  
      "Type": "FOLDER",  
      "Name": "events"  
    }  
  ]  
}
```

例 2: フォルダに保存されている項目 (オブジェクトとフォルダ) のリストを表示するには

次のlist-items例では、指定されたフォルダに保存されている項目 (オブジェクトとフォルダ) のリストを表示します。

```
aws mediastore-data list-items \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball
```

出力:

```
{
  "Items": [
    {
      "ETag":
"2aa333bbcc8d8d22d777e999c88d4aa9eeeeeee4dd89ff7f55555555555555555da6d3",
      "ContentType": "image/jpeg",
      "Type": "OBJECT",
      "ContentLength": 3860266,
      "LastModified": 1563573031.872,
      "Name": "setup.jpg"
    }
  ]
}
```

詳細については、「AWS Elemental MediaStore ユーザーガイド」の「[オブジェクトのリストの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListItems](#)」の「」を参照してください。

put-object

次のコード例は、put-object を使用方法を示しています。

AWS CLI

例 1: オブジェクトをコンテナにアップロードするには

次のput-object例では、指定されたコンテナにオブジェクトをアップロードします。

```
aws mediastore-data put-object \
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \
  --body ReadMe.md \
  --path ReadMe.md \
  --cache-control "max-age=6, public" \
  --content-type binary/octet-stream
```

出力:

```
{
```

```

    "ContentSHA256":
      "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
      "StorageClass": "TEMPORAL",
      "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeeee4dd89ff7f555555555555da6d3"
  }

```

例 2: オブジェクトをコンテナ内のフォルダにアップロードするには

次のput-object例では、コンテナ内の指定されたフォルダにオブジェクトをアップロードします。

```

aws mediastore-data put-object \
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \
  --body ReadMe.md \
  --path /september-events/ReadMe.md \
  --cache-control "max-age=6, public" \
  --content-type binary/octet-stream

```

出力:

```

{
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeeee4dd89ff7f555555555555da6d3",
  "ContentSHA256":
    "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
    "StorageClass": "TEMPORAL"
}

```

詳細については、「[AWS Elemental MediaStore ユーザーガイド](#)」の「[オブジェクトのアップロード](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutObject](#)」の「」を参照してください。

MediaTailor を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています MediaTailor。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-playback-configuration

次のコード例は、delete-playback-configuration を使用する方法を示しています。

AWS CLI

設定を削除するには

以下は、という名前の設定delete-playback-configurationを削除しますcampaign_short。

```
aws mediatailor delete-playback-configuration \  
  --name campaign_short
```

このコマンドでは何も出力されません。

詳細については、「[AWS Elemental MediaTailor ユーザーガイド](#)」の「[設定の削除](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeletePlaybackConfiguration](#)」の「」を参照してください。

get-playback-configuration

次の例は、get-playback-configuration を使用する方法を説明しています。

AWS CLI

設定を記述するには

以下は、という名前の設定の詳細をすべてget-playback-configuration表示しますwest_campaign。

```
aws mediatailor get-playback-configuration \  
  --name west_campaign
```

出力:

```
{  
  "AdDecisionServerUrl": "http://your.ads.url",  
  "CdnConfiguration": {},  
  "DashConfiguration": {  
    "ManifestEndpointPrefix":  
    "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/  
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",  
    "MpdLocation": "EMT_DEFAULT",  
    "OriginManifestType": "MULTI_PERIOD"  
  },  
  "HlsConfiguration": {  
    "ManifestEndpointPrefix":  
    "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/  
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"  
  },  
  "Name": "west_campaign",  
  "PlaybackConfigurationArn": "arn:aws:mediatailor:us-  
west-2:123456789012:playbackConfiguration/west_campaign",  
  "PlaybackEndpointPrefix":  
  "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",  
  "SessionInitializationEndpointPrefix":  
  "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/  
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",  
  "Tags": {},  
  "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-  
west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"  
}
```

詳細については、「[AWS Elemental MediaTailor ユーザーガイド](#)」の「[設定の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetPlaybackConfiguration](#)」の「」を参照してください。

list-playback-configurations

次の例は、list-playback-configurations を使用する方法を説明しています。

AWS CLI

すべての設定を一覧表示するには

以下は、現在の AWS アカウントの設定のすべての詳細 `list-playback-configurations` を表示します。

```
aws mediatailor list-playback-configurations
```

出力:

```
{
  "Items": [
    {
      "AdDecisionServerUrl": "http://your.ads.url",
      "CdnConfiguration": {},
      "DashConfiguration": {
        "ManifestEndpointPrefix":
          "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
        "MpdLocation": "EMT_DEFAULT",
        "OriginManifestType": "MULTI_PERIOD"
      },
      "HlsConfiguration": {
        "ManifestEndpointPrefix":
          "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/"
      },
      "Name": "west_campaign",
      "PlaybackConfigurationArn": "arn:aws:mediatailor:us-west-2:123456789012:playbackConfiguration/west_campaign",
      "PlaybackEndpointPrefix":
        "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com",
      "SessionInitializationEndpointPrefix":
        "https://170c14299689462897d0cc45fc2000bb.mediatailor.us-west-2.amazonaws.com/v1/session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/west_campaign/",
      "Tags": {},
      "VideoContentSourceUrl": "https://8343f7014c0ea438.mediapackage.us-west-2.amazonaws.com/out/v1/683f0f2ff7cd43a48902e6dcd5e16dcf/index.m3u8"
    },
    {
      "AdDecisionServerUrl": "http://your.ads.url",
      "CdnConfiguration": {},

```

```

    "DashConfiguration": {
      "ManifestEndpointPrefix":
        "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediataylor.us-west-2.amazonaws.com/v1/
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
      "MpdLocation": "DISABLED",
      "OriginManifestType": "MULTI_PERIOD"
    },
    "HlsConfiguration": {
      "ManifestEndpointPrefix":
        "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediataylor.us-west-2.amazonaws.com/v1/
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/"
    },
    "Name": "sports_campaign",
    "PlaybackConfigurationArn": "arn:aws:mediataylor:us-
west-2:123456789012:playbackConfiguration/sports_campaign",
    "PlaybackEndpointPrefix":
      "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediataylor.us-west-2.amazonaws.com",
    "SessionInitializationEndpointPrefix":
      "https://73511f91d6a24ca2b93f3cf1d7cedd67.mediataylor.us-west-2.amazonaws.com/v1/
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/sports_campaign/",
    "SlateAdUrl": "http://s3.bucket/slate_ad.mp4",
    "Tags": {},
    "VideoContentSourceUrl": "https://c4af3793bf76b33c.mediapackage.us-
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/sports_endpoint/
index.m3u8"
  }
]
}

```

詳細については、「AWS Elemental MediaTailor ユーザーガイド」の「Configuration<<https://docs.aws.amazon.com/mediataylor/latest/ug/configurations-view.html>> の表示」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPlaybackConfigurations](#)」の「」を参照してください。

put-playback-configuration

次のコード例は、put-playback-configuration を使用する方法を示しています。

AWS CLI

設定を作成するには

次に、という名前の設定put-playback-configurationを作成しますcampaign_short。

```
aws mediatailor put-playback-configuration \  
  --name campaign_short \  
  --ad-decision-server-url http://your.ads.url \  
  --video-content-source-url http://video.bucket/index.m3u8
```

出力:

```
{  
  "AdDecisionServerUrl": "http://your.ads.url",  
  "CdnConfiguration": {},  
  "DashConfiguration": {  
    "ManifestEndpointPrefix":  
      "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/  
dash/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",  
    "MpdLocation": "EMT_DEFAULT",  
    "OriginManifestType": "MULTI_PERIOD"  
  },  
  "HlsConfiguration": {  
    "ManifestEndpointPrefix":  
      "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/  
master/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/"  
  },  
  "Name": "campaign_short",  
  "PlaybackConfigurationArn": "arn:aws:mediatailor:us-  
west-2:123456789012:playbackConfiguration/campaign_short",  
  "PlaybackEndpointPrefix":  
    "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com",  
  "SessionInitializationEndpointPrefix":  
    "https://13484114d38f4383bc0d6a7cb879bd00.mediatailor.us-west-2.amazonaws.com/v1/  
session/1cbfeaaecb69778e0c167d0505a2bc57da2b1754/campaign_short/",  
  "Tags": {},  
  "VideoContentSourceUrl": "http://video.bucket/index.m3u8"  
}
```

詳細については、「[AWS Elemental MediaTailor ユーザーガイド](#)」の「[設定の作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutPlaybackConfiguration](#)」の「」を参照してください。

を使用した MemoryDB の例 AWS CLI

次のコード例は、MemoryDB AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

copy-snapshot

次の例は、copy-snapshot を使用する方法を説明しています。

AWS CLI

スナップショットをコピーするには

次のcopy-snapshot例では、スナップショットのコピーを作成します。

```
aws memorydb copy-snapshot \  
  --source-snapshot-name my-cluster-snapshot \  
  --target-snapshot-name my-cluster-snapshot-copy
```

出力

```
{  
  "Snapshot": {  
    "Name": "my-cluster-snapshot-copy",  
    "Status": "creating",  
    "Source": "manual",  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:snapshot/my-cluster-snapshot-copy",  
    "ClusterConfiguration": {
```

```
    "Name": "my-cluster",
    "Description": " ",
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "Port": 6379,
    "ParameterGroupName": "default.memorydb-redis6",
    "SubnetGroupName": "my-sg",
    "VpcId": "vpc-xx2574fc",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "04:30-05:30",
    "NumShards": 2
  }
}
```

詳細については、MemoryDB ユーザーガイドの [「スナップショットのコピー」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CopySnapshot](#)」の「」を参照してください。

create-acl

次のコード例は、create-acl を使用する方法を示しています。

AWS CLI

を作成するには ACL

次のcreate-acl例では、新しいアクセスコントロールリストを作成します。

```
aws memorydb create-acl \  
  --acl-name "new-acl-1" \  
  --user-names "my-user"
```

出力:

```
{  
  "ACL": {  
    "Name": "new-acl-1",  
    "Status": "creating",
```

```
    "UserNames": [  
      "my-user"  
    ],  
    "MinimumEngineVersion": "6.2",  
    "Clusters": [],  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"  
  }  
}
```

詳細については、MemoryDB ユーザーガイドの「[アクセスコントロールリストを使用したユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAcl](#)」の「」を参照してください。

create-cluster

次のコード例は、create-cluster を使用する方法を示しています。

AWS CLI

クラスターを作成するには

次のcreate-cluster例では、新しいクラスターを作成します。

```
aws memorydb create-cluster \  
  --cluster-name my-new-cluster \  
  --node-type db.r6g.large \  
  --acl-name my-acl \  
  --subnet-group my-sg
```

出力:

```
{  
  "Cluster": {  
    "Name": "my-new-cluster",  
    "Status": "creating",  
    "NumberOfShards": 1,  
    "AvailabilityMode": "MultiAZ",  
    "ClusterEndpoint": {  
      "Port": 6379  
    },  
  },  
}
```

```
"NodeType": "db.r6g.large",
"EngineVersion": "6.2",
"EnginePatchVersion": "6.2.6",
"ParameterGroupName": "default.memorydb-redis6",
"ParameterGroupStatus": "in-sync",
"SubnetGroupName": "my-sg",
"TLSEnabled": true,
"ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-new-cluster",
"SnapshotRetentionLimit": 0,
"MaintenanceWindow": "sat:10:00-sat:11:00",
"SnapshotWindow": "07:30-08:30",
"ACLName": "my-acl",
"AutoMinorVersionUpgrade": true
}
}
```

詳細については、MemoryDB ユーザーガイドの「[クラスタの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCluster](#)」の「」を参照してください。

create-parameter-group

次の例は、create-parameter-group を使用する方法を説明しています。

AWS CLI

パラメータグループを作成するには

次のcreate-parameter-group例では、パラメータグループを作成します。

```
aws memorydb create-parameter-group \
  --parameter-group-name myRedis6x \
  --family memorydb_redis6 \
  --description "my-parameter-group"
```

出力:

```
{
  "ParameterGroup": {
    "Name": "myredis6x",
    "Family": "memorydb_redis6",
    "Description": "my-parameter-group",
```

```
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/myredis6x"
  }
}
```

詳細については、MemoryDB ユーザーガイドの「[パラメータグループの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateParameterGroup](#)」の「」を参照してください。

create-snapshot

次のコード例は、create-snapshot を使用する方法を示しています。

AWS CLI

スナップショットを作成するには

次のcreate-snapshot例では、スナップショットを作成します。

```
aws memorydb create-snapshot \  
  --cluster-name my-cluster \  
  --snapshot-name my-cluster-snapshot
```

出力:

```
{  
  "Snapshot": {  
    "Name": "my-cluster-snapshot1",  
    "Status": "creating",  
    "Source": "manual",  
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-snapshot",  
    "ClusterConfiguration": {  
      "Name": "my-cluster",  
      "Description": "",  
      "NodeType": "db.r6g.large",  
      "EngineVersion": "6.2",  
      "MaintenanceWindow": "wed:03:00-wed:04:00",  
      "Port": 6379,  
      "ParameterGroupName": "default.memorydb-redis6",  
      "SubnetGroupName": "my-sg",
```



```
        "VpcId": "vpc-862xxxxc",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "04:30-05:30",
        "NumShards": 2
    }
}
```

詳細については、MemoryDB ユーザーガイドの「[手動スナップショットの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSnapshot](#)」の「」を参照してください。

create-subnet-group

次のコード例は、create-subnet-group を使用する方法を示しています。

AWS CLI

サブネットグループを作成するには

次のcreate-subnet-group例では、サブネットグループを作成します。

```
aws memorydb create-subnet-group \  
  --subnet-group-name mysubnetgroup \  
  --description "my subnet group" \  
  --subnet-ids subnet-5623xxxx
```

出力:

```
{  
  "SubnetGroup": {  
    "Name": "mysubnetgroup",  
    "Description": "my subnet group",  
    "VpcId": "vpc-86257xxx",  
    "Subnets": [  
      {  
        "Identifier": "subnet-5623xxxx",  
        "AvailabilityZone": {  
          "Name": "us-east-1a"  
        }  
      }  
    ]  
  }  
}
```

```

    }
  ],
  "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"
}
}

```

詳細については、MemoryDB ユーザーガイドの「[サブネットグループの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubnetGroup](#)」の「」を参照してください。

create-user

次のコード例は、create-user を使用する方法を示しています。

AWS CLI

ユーザーを作成するには

次のcreate-user例では、新しいユーザーを作成します。

```

aws memorydb create-user \
  --user-name user-name-1 \
  --access-string "~objects:* ~items:* ~public:*" \
  --authentication-mode \
    Passwords="enterapasswordhere",Type=password

```

出力:

```

{
  "User": {
    "Name": "user-name-1",
    "Status": "active",
    "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
    "ACLNames": [],
    "MinimumEngineVersion": "6.2",
    "Authentication": {
      "Type": "password",
      "PasswordCount": 1
    },
    "ARN": "arn:aws:memorydb:us-west-2:491658xxxxxx:user/user-name-1"
  }
}

```

```
}  
}
```

詳細については、MemoryDB ユーザーガイドの「[アクセスコントロールリストを使用したユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUser](#)」の「」を参照してください。

delete-acl

次のコード例は、delete-acl を使用する方法を示しています。

AWS CLI

を削除するには ACL

次のdelete-acl例では、アクセスコントロールリストを削除します。

```
aws memorydb delete-acl \  
  --acl-name "new-acl-1"
```

出力:

```
{  
  "ACL": {  
    "Name": "new-acl-1",  
    "Status": "deleting",  
    "UserNames": [  
      "pat"  
    ],  
    "MinimumEngineVersion": "6.2",  
    "Clusters": [],  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/new-acl-1"  
  }  
}
```

詳細については、MemoryDB ユーザーガイドの「[アクセスコントロールリストを使用したユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAcl](#)」の「」を参照してください。

delete-cluster

次の例は、delete-cluster を使用する方法を説明しています。

AWS CLI

クラスターを削除するには

次のdelete-cluster例では、クラスターを削除します。

```
aws memorydb delete-cluster \  
  --cluster-name my-new-cluster
```

出力:

```
{  
  "Cluster": {  
    "Name": "my-new-cluster",  
    "Status": "deleting",  
    "NumberOfShards": 1,  
    "ClusterEndpoint": {  
      "Address": "clustercfg.my-new-cluster.xxxxx.memorydb.us-  
east-1.amazonaws.com",  
      "Port": 6379  
    },  
    "NodeType": "db.r6g.large",  
    "EngineVersion": "6.2",  
    "EnginePatchVersion": "6.2.6",  
    "ParameterGroupName": "default.memorydb-redis6",  
    "ParameterGroupStatus": "in-sync",  
    "SubnetGroupName": "my-sg",  
    "TLSEnabled": true,  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-new-cluster",  
    "SnapshotRetentionLimit": 0,  
    "MaintenanceWindow": "sat:10:00-sat:11:00",  
    "SnapshotWindow": "07:30-08:30",  
    "AutoMinorVersionUpgrade": true  
  }  
}
```

詳細については、MemoryDB ユーザーガイドの「[クラスターの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCluster](#)」の「」を参照してください。

delete-parameter-group

次のコード例は、delete-parameter-group を使用する方法を示しています。

AWS CLI

パラメータグループを削除するには

次のdelete-parameter-group例では、パラメータグループを削除します。

```
aws memorydb delete-parameter-group \  
  --parameter-group-name myRedis6x
```

出力:

```
{  
  "ParameterGroup": {  
    "Name": "myredis6x",  
    "Family": "memorydb_redis6",  
    "Description": "my-parameter-group",  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/myredis6x"  
  }  
}
```

詳細については、MemoryDB ユーザーガイドの「[パラメータグループの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteParameterGroup](#)」の「」を参照してください。

delete-snapshot

次の例は、delete-snapshot を使用する方法を説明しています。

AWS CLI

スナップショットを削除するには

次のdelete-snapshot例では、スナップショットを削除します。

```
aws memorydb delete-snapshot \  
  --snapshot-name my-cluster-snapshot
```

出力:

```
{
  "Snapshot": {
    "Name": "my-cluster-snapshot",
    "Status": "deleting",
    "Source": "manual",
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:snapshot/my-cluster-snapshot",
    "ClusterConfiguration": {
      "Name": "my-cluster",
      "Description": "",
      "NodeType": "db.r6g.large",
      "EngineVersion": "6.2",
      "MaintenanceWindow": "wed:03:00-wed:04:00",
      "Port": 6379,
      "ParameterGroupName": "default.memorydb-redis6",
      "SubnetGroupName": "my-sg",
      "VpcId": "vpc-862xxxxc",
      "SnapshotRetentionLimit": 0,
      "SnapshotWindow": "04:30-05:30",
      "NumShards": 2
    }
  }
}
```

詳細については、MemoryDB ユーザーガイドの「[スナップショットの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSnapshot](#)」の「」を参照してください。

delete-subnet-group

次のコード例は、delete-subnet-group を使用する方法を示しています。

AWS CLI

サブネットグループを削除するには

次のdelete-subnet-group例では、サブネットを削除します。

```
aws memorydb delete-subnet-group \
```

```
--subnet-group-name mysubnetgroup
```

出力:

```
{
  "SubnetGroup": {
    "Name": "mysubnetgroup",
    "Description": "my subnet group",
    "VpcId": "vpc-86xxxx4fc",
    "Subnets": [
      {
        "Identifier": "subnet-56xxx61b",
        "AvailabilityZone": {
          "Name": "us-east-1a"
        }
      }
    ],
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/mysubnetgroup"
  }
}
```

詳細については、MemoryDB ユーザーガイドの「[サブネットグループの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSubnetGroup](#)」の「」を参照してください。

delete-user

次の例は、delete-user を使用する方法を説明しています。

AWS CLI

ユーザーを削除するには

次のdelete-user例では、ユーザーを削除します。

```
aws memorydb delete-user \  
  --user-name my-user
```

出力:

```
{
  "User": {
    "Name": "my-user",
    "Status": "deleting",
    "AccessString": "on ~app:* resetchannels -@all +@read",
    "ACLNames": [
      "my-acl"
    ],
    "MinimumEngineVersion": "6.2",
    "Authentication": {
      "Type": "password",
      "PasswordCount": 1
    },
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
  }
}
```

詳細については、MemoryDB ユーザーガイドの「[アクセスコントロールリストを使用したユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUser](#)」の「」を参照してください。

describe-acls

次の例は、describe-acls を使用する方法を説明しています。

AWS CLI

のリストを返すには ACLs

次の describe-acls` は、 のリストを返しますACLs。

```
aws memorydb describe-acls
```

出力:

```
{
  "ACLs": [
    {
      "Name": "open-access",
      "Status": "active",
```



```
    "UserNames": [
      "default"
    ],
    "MinimumEngineVersion": "6.2",
    "Clusters": [],
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/open-access"
  },
  {
    "Name": "my-acl",
    "Status": "active",
    "UserNames": [],
    "MinimumEngineVersion": "6.2",
    "Clusters": [
      "my-cluster"
    ],
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/my-acl"
  }
]
}
```

詳細については、MemoryDB ユーザーガイドの [「アクセスコントロールリストを使用したユーザーの認証」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAcls](#)」 の「」を参照してください。

describe-clusters

次の例は、describe-clusters を使用する方法を説明しています。

AWS CLI

クラスターのリストを返すには

次の describe-clusters` は、クラスターのリストを返します。

```
aws memorydb describe-clusters
```

出力:

```
{
  "Clusters": [
    {
```

```
    "Name": "my-cluster",
    "Status": "available",
    "NumberOfShards": 2,
    "ClusterEndpoint": {
      "Address": "clustercfg.my-cluster.llru6f.memorydb.us-
east-1.amazonaws.com",
      "Port": 6379
    },
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "EnginePatchVersion": "6.2.6",
    "ParameterGroupName": "default.memorydb-redis6",
    "ParameterGroupStatus": "in-sync",
    "SecurityGroups": [
      {
        "SecurityGroupId": "sg-0a1434xxxxxc9fae",
        "Status": "active"
      }
    ],
    "SubnetGroupName": "pat-sg",
    "TLSEnabled": true,
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:cluster/my-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "SnapshotWindow": "04:30-05:30",
    "ACLName": "my-acl",
    "AutoMinorVersionUpgrade": true
  }
]
}
```

詳細については、MemoryDB ユーザーガイドの「[クラスタの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClusters](#)」の「」を参照してください。

describe-engine-versions

次の例は、describe-engine-versions を使用する方法を説明しています。

AWS CLI

エンジンバージョンのリストを返すには

次の `describe-engine-versions`` は、エンジンバージョンのリストを返します。

```
aws memorydb describe-engine-versions
```

出力:

```
{
  "EngineVersions": [
    {
      "EngineVersion": "6.2",
      "EnginePatchVersion": "6.2.6",
      "ParameterGroupFamily": "memorydb_redis6"
    }
  ]
}
```

詳細については、MemoryDB ユーザーガイドの [「エンジンのバージョンとアップグレード」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEngineVersions](#)」の「」を参照してください。

describe-events

次の例は、`describe-events` を使用する方法を説明しています。

AWS CLI

イベントのリストを返すには

次の `describe-events`` はイベントのリストを返します。

```
aws memorydb describe-events
```

出力:

```
{
  "Events": [
    {
      "SourceName": "my-cluster",
      "SourceType": "cluster",

```

```

    "Message": "Increase replica count started for replication group my-
cluster on 2022-07-22T14:09:01.440Z",
    "Date": "2022-07-22T07:09:01.443000-07:00"
  },
  {
    "SourceName": "my-user",
    "SourceType": "user",
    "Message": "Create user my-user operation completed.",
    "Date": "2022-07-22T07:00:02.975000-07:00"
  }
]
}

```

詳細については、MemoryDB ユーザーガイドの「[イベントのモニタリング](#)」を参照してください。MemoryDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

describe-parameter-groups

次のコード例は、describe-parameter-groups を使用する方法を示しています。

AWS CLI

パラメータグループのリストを返すには

次の describe-parameter-groups` は、パラメータグループのリストを返します。

```
aws memorydb describe-parameter-groups
```

出力:

```

{
  "ParameterGroups": [
    {
      "Name": "default.memorydb-redis6",
      "Family": "memorydb_redis6",
      "Description": "Default parameter group for memorydb_redis6",
      "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/default.memorydb-redis6"
    }
  ]
}

```

```
}
```

詳細については、MemoryDB ユーザーガイドの「[パラメータグループを使用したエンジンパラメータの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeParameterGroups](#)」の「」を参照してください。

describe-parameters

次のコード例は、describe-parameters を使用する方法を示しています。

AWS CLI

パラメータのリストを返すには

次の describe-parameters` はパラメータのリストを返します。

```
aws memorydb describe-parameters
```

出力:

```
{
  "Parameters": [
    {
      "Name": "acllog-max-len",
      "Value": "128",
      "Description": "The maximum length of the ACL Log",
      "DataType": "integer",
      "AllowedValues": "1-10000",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "activedefrag",
      "Value": "no",
      "Description": "Enabled active memory defragmentation",
      "DataType": "string",
      "AllowedValues": "yes,no",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "active-defrag-cycle-max",
      "Value": "75",
```

```

    "Description": "Maximal effort for defrag in CPU percentage",
    "DataType": "integer",
    "AllowedValues": "1-75",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-cycle-min",
    "Value": "5",
    "Description": "Minimal effort for defrag in CPU percentage",
    "DataType": "integer",
    "AllowedValues": "1-75",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-ignore-bytes",
    "Value": "104857600",
    "Description": "Minimum amount of fragmentation waste to start active
defrag",
    "DataType": "integer",
    "AllowedValues": "1048576-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-max-scan-fields",
    "Value": "1000",
    "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
    "DataType": "integer",
    "AllowedValues": "1-1000000",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-threshold-lower",
    "Value": "10",
    "Description": "Minimum percentage of fragmentation to start active
defrag",
    "DataType": "integer",
    "AllowedValues": "1-100",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-threshold-upper",
    "Value": "100",

```

```
    "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
    "DataType": "integer",
    "AllowedValues": "1-100",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-expire-effort",
    "Value": "1",
    "Description": "The amount of effort that redis uses to expire items in
the active expiration job",
    "DataType": "integer",
    "AllowedValues": "1-10",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "activeresharding",
    "Value": "yes",
    "Description": "Apply rehashing or not",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-normal-hard-limit",
    "Value": "0",
    "Description": "Normal client output buffer hard limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-normal-soft-limit",
    "Value": "0",
    "Description": "Normal client output buffer soft limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-normal-soft-seconds",
    "Value": "0",
    "Description": "Normal client output buffer soft limit in seconds",
    "DataType": "integer",
```

```
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-pubsub-hard-limit",
    "Value": "33554432",
    "Description": "Pubsub client output buffer hard limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-pubsub-soft-limit",
    "Value": "8388608",
    "Description": "Pubsub client output buffer soft limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-pubsub-soft-seconds",
    "Value": "60",
    "Description": "Pubsub client output buffer soft limit in seconds",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "hash-max-ziplist-entries",
    "Value": "512",
    "Description": "The maximum number of hash entries in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "hash-max-ziplist-value",
    "Value": "64",
    "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  }
}
```



```
  },
  {
    "Name": "hll-sparse-max-bytes",
    "Value": "3000",
    "Description": "HyperLogLog sparse representation bytes limit",
    "DataType": "integer",
    "AllowedValues": "1-16000",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-eviction",
    "Value": "no",
    "Description": "Perform an asynchronous delete on evictions",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-expire",
    "Value": "no",
    "Description": "Perform an asynchronous delete on expired keys",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-server-del",
    "Value": "no",
    "Description": "Perform an asynchronous delete on key updates",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-user-del",
    "Value": "no",
    "Description": "Specifies whether the default behavior of DEL command
acts the same as UNLINK",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lfu-decay-time",
```

```
    "Value": "1",
    "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policyd",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lfu-log-factor",
    "Value": "10",
    "Description": "The log factor for incrementing key counter for LFU
eviction policy",
    "DataType": "integer",
    "AllowedValues": "1-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "list-compress-depth",
    "Value": "0",
    "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "maxmemory-policy",
    "Value": "noeviction",
    "Description": "Max memory policy",
    "DataType": "string",
    "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "maxmemory-samples",
    "Value": "3",
    "Description": "Max memory samples",
    "DataType": "integer",
    "AllowedValues": "1-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
```

```
    "Name": "notify-keyspace-events",
    "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
    "DataType": "string",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "set-max-intset-entries",
    "Value": "512",
    "Description": "The limit in the size of the set in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "slowlog-log-slower-than",
    "Value": "10000",
    "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command",
    "DataType": "integer",
    "AllowedValues": "-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "slowlog-max-len",
    "Value": "128",
    "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "stream-node-max-bytes",
    "Value": "4096",
    "Description": "The maximum size of a single node in a stream in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
```

```

    "Name": "stream-node-max-entries",
    "Value": "100",
    "Description": "The maximum number of items a single node in a stream
can contain",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "tcp-keepalive",
    "Value": "300",
    "Description": "If non-zero, send ACKs every given number of seconds",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "timeout",
    "Value": "0",
    "Description": "Close connection if client is idle for a given number of
seconds, or never if 0",
    "DataType": "integer",
    "AllowedValues": "0,20-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "tracking-table-max-keys",
    "Value": "1000000",
    "Description": "The maximum number of keys allowed for the tracking
table for client side caching",
    "DataType": "integer",
    "AllowedValues": "1-1000000000",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "zset-max-ziplist-entries",
    "Value": "128",
    "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {

```

```
        "Name": "zset-max-ziplist-value",
        "Value": "64",
        "Description": "The threshold of biggest sorted set entries in order for
the dataset to be compressed",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    }
]
}
```

詳細については、MemoryDB ユーザーガイドの「[パラメータグループを使用したエンジンパラメータの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeParameters](#)」の「」を参照してください。

describe-snapshots

次の例は、describe-snapshots を使用する方法を説明しています。

AWS CLI

スナップショットのリストを返すには

次の describe-snapshots` はスナップショットのリストを返します。

```
aws memorydb describe-snapshots
```

出力:

```
{
  "Snapshots": [
    {
      "Name": "my-cluster-snapshot",
      "Status": "available",
      "Source": "manual",
      "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx2:snapshot/my-cluster-
snapshot",
      "ClusterConfiguration": {
        "Name": "my-cluster",
        "Description": " ",

```

```
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "Port": 6379,
    "ParameterGroupName": "default.memorydb-redis6",
    "SubnetGroupName": "my-sg",
    "VpcId": "vpc-862574fc",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "04:30-05:30",
    "NumShards": 2
  }
}
```

詳細については、MemoryDB ユーザーガイドの「[スナップショットと復元](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSnapshots](#)」の「」を参照してください。

describe-subnet-groups

次のコード例は、describe-subnet-groups を使用する方法を示しています。

AWS CLI

サブネットグループのリストを返すには

次の describe-subnet-groups` は、サブネットグループのリストを返します。

```
aws memorydb describe-subnet-groups
```

出力

```
{
  "SubnetGroups": [
    {
      "Name": "my-sg",
      "Description": "pat-sg",
      "VpcId": "vpc-86xxx4fc",
      "Subnets": [
        {
```

```

        "Identifier": "subnet-faxx84a6",
        "AvailabilityZone": {
            "Name": "us-east-1b"
        }
    },
    {
        "Identifier": "subnet-56xxf61b",
        "AvailabilityZone": {
            "Name": "us-east-1a"
        }
    }
],
"ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:subnetgroup/my-sg"
}
]
}

```

詳細については、MemoryDB ユーザーガイドの「[サブネットとサブネットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSubnetGroups](#)」の「」を参照してください。

describe-users

次の例は、describe-users を使用する方法を説明しています。

AWS CLI

ユーザーのリストを返すには

次の describe-users` は、ユーザーのリストを返します。

```
aws memorydb describe-users
```

出力

```

{
  "Users": [
    {
      "Name": "default",
      "Status": "active",
      "AccessString": "on ~* &* +@all",

```

```

    "ACLNames": [
      "open-access"
    ],
    "MinimumEngineVersion": "6.0",
    "Authentication": {
      "Type": "no-password"
    },
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/default"
  },
  {
    "Name": "my-user",
    "Status": "active",
    "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",
    "ACLNames": [],
    "MinimumEngineVersion": "6.2",
    "Authentication": {
      "Type": "password",
      "PasswordCount": 2
    },
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"
  }
]
}

```

詳細については、MemoryDB ユーザーガイドの「[アクセスコントロールリストを使用したユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUsers](#)」の「」を参照してください。

failover-shard

次の例は、failover-shard を使用する方法を説明しています。

AWS CLI

シャードをフェイルオーバーするには

次の failover-shard` はシャードで失敗します。

```
aws memorydb failover-shard \
  --cluster-name my-cluster --shard-name 0001
```


出力:

```
{
  "Cluster": {
    "Name": "my-cluster",
    "Status": "available",
    "NumberOfShards": 2,
    "ClusterEndpoint": {
      "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
      "Port": 6379
    },
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "EnginePatchVersion": "6.2.6",
    "ParameterGroupName": "default.memorydb-redis6",
    "ParameterGroupStatus": "in-sync",
    "SecurityGroups": [
      {
        "SecurityGroupId": "sg-0a143xxxx45c9fae",
        "Status": "active"
      }
    ],
    "SubnetGroupName": "my-sg",
    "TLSEnabled": true,
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "SnapshotWindow": "04:30-05:30",
    "AutoMinorVersionUpgrade": true
  }
}
```

詳細については、MemoryDB ユーザーガイドの[MultiAZ によるダウンタイムの最小化](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[FailoverShard](#)」の「」を参照してください。

list-allowed-node-type-updates

次の例は、list-allowed-node-type-updates を使用する方法を説明しています。

AWS CLI

許可されたノードタイプの更新のリストを返すには

次の `list-allowed-node-type-updates` は、使用可能なノードタイプの更新のリストを返します。

```
aws memorydb list-allowed-node-type-updates
```

出力:

```
{
  "Cluster": {
    "Name": "my-cluster",
    "Status": "available",
    "NumberOfShards": 2,
    "ClusterEndpoint": {
      "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-east-1.amazonaws.com",
      "Port": 6379
    },
    "NodeType": "db.r6g.large",
    "EngineVersion": "6.2",
    "EnginePatchVersion": "6.2.6",
    "ParameterGroupName": "default.memorydb-redis6",
    "ParameterGroupStatus": "in-sync",
    "SecurityGroups": [
      {
        "SecurityGroupId": "sg-0a143xxxx45c9fae",
        "Status": "active"
      }
    ],
    "SubnetGroupName": "my-sg",
    "TLSEnabled": true,
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "SnapshotWindow": "04:30-05:30",
    "AutoMinorVersionUpgrade": true
  }
}
```

詳細については、MemoryDB ユーザーガイドの [「スケーリング」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAllowedNodeTypeUpdates](#)」の「」を参照してください。

list-tags

次のコード例は、list-tags を使用する方法を示しています。

AWS CLI

タグのリストを返すには

次のリストタグは、タグのリストを返します。

```
aws memorydb list-tags \  
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster
```

出力:

```
{  
  "TagList": [  
    {  
      "Key": "mytag",  
      "Value": "myvalue"  
    }  
  ]  
}
```

詳細については、MemoryDB ユーザーガイドの「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTags](#)」の「」を参照してください。

reset-parameter-group

次のコード例は、reset-parameter-group を使用する方法を示しています。

AWS CLI

パラメータグループをリセットするには

次の reset-parameter-group` はパラメータグループをリセットします。

```
aws memorydb reset-parameter-group \  
  --parameter-group-name my-parameter-group \  
  --all-parameters
```

出力:

```
{  
  "ParameterGroup": {  
    "Name": "my-parameter-group",  
    "Family": "memorydb_redis6",  
    "Description": "my parameter group",  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/my-parameter-group"  
  }  
}
```

詳細については、MemoryDB ユーザーガイドの「[パラメータグループを使用したエンジンパラメータの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetParameterGroup](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを付けるには

次の tag-resource` は、リソースにタグを追加します。

```
aws memorydb tag-resource \  
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster \  
  --tags Key="mykey",Value="myvalue"
```

出力:

```
{  
  "TagList": [  
    {  
      "Key": "mykey",  
      "Value": "myvalue"  
    }  
  ]  
}
```

```
{
  "Key": "mytag",
  "Value": "myvalue"
},
{
  "Key": "mykey",
  "Value": "myvalue"
}
]
```

詳細については、MemoryDB ユーザーガイドの「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

を更新するには ACL

次の update-acl` ACLは、ユーザーを追加して を更新します。

```
aws memorydb untag-resource \
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \
  --tag-keys mykey
```

出力:

```
{
  "TagList": [
    {
      "Key": "mytag",
      "Value": "myvalue"
    }
  ]
}
```

詳細については、MemoryDB ユーザーガイドの「[リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-cluster

次のコード例は、update-cluster を使用する方法を示しています。

AWS CLI

クラスターを更新するには

次の update-cluster` は、クラスターのパラメータグループを に更新します my-parameter-group。

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --parameter-group-name my-parameter-group
```

出力:

```
{  
  "Cluster": {  
    "Name": "my-cluster",  
    "Status": "available",  
    "NumberOfShards": 2,  
    "AvailabilityMode": "MultiAZ",  
    "ClusterEndpoint": {  
      "Address": "clustercfg.my-cluster.llru6f.memorydb.us-  
east-1.amazonaws.com",  
      "Port": 6379  
    },  
    "NodeType": "db.r6g.large",  
    "EngineVersion": "6.2",  
    "EnginePatchVersion": "6.2.6",  
    "ParameterGroupName": "my-parameter-group",  
    "ParameterGroupStatus": "in-sync",  
    "SecurityGroups": [  
      {  
        "SecurityGroupId": "sg-0a143xxxxxc9fae",  
        "Status": "active"  
      }  
    ],  
  },  
}
```

```
    "SubnetGroupName": "pat-sg",
    "TLSEnabled": true,
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "SnapshotWindow": "04:30-05:30",
    "ACLName": "my-acl",
    "AutoMinorVersionUpgrade": true
  }
}
```

詳細については、MemoryDB ユーザーガイドの「[クラスターの変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCluster](#)」の「」を参照してください。

update-parameter-group

次のコード例は、update-parameter-group を使用する方法を示しています。

AWS CLI

パラメータグループを更新するには

次の update-parameter-group` はパラメータグループを更新します。

```
aws memorydb update-parameter-group \
  --parameter-group-name my-parameter-group \
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

出力:

```
{
  "ParameterGroup": {
    "Name": "my-parameter-group",
    "Family": "memorydb_redis6",
    "Description": "my parameter group",
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/my-parameter-
group"
  }
}
```

詳細については、MemoryDB ユーザーガイドの「[パラメータグループの変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateParameterGroup](#)」の「」を参照してください。

update-subnet-group

次の例は、update-subnet-group を使用する方法を説明しています。

AWS CLI

サブネットグループを更新するには

次の update-subnet-group` は、サブネットグループのサブネット ID を更新します。

```
aws memorydb update-subnet-group \  
  --subnet-group-name my-sg \  
  --subnet-ids subnet-01f29d458f3xxxxx
```

出力:

```
{  
  "SubnetGroup": {  
    "Name": "my-sg-1",  
    "Description": "my-sg",  
    "VpcId": "vpc-09d2cfc01xxxxxxxx",  
    "Subnets": [  
      {  
        "Identifier": "subnet-01f29d458f3xxxxx",  
        "AvailabilityZone": {  
          "Name": "us-east-1a"  
        }  
      }  
    ],  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/my-sg"  
  }  
}
```

詳細については、MemoryDB ユーザーガイドの「[サブネットとサブネットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSubnetGroup](#)」の「」を参照してください。

update-user

次の例は、update-user を使用する方法を説明しています。

AWS CLI

ユーザーを更新するには

以下は、ユーザーのアクセス文字列update-userを変更します。

```
aws memorydb update-user \  
  --user-name my-user \  
  --access-string "off ~objects:* ~items:* ~public:* resetchannels -@all"
```

出力:

```
{  
  "User": {  
    "Name": "my-user",  
    "Status": "modifying",  
    "AccessString": "off ~objects:* ~items:* ~public:* resetchannels -@all",  
    "ACLNames": [  
      "myt-acl"  
    ],  
    "MinimumEngineVersion": "6.2",  
    "Authentication": {  
      "Type": "password",  
      "PasswordCount": 2  
    },  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:user/my-user"  
  }  
}
```

詳細については、MemoryDB ユーザーガイドの「[アクセスコントロールリストを使用したユーザーの認証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUser](#)」の「」を参照してください。

を使用した Amazon MSK の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますMSK。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-cluster

次のコード例は、create-cluster を使用する方法を示しています。

AWS CLI

Amazon MSK クラスターを作成するには

次のcreate-cluster例では、という名前のMSKクラスターを3つのブローカーノードMessagingClusterで作成します。という名前のJSONファイルは、Amazon がブローカーノードを分散MSKする3つのサブネットbroker-node-group-info.jsonを指定します。この例ではモニタリングレベルを指定しないため、クラスターはDEFAULTレベルを取得します。

```
aws kafka create-cluster \  
  --cluster-name "MessagingCluster" \  
  --broker-node-group-info file://broker-node-group-info.json \  
  --kafka-version "2.2.1" \  
  --number-of-broker-nodes 3
```

broker-node-group-info.json の内容:

```
{
```

```
"InstanceType": "kafka.m5.xlarge",
"BrokerAZDistribution": "DEFAULT",
"ClientSubnets": [
    "subnet-0123456789111abcd",
    "subnet-0123456789222abcd",
    "subnet-0123456789333abcd"
]
```

出力:

```
{
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
  "ClusterName": "MessagingCluster",
  "State": "CREATING"
}
```

詳細については、「Amazon Managed Streaming for Apache Kafka」の「Amazon [MSKクラスターの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCluster](#)」の「」を参照してください。

create-configuration

次のコード例は、create-configuration を使用する方法を示しています。

AWS CLI

カスタム Amazon MSK設定を作成するには

次のcreate-configuration例では、入力ファイルで指定されたサーバープロパティを使用してカスタムMSK設定を作成します。

```
aws kafka create-configuration \
  --name "CustomConfiguration" \
  --description "Topic autcreation enabled; Apache ZooKeeper timeout 2000 ms; Log
rolling 60480000 ms." \
  --kafka-versions "2.2.1" \
  --server-properties file://configuration.txt
```

configuration.txt の内容:

```
auto.create.topics.enable = true
zookeeper.connection.timeout.ms = 2000
log.roll.ms = 604800000
```

このコマンドでは何も出力されません。出力:

```
{
  "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
  "CreationTime": "2019-10-09T15:26:05.548Z",
  "LatestRevision":
    {
      "CreationTime": "2019-10-09T15:26:05.548Z",
      "Description": "Topic autocreation enabled; Apache ZooKeeper timeout
2000 ms; Log rolling 604800000 ms.",
      "Revision": 1
    },
  "Name": "CustomConfiguration"
}
```

詳細については、「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「Amazon [MSK設定オペレーション](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConfiguration](#)」の「」を参照してください。

describe-cluster

次のコード例は、describe-cluster を使用する方法を示しています。

AWS CLI

クラスターを記述するには

次のdescribe-cluster例では、Amazon MSKクラスターについて説明します。

```
aws kafka describe-cluster \
  --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

出力:

```
{
  "ClusterInfo": {
    "BrokerNodeGroupInfo": {
      "BrokerAZDistribution": "DEFAULT",
      "ClientSubnets": [
        "subnet-cbfff283",
        "subnet-6746046b"
      ],
      "InstanceType": "kafka.m5.large",
      "SecurityGroups": [
        "sg-f839b688"
      ],
      "StorageInfo": {
        "EbsStorageInfo": {
          "VolumeSize": 100
        }
      }
    },
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
    "ClusterName": "demo-cluster-1",
    "CreationTime": "2020-07-09T02:31:36.223000+00:00",
    "CurrentBrokerSoftwareInfo": {
      "KafkaVersion": "2.2.1"
    },
    "CurrentVersion": "K3AEGXETSR30VB",
    "EncryptionInfo": {
      "EncryptionAtRest": {
        "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a7ca56d5-0768-4b64-a670-339a9fbef81c"
      },
      "EncryptionInTransit": {
        "ClientBroker": "TLS_PLAINTEXT",
        "InCluster": true
      }
    },
    "EnhancedMonitoring": "DEFAULT",
    "OpenMonitoring": {
      "Prometheus": {
        "JmxExporter": {
          "EnabledInBroker": false
        }
      }
    }
  }
}
```

```
        "NodeExporter": {
            "EnabledInBroker": false
        }
    },
    "NumberOfBrokerNodes": 2,
    "State": "ACTIVE",
    "Tags": {},
    "ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
}
}
```

詳細については、[「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「Amazon MSK クラスターの一覧表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeCluster](#)」の「」を参照してください。

get-bootstrap-brokers

次のコード例は、get-bootstrap-brokers を使用する方法を示しています。

AWS CLI

ブートストラップブローカーを取得するには

次のget-bootstrap-brokers例では、Amazon MSK クラスターのブートストラップブローカー情報を取得します。

```
aws kafka get-bootstrap-brokers \
  --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

出力:

```
{
  "BootstrapBrokerString": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:9092,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:9092",
```

```
"BootstrapBrokerStringTls": "b-1.demo-cluster-1.xuy0sb.c5.kafka.us-east-1.amazonaws.com:9094,b-2.demo-cluster-1.xuy0sb.c5.kafka.us-east-1.amazonaws.com:9094"
}
```

詳細については、[「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「ブートストラップブローカーの取得」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBootstrapBrokers](#)」の「」を参照してください。

list-clusters

次のコード例は、list-clusters を使用する方法を示しています。

AWS CLI

使用可能なクラスターを一覧表示するには

次のlist-clusters例では、AWS アカウントの Amazon MSKクラスターを一覧表示します。

```
aws kafka list-clusters
```

出力:

```
{
  "ClusterInfoList": [
    {
      "BrokerNodeGroupInfo": {
        "BrokerAZDistribution": "DEFAULT",
        "ClientSubnets": [
          "subnet-cbfff283",
          "subnet-6746046b"
        ],
        "InstanceType": "kafka.m5.large",
        "SecurityGroups": [
          "sg-f839b688"
        ],
        "StorageInfo": {
          "EbsStorageInfo": {
            "VolumeSize": 100
          }
        }
      }
    }
  ]
}
```

```
    },
    "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
    "ClusterName": "demo-cluster-1",
    "CreationTime": "2020-07-09T02:31:36.223000+00:00",
    "CurrentBrokerSoftwareInfo": {
      "KafkaVersion": "2.2.1"
    },
    "CurrentVersion": "K3AEGXETSR30VB",
    "EncryptionInfo": {
      "EncryptionAtRest": {
        "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
      },
      "EncryptionInTransit": {
        "ClientBroker": "TLS_PLAINTEXT",
        "InCluster": true
      }
    },
    "EnhancedMonitoring": "DEFAULT",
    "OpenMonitoring": {
      "Prometheus": {
        "JmxExporter": {
          "EnabledInBroker": false
        },
        "NodeExporter": {
          "EnabledInBroker": false
        }
      }
    },
    "NumberOfBrokerNodes": 2,
    "State": "ACTIVE",
    "Tags": {},
    "ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
  }
]
}
```

詳細については、[「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「Amazon MSK クラスターの一覧表示」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListClusters](#)」の「」を参照してください。

update-broker-storage

次のコード例は、update-broker-storage を使用する方法を示しています。

AWS CLI

ブローカーのEBSストレージを更新するには

次のupdate-broker-storage例では、クラスター内のすべてのブローカーのEBSストレージ量を更新します。Amazon は、各ブローカーのターゲットストレージ量を、この例で指定された量MSKに設定します。クラスターを記述するか、すべてのクラスターを一覧表示することで、クラスターの最新バージョンを取得できます。

```
aws kafka update-broker-storage \  
  --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \  
  --current-version "K21V3IB1VIZYYH" \  
  --target-broker-efs-volume-info "KafkaBrokerNodeId=ALL,VolumeSizeGB=1100"
```

出力は、このupdate-broker-storageオペレーションARNの を返します。このオペレーションが完了したかどうかを判断するには、describe-cluster-operation コマンドを使用してこれを入力ARNとして使用します。

```
{  
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",  
  "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-  
operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-  
bcde-33333EXAMPLE"  
}
```

詳細については、[「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「ブローカーのEBSストレージを更新する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateBrokerStorage](#)」の「」を参照してください。

update-cluster-configuration

次のコード例は、update-cluster-configuration を使用する方法を示しています。

AWS CLI

Amazon MSK クラスターの設定を更新するには

次の update-cluster-configuration 例では、指定された既存の MSK クラスターの設定を更新します。カスタム MSK 設定を使用します。

```
aws kafka update-cluster-configuration \  
  --cluster-arn "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2" \  
  --configuration-info file://configuration-info.json \  
  --current-version "K21V3IB1VIZYYH"
```

configuration-info.json の内容:

```
{  
  "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",  
  "Revision": 1  
}
```

出力は、この update-cluster-configuration オペレーション ARN の を返します。このオペレーションが完了したかどうかを判断するには、describe-cluster-operation コマンドを使用してこれを入力 ARN として使用します。

```
{  
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",  
  "ClusterOperationArn": "arn:aws:kafka:us-west-2:123456789012:cluster-  
operation/V123450123/a1b2c3d4-1234-abcd-cdef-22222EXAMPLE-2/a1b2c3d4-abcd-1234-  
bcde-33333EXAMPLE"  
}
```

詳細については、「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「Amazon [MSK クラスターの設定を更新する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateClusterConfiguration](#)」の「」を参照してください。

を使用した Network Manager の例 AWS CLI

次のコード例は、Network Manager AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-customer-gateway

次のコード例は、associate-customer-gateway を使用する方法を示しています。

AWS CLI

カスタマーゲートウェイを関連付けるには

次のassociate-customer-gateway例では、指定されたグローバルネットワークcgw-11223344556677889内のカスタマーゲートウェイをデバイスに関連付けますdevice-07f6fd08867abc123。

```
aws networkmanager associate-customer-gateway \  
  --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/  
cgw-11223344556677889 \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --region us-west-2
```

出力:

```
{  
  "CustomerGatewayAssociation": {
```

```
    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/cgw-11223344556677889",
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "State": "PENDING"
  }
}
```

詳細については、「Transit [Gateway Network Manager Guide](#)」の「[Customer Gateway Associations](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスAssociateCustomerGateway](#)」の「」を参照してください。

associate-link

次の例は、associate-link を使用する方法を説明しています。

AWS CLI

リンクを関連付けるには

次のassociate-link例では、リンクをデバイス link-11112222aaaabbbb1に関連付けますdevice-07f6fd08867abc123。リンクとデバイスは、指定されたグローバルネットワークにあります。

```
aws networkmanager associate-link \
  --global-network-id global-network-01231231231231231 \
  --device-id device-07f6fd08867abc123 \
  --link-id link-11112222aaaabbbb1 \
  --region us-west-2
```

出力:

```
{
  "LinkAssociation": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "LinkId": "link-11112222aaaabbbb1",
    "LinkAssociationState": "PENDING"
  }
}
```

```
}
```

詳細については、「Transit Gateway Network Manager ガイド」の [「デバイスとリンクの関連付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateLink](#)」の「」を参照してください。

create-core-network

次の例は、create-core-network を使用する方法を説明しています。

AWS CLI

コアネットワークを作成するには

次のcreate-core-network例では、AWS Cloud WAN グローバルネットワーク内でオプションの説明とタグを使用してコアネットワークを作成します。

```
aws networkmanager create-core-network \  
  --global-network-id global-network-cdef-EXAMPLE22222 \  
  --description "Main headquarters location" \  
  --tags Key=Name,Value="New York City office"
```

出力:

```
{  
  "CoreNetwork": {  
    "GlobalNetworkId": "global-network-cdef-EXAMPLE22222",  
    "CoreNetworkId": "core-network-cdef-EXAMPLE33333",  
    "CoreNetworkArn": "arn:aws:networkmanager::987654321012:core-network/core-network-cdef-EXAMPLE33333",  
    "Description": "Main headquarters location",  
    "CreatedAt": "2022-01-10T19:53:59+00:00",  
    "State": "AVAILABLE",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "New York City office"  
      }  
    ]  
  }  
}
```

```
}
```

詳細については、「クラウドユーザーガイド」の「[グローバルネットワークとコアネットワーク](#)」を参照してください。AWS WAN

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCoreNetwork](#)」の「」を参照してください。

create-device

次のコード例は、create-device を使用する方法を示しています。

AWS CLI

デバイスを作成するには

次のcreate-device例では、指定されたグローバルネットワークにデバイスを作成します。デバイスの詳細には、説明、タイプ、ベンダー、モデル、シリアル番号が含まれます。

```
aws networkmanager create-device
--global-network-id global-network-01231231231231231 \
--description "New York office device" \
--type "office device" \
--vendor "anycompany" \
--model "abcabc" \
--serial-number "1234" \
--region us-west-2
```

出力:

```
{
  "Device": {
    "DeviceId": "device-07f6fd08867abc123",
    "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
    "GlobalNetworkId": "global-network-01231231231231231",
    "Description": "New York office device",
    "Type": "office device",
    "Vendor": "anycompany",
    "Model": "abcabc",
    "SerialNumber": "1234",
    "CreatedAt": 1575554005.0,
```

```
    "State": "PENDING"
  }
}
```

詳細については、「Transit Gateway Network Manager ガイド」の「[デバイスの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDevice](#)」の「」を参照してください。

create-global-network

次のコード例は、create-global-network を使用する方法を示しています。

AWS CLI

グローバルネットワークを作成するには

次のcreate-global-network例では、新しいグローバルネットワークを作成します。作成時の初期状態は PENDING です。

```
aws networkmanager create-global-network
```

出力:

```
{
  "GlobalNetwork": {
    "GlobalNetworkId": "global-network-00a77fc0f722dae74",
    "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/global-network-00a77fc0f722dae74",
    "CreatedAt": "2022-03-14T20:31:56+00:00",
    "State": "PENDING"
  }
}
```

- API 詳細については、「コマンドリファレンス[CreateGlobalNetwork](#)」の「」を参照してください。AWS CLI

create-link

次の例は、create-link を使用する方法を説明しています。

AWS CLI

リンクを作成するには

次のcreate-link例では、指定されたグローバルネットワークにリンクを作成します。リンクには、リンクタイプ、帯域幅、プロバイダーの説明と詳細が含まれます。サイト ID は、リンクが関連付けられているサイトを示します。

```
aws networkmanager create-link \  
  --global-network-id global-network-01231231231231231 \  
  --description "VPN Link" \  
  --type "broadband" \  
  --bandwidth UploadSpeed=10,DownloadSpeed=20 \  
  --provider "AnyCompany" \  
  --site-id site-444555aaabbb11223 \  
  --region us-west-2
```

出力:

```
{  
  "Link": {  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkArn": "arn:aws:networkmanager::123456789012:link/global-  
network-01231231231231231/link-11112222aaaabbbb1",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "SiteId": "site-444555aaabbb11223",  
    "Description": "VPN Link",  
    "Type": "broadband",  
    "Bandwidth": {  
      "UploadSpeed": 10,  
      "DownloadSpeed": 20  
    },  
    "Provider": "AnyCompany",  
    "CreatedAt": 1575555811.0,  
    "State": "PENDING"  
  }  
}
```

詳細については、「[Transit Gateway Network Manager ガイド](#)」の「[リンクの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateLink](#)」の「」を参照してください。

create-site

次のコード例は、create-site を使用する方法を示しています。

AWS CLI

サイトを作成するには

次のcreate-site例では、指定されたグローバルネットワークにサイトを作成します。サイトの詳細には、説明と位置情報が含まれます。

```
aws networkmanager create-site \  
  --global-network-id global-network-01231231231231231 \  
  --description "New York head office" \  
  --location Latitude=40.7128,Longitude=-74.0060 \  
  --region us-west-2
```

出力:

```
{  
  "Site": {  
    "SiteId": "site-444555aaabbb11223",  
    "SiteArn": "arn:aws:networkmanager::123456789012:site/global-  
network-01231231231231231/site-444555aaabbb11223",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "Description": "New York head office",  
    "Location": {  
      "Latitude": "40.7128",  
      "Longitude": "-74.0060"  
    },  
    "CreatedAt": 1575554300.0,  
    "State": "PENDING"  
  }  
}
```

詳細については、「Transit Gateway Network Manager ガイド」の「[サイトの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSite](#)」の「」を参照してください。

create-vpc-attachment

次のコード例は、create-vpc-attachment を使用する方法を示しています。

AWS CLI

VPCアタッチメントを作成するには

次のcreate-vpc-attachment例では、コアネットワークで IPv6 をサポートするVPCアタッチメントを作成します。

```
aws networkmanager create-vpc-attachment \  
  --core-network-id core-network-0fab62fe438d94db6 \  
  --vpc-arn arn:aws:ec2:us-east-1:987654321012:vpc/vpc-09f37f69e2786eeb8 \  
  --subnet-arns arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7 \  
  --Ipv6Support=true
```

出力:

```
{  
  "VpcAttachment": {  
    "Attachment": {  
      "CoreNetworkId": "core-network-0fab62fe438d94db6",  
      "AttachmentId": "attachment-05e1da6eba87a06e6",  
      "OwnerAccountId": "987654321012",  
      "AttachmentType": "VPC",  
      "State": "CREATING",  
      "EdgeLocation": "us-east-1",  
      "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/  
vpc-09f37f69e2786eeb8",  
      "Tags": [],  
      "CreatedAt": "2022-03-10T20:59:14+00:00",  
      "UpdatedAt": "2022-03-10T20:59:14+00:00"  
    },  
    "SubnetArns": [  
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-04ca4e010857e7bb7"  
    ],  
    "Options": {  
      "Ipv6Support": true  
    }  
  }  
}
```

詳細については、「クラウドユーザーガイド」の「[アタッチメントの作成](#)」を参照してください。
WAN

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVpcAttachment](#)」の「」を参照してください。

delete-attachment

次の例は、delete-attachment を使用する方法を説明しています。

AWS CLI

アタッチメントを削除するには

次のdelete-attachment例では、Connect アタッチメントを削除します。

```
aws networkmanager delete-attachment \  
--attachment-id attachment-01feddaeae26ab68c
```

出力:

```
{  
  "Attachment": {  
    "CoreNetworkId": "core-network-0f4b0a9d5ee7761d1",  
    "AttachmentId": "attachment-01feddaeae26ab68c",  
    "OwnerAccountId": "987654321012",  
    "AttachmentType": "CONNECT",  
    "State": "DELETING",  
    "EdgeLocation": "us-east-1",  
    "ResourceArn": "arn:aws:networkmanager::987654321012:attachment/  
attachment-02c3964448fedf5aa",  
    "CreatedAt": "2022-03-15T19:18:41+00:00",  
    "UpdatedAt": "2022-03-15T19:28:59+00:00"  
  }  
}
```

詳細については、「クラウドユーザーガイド」の「[添付ファイルの削除](#)」を参照してください。
WAN

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAttachment](#)」の「」を参照してください。

delete-bucket-analytics-configuration

次のコード例は、delete-bucket-analytics-configuration を使用する方法を示しています。

AWS CLI

バケットの分析設定を削除するには

次の delete-bucket-analytics-configuration の例では、指定されたバケットと ID の分析設定を削除します。

```
aws s3api delete-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteBucketAnalyticsConfiguration](#)」の「」を参照してください。

delete-bucket-metrics-configuration

次の例は、delete-bucket-metrics-configuration を使用する方法を説明しています。

AWS CLI

バケットのメトリクス設定を削除するには

次の delete-bucket-metrics-configuration の例では、指定したバケットと ID のメトリクス設定を削除します。

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteBucketMetricsConfiguration](#)」の「」を参照してください。

delete-core-network

次のコード例は、delete-core-network を使用する方法を示しています。

AWS CLI

コアネットワークを削除するには

次のdelete-core-network例では、クラウドWANグローバルネットワークからコアネットワークを削除します。

```
aws networkmanager delete-core-network \  
  --core-network-id core-network-0fab62fe438d94db6
```

出力:

```
{  
  "CoreNetwork": {  
    "GlobalNetworkId": "global-network-0d59060f16a73bc41",  
    "CoreNetworkId": "core-network-0fab62fe438d94db6",  
    "Description": "Main headquarters location",  
    "CreatedAt": "2021-12-09T18:31:11+00:00",  
    "State": "DELETING",  
    "Segments": [  
      {  
        "Name": "dev",  
        "EdgeLocations": [  
          "us-east-1"  
        ],  
        "SharedSegments": []  
      }  
    ],  
    "Edges": [  
      {  
        "EdgeLocation": "us-east-1",  
        "Asn": 64512,  
        "InsideCidrBlocks": []  
      }  
    ]  
  }  
}
```

詳細については、「クラウドユーザーガイド」の「[コアネットワーク](#)」を参照してください。

WAN

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCoreNetwork](#)」の「」を参照してください。

delete-device

次のコード例は、delete-device を使用する方法を示しています。

AWS CLI

デバイスを削除するには

次のdelete-device例では、指定されたグローバルネットワークから指定されたデバイスを削除します。

```
aws networkmanager delete-device \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --region us-west-2
```

出力:

```
{  
  "Device": {  
    "DeviceId": "device-07f6fd08867abc123",  
    "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-  
network-01231231231231231/device-07f6fd08867abc123",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "Description": "New York office device",  
    "Type": "office device",  
    "Vendor": "anycompany",  
    "Model": "abcabc",  
    "SerialNumber": "1234",  
    "SiteId": "site-444555aaabbb11223",  
    "CreatedAt": 1575554005.0,  
    "State": "DELETING"  
  }  
}
```

詳細については、「Transit Gateway Network Manager ガイド」の「[デバイスの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDevice](#)」の「」を参照してください。

delete-global-network

次のコード例は、delete-global-network を使用する方法を示しています。

AWS CLI

グローバルネットワークを削除するには

次のdelete-global-network例では、グローバルネットワークを削除します。

```
aws networkmanager delete-global-network \  
--global-network-id global-network-052bedddccb193b6b
```

出力:

```
{  
  "GlobalNetwork": {  
    "GlobalNetworkId": "global-network-052bedddccb193b6b",  
    "GlobalNetworkArn": "arn:aws:networkmanager::987654321012:global-network/  
global-network-052bedddccb193b6b",  
    "CreatedAt": "2021-12-09T18:19:12+00:00",  
    "State": "DELETING"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGlobalNetwork](#)」の「」を参照してください。

delete-link

次のコード例は、delete-link を使用する方法を示しています。

AWS CLI

リンクを削除するには

次のdelete-link例では、指定されたグローバルネットワークから指定されたリンクを削除します。

```
aws networkmanager delete-link \  
  --global-network-id global-network-01231231231231231 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

出力:

```
{  
  "Link": {  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkArn": "arn:aws:networkmanager::123456789012:link/global-  
network-01231231231231231/link-11112222aaaabbbb1",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "SiteId": "site-444555aaaabbb11223",  
    "Description": "VPN Link",  
    "Type": "broadband",  
    "Bandwidth": {  
      "UploadSpeed": 20,  
      "DownloadSpeed": 20  
    },  
    "Provider": "AnyCompany",  
    "CreatedAt": 1575555811.0,  
    "State": "DELETING"  
  }  
}
```

詳細については、「[Transit Gateway Network Manager ガイド](#)」の「[リンクの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLink](#)」の「」を参照してください。

delete-public-access-block

次の例は、delete-public-access-block を使用する方法を説明しています。

AWS CLI

バケットのブロックパブリックアクセス設定を削除するには

次の `delete-public-access-block` の例では、指定したバケットのブロックパブリックアクセス設定を削除します。

```
aws s3api delete-public-access-block \  
  --bucket my-bucket
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeletePublicAccessBlock](#)」の「」を参照してください。

delete-site

次のコード例は、`delete-site` を使用する方法を示しています。

AWS CLI

サイトを削除するには

次の `delete-site` 例では、指定されたグローバルネットワーク内の指定されたサイト (`site-444555aaabbb11223`) を削除します。

```
aws networkmanager delete-site \  
  --global-network-id global-network-01231231231231231 \  
  --site-id site-444555aaabbb11223 \  
  --region us-west-2
```

出力:

```
{  
  "Site": {  
    "SiteId": "site-444555aaabbb11223",  
    "SiteArn": "arn:aws:networkmanager::123456789012:site/global-  
network-01231231231231231/site-444555aaabbb11223",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "Description": "New York head office",  
    "Location": {  
      "Latitude": "40.7128",  
      "Longitude": "-74.0060"  
    },  
    "CreatedAt": 1575554300.0,  
    "State": "DELETING"  
  }  
}
```

```
}  
}
```

詳細については、「Transit Gateway Network Manager ガイド」の[「サイトの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSite](#)」の「」を参照してください。

deregister-transit-gateway

次の例は、deregister-transit-gateway を使用する方法を説明しています。

AWS CLI

グローバルネットワークからトランジットゲートウェイの登録を解除するには

次のderegister-transit-gateway例では、指定されたグローバルネットワークから指定されたトランジットゲートウェイを登録解除します。

```
aws networkmanager deregister-transit-gateway \  
  --global-network-id global-network-01231231231231231 \  
  --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-123abc05e04123abc \  
  --region us-west-2
```

出力:

```
{  
  "TransitGatewayRegistration": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-123abc05e04123abc",  
    "State": {  
      "Code": "DELETING"  
    }  
  }  
}
```

詳細については、[「Transit Gateway Network Manager ガイド」の「Transit Gateway 登録」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterTransitGateway](#)」の「」を参照してください。

describe-global-networks

次のコード例は、describe-global-networks を使用する方法を示しています。

AWS CLI

グローバルネットワークを記述するには

次のdescribe-global-networks例では、アカウント内のすべてのグローバルネットワークについて説明します。

```
aws networkmanager describe-global-networks \  
  --region us-west-2
```

出力:

```
{  
  "GlobalNetworks": [  
    {  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-  
network/global-network-01231231231231231",  
      "Description": "Company 1 global network",  
      "CreatedAt": 1575553525.0,  
      "State": "AVAILABLE"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGlobalNetworks](#)」の「」を参照してください。

disassociate-customer-gateway

次の例は、disassociate-customer-gateway を使用する方法を説明しています。

AWS CLI

カスタマーゲートウェイの関連付けを解除するには

次のdisassociate-customer-gateway例では、指定されたカスタマーゲートウェイ (cgw-11223344556677889) と指定されたグローバルネットワークの関連付けを解除します。

```
aws networkmanager disassociate-customer-gateway \  
  --global-network-id global-network-01231231231231 \  
  --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/  
cgw-11223344556677889 \  
  --region us-west-2
```

出力:

```
{  
  "CustomerGatewayAssociation": {  
    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/  
cgw-11223344556677889",  
    "GlobalNetworkId": "global-network-01231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "State": "DELETING"  
  }  
}
```

詳細については、「Transit [Gateway Network Manager Guide](#)」の「[Customer Gateway Associations](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateCustomerGateway](#)」の「」を参照してください。

disassociate-link

次の例は、disassociate-link を使用する方法を説明しています。

AWS CLI

リンクの関連付けを解除するには

次のdisassociate-link例では、指定されたグローバルネットワークdevice-07f6fd08867abc123内のデバイスから指定されたリンクの関連付けを解除します。

```
aws networkmanager disassociate-link \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

出力:

```
{  
  "LinkAssociation": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkAssociationState": "DELETING"  
  }  
}
```

詳細については、「Transit Gateway Network Manager ガイド」の [「デバイスとリンクの関連付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateLink](#)」の「」を参照してください。

get-bucket-analytics-configuration

次のコード例は、get-bucket-analytics-configuration を使用する方法を示しています。

AWS CLI

特定の ID を持つバケットの分析設定を取得するには

次の get-bucket-analytics-configuration の例では、指定されたバケットと ID の分析設定を表示します。

```
aws s3api get-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

出力:

```
{
```

```
"AnalyticsConfiguration": {
  "StorageClassAnalysis": {},
  "Id": "1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketAnalyticsConfiguration](#)」の「」を参照してください。

get-bucket-metrics-configuration

次のコード例は、get-bucket-metrics-configuration を使用する方法を示しています。

AWS CLI

特定の ID を持つバケットのメトリクス設定を取得するには

次の get-bucket-metrics-configuration の例では、指定したバケットと ID のメトリクス設定を表示します。

```
aws s3api get-bucket-metrics-configuration \
  --bucket my-bucket \
  --id 123
```

出力:

```
{
  "MetricsConfiguration": {
    "Filter": {
      "Prefix": "logs"
    },
    "Id": "123"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketMetricsConfiguration](#)」の「」を参照してください。

get-customer-gateway-associations

次の例は、get-customer-gateway-associations を使用する方法を説明しています。

AWS CLI

カスタマーゲートウェイの関連付けを取得するには

次のget-customer-gateway-associations例では、指定されたグローバルネットワークのカスタマーゲートウェイの関連付けを取得します。

```
aws networkmanager get-customer-gateway-associations \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

出力:

```
{  
  "CustomerGatewayAssociations": [  
    {  
      "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-  
gateway/cgw-11223344556677889",  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "DeviceId": "device-07f6fd08867abc123",  
      "State": "AVAILABLE"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCustomerGatewayAssociations](#)」の「」を参照してください。

get-devices

次のコード例は、get-devices を使用する方法を示しています。

AWS CLI

デバイスを取得するには

次のget-devices例では、指定されたグローバルネットワーク内のデバイスを取得します。

```
aws networkmanager get-devices \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

出力:

```
{
  "Devices": [
    {
      "DeviceId": "device-07f6fd08867abc123",
      "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123",
      "GlobalNetworkId": "global-network-01231231231231231",
      "Description": "NY office device",
      "Type": "office device",
      "Vendor": "anycompany",
      "Model": "abcabc",
      "SerialNumber": "1234",
      "CreatedAt": 1575554005.0,
      "State": "AVAILABLE"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDevices](#)」の「」を参照してください。

get-link-associations

次のコード例は、get-link-associations を使用する方法を示しています。

AWS CLI

リンクの関連付けを取得するには

次のget-link-associations例では、指定されたグローバルネットワーク内のリンク関連付けを取得します。

```
aws networkmanager get-link-associations \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2
```

出力:

```
{
```



```
"LinkAssociations": [  
  {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkAssociationState": "AVAILABLE"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLinkAssociations](#)」の「」を参照してください。

get-links

次のコード例は、get-links を使用する方法を示しています。

AWS CLI

リンクを取得するには

次のget-links例では、指定されたグローバルネットワーク内のリンクを取得します。

```
aws networkmanager get-links \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

出力:

```
{  
  "Links": [  
    {  
      "LinkId": "link-11112222aaaabbbb1",  
      "LinkArn": "arn:aws:networkmanager::123456789012:link/global-  
network-01231231231231231/link-11112222aaaabbbb1",  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "SiteId": "site-444555aaaabbb11223",  
      "Description": "VPN Link",  
      "Type": "broadband",  
      "Bandwidth": {  
        "UploadSpeed": 10,  
        "DownloadSpeed": 20  
      }  
    }  
  ]  
}
```

```
    },  
    "Provider": "AnyCompany",  
    "CreatedAt": 1575555811.0,  
    "State": "AVAILABLE"  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLinks](#)」の「」を参照してください。

get-object-retention

次のコード例は、`get-object-retention` を使用する方法を示しています。

AWS CLI

オブジェクトのオブジェクト保持設定を取得するには

次の `get-object-retention` 例では、指定されたオブジェクトの保持設定を取得します。

```
aws s3api get-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

出力:

```
{  
  "Retention": {  
    "Mode": "GOVERNANCE",  
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetObjectRetention](#)」の「」を参照してください。

get-public-access-block

次の例は、`get-public-access-block` を使用する方法を説明しています。

AWS CLI

バケットのブロックパブリックアクセス設定を指定または削除するには

次の `get-public-access-block` の例では、指定したバケットのブロックパブリックアクセス設定を表示します。

```
aws s3api get-public-access-block --bucket my-bucket
```

出力:

```
{
  "PublicAccessBlockConfiguration": {
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "BlockPublicAcls": true,
    "RestrictPublicBuckets": true
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPublicAccessBlock](#)」の「」を参照してください。

get-sites

次のコード例は、`get-sites` を使用する方法を示しています。

AWS CLI

サイトを取得するには

次の `get-sites` 例では、指定されたグローバルネットワーク内のサイトを取得します。

```
aws networkmanager get-sites \
  --global-network-id global-network-01231231231231 \
  --region us-west-2
```

出力:

```
{
  "Sites": [
    {
```

```

    "SiteId": "site-444555aaabbb11223",
    "SiteArn": "arn:aws:networkmanager::123456789012:site/global-
network-01231231231231231/site-444555aaabbb11223",
    "GlobalNetworkId": "global-network-01231231231231231",
    "Description": "NY head office",
    "Location": {
      "Latitude": "40.7128",
      "Longitude": "-74.0060"
    },
    "CreatedAt": 1575554528.0,
    "State": "AVAILABLE"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSites](#)」の「」を参照してください。

get-transit-gateway-registrations

次の例は、get-transit-gateway-registrations を使用する方法を説明しています。

AWS CLI

Transit Gateway の登録を取得するには

次のget-transit-gateway-registrations例では、指定されたグローバルネットワークに登録されているトランジットゲートウェイを取得します。

```

aws networkmanager get-transit-gateway-registrations \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2

```

出力:

```

{
  "TransitGatewayRegistrations": [
    {
      "GlobalNetworkId": "global-network-01231231231231231",
      "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-
gateway/tgw-123abc05e04123abc",
      "State": {

```

```

    "Code": "AVAILABLE"
  }
}
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTransitGatewayRegistrations](#)」の「」を参照してください。

get-vpc-attachment

次のコード例は、get-vpc-attachment を使用する方法を示しています。

AWS CLI

VPCアタッチメントを取得するには

次のget-vpc-attachment例では、VPC添付ファイルに関する情報を返します。

```

aws networkmanager get-vpc-attachment \
  --attachment-id attachment-03b7ea450134787da

```

出力:

```

{
  "VpcAttachment": {
    "Attachment": {
      "CoreNetworkId": "core-network-0522de1b226a5d7b3",
      "AttachmentId": "attachment-03b7ea450134787da",
      "OwnerAccountId": "987654321012",
      "AttachmentType": "VPC",
      "State": "CREATING",
      "EdgeLocation": "us-east-1",
      "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",
      "Tags": [
        {
          "Key": "Name",
          "Value": "DevVPC"
        }
      ],
      "CreatedAt": "2022-03-11T17:48:58+00:00",
      "UpdatedAt": "2022-03-11T17:48:58+00:00"
    }
  }
}

```

```
    },
    "SubnetArns": [
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-202cde6c",
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-e5022dba",
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-2387ae02",
      "arn:aws:ec2:us-east-1:987654321012:subnet/subnet-cda9dfffc"
    ],
    "Options": {
      "Ipv6Support": false
    }
  }
}
```

詳細については、「[クラウドWANユーザーガイド](#)」の「[アタッチメント](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスGetVpcAttachment](#)」の「」を参照してください。

list-bucket-analytics-configurations

次のコード例は、list-bucket-analytics-configurations を使用する方法を示しています。

AWS CLI

バケットの分析設定のリストを取得するには

次の list-bucket-analytics-configurations の例では、指定したバケットの分析設定のリストを取得します。

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

出力:

```
{
  "AnalyticsConfigurationList": [
    {
      "StorageClassAnalysis": {},
      "Id": "1"
    }
  ],
}
```

```
"IsTruncated": false
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBucketAnalyticsConfigurations](#)」の「」を参照してください。

list-bucket-metrics-configurations

次の例は、list-bucket-metrics-configurations を使用する方法を説明しています。

AWS CLI

バケットのメトリクス設定のリストを取得するには

次のlist-bucket-metrics-configurations例では、指定されたバケットのメトリクス設定のリストを取得します。

```
aws s3api list-bucket-metrics-configurations \
  --bucket my-bucket
```

出力:

```
{
  "IsTruncated": false,
  "MetricsConfigurationList": [
    {
      "Filter": {
        "Prefix": "logs"
      },
      "Id": "123"
    },
    {
      "Filter": {
        "Prefix": "tmp"
      },
      "Id": "234"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBucketMetricsConfigurations](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたデバイスリソース () のタグを一覧表示しますdevice-07f6fd08867abc123。

```
aws networkmanager list-tags-for-resource \
  --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123 \
  --region us-west-2
```

出力:

```
{
  "TagList": [
    {
      "Key": "Network",
      "Value": "Northeast"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

put-bucket-metrics-configuration

次の例は、put-bucket-metrics-configuration を使用する方法を説明しています。

AWS CLI

バケットのメトリクス設定を設定するには

次のput-bucket-metrics-configuration例では、指定されたバケットの ID 123 のメトリクス設定を設定します。

```
aws s3api put-bucket-metrics-configuration \
```



```
--bucket my-bucket \  
--id 123 \  
--metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketMetricsConfiguration](#)」の「」を参照してください。

put-object-retention

次のコード例は、put-object-retention を使用する方法を示しています。

AWS CLI

オブジェクトのオブジェクト保持設定を設定するには

次の put-object-retention 例では、指定されたオブジェクトのオブジェクト保持設定を 2025-01-01 まで設定します。

```
aws s3api put-object-retention \  
--bucket my-bucket-with-object-lock \  
--key doc1.rtf \  
--retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutObjectRetention](#)」の「」を参照してください。

put-public-access-block

次のコード例は、put-public-access-block を使用する方法を示しています。

AWS CLI

バケットのブロックパブリックアクセス設定を設定するには

次のput-public-access-block例では、指定されたバケットの制限付きブロックパブリックアクセス設定を設定します。

```
aws s3api put-public-access-block \  

```

```
--bucket my-bucket \  
--public-access-block-  
configuration "BlockPublicAcls=true, IgnorePublicAcls=true, BlockPublicPolicy=true, RestrictPub
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutPublicAccessBlock](#)」の「」を参照してください。

register-transit-gateway

次の例は、register-transit-gateway を使用する方法を説明しています。

AWS CLI

グローバルネットワークにトランジットゲートウェイを登録するには

次のregister-transit-gateway例では、指定されたグローバルネットワークtgw-123abc05e04123abcにトランジットゲートウェイを登録します。

```
aws networkmanager register-transit-gateway \  
--global-network-id global-network-01231231231231231 \  
--transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-123abc05e04123abc \  
--region us-west-2
```

出力:

```
{  
  "TransitGatewayRegistration": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/  
tgw-123abc05e04123abc",  
    "State": {  
      "Code": "PENDING"  
    }  
  }  
}
```

詳細については、[「Transit Gateway Network Manager ガイド」](#)の「[Transit Gateway 登録](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterTransitGateway](#)」の「」を参照してください。

reject-attachment

次の例は、reject-attachment を使用する方法を説明しています。

AWS CLI

アタッチメントを拒否するには

次のreject-attachment例では、VPCアタッチメントリクエストを拒否します。

```
aws networkmanager reject-attachment \  
  --attachment-id attachment-03b7ea450134787da
```

出力:

```
{  
  "Attachment": {  
    "CoreNetworkId": "core-network-0522de1b226a5d7b3",  
    "AttachmentId": "attachment-03b7ea450134787da",  
    "OwnerAccountId": "987654321012",  
    "AttachmentType": "VPC",  
    "State": "AVAILABLE",  
    "EdgeLocation": "us-east-1",  
    "ResourceArn": "arn:aws:ec2:us-east-1:987654321012:vpc/vpc-a7c4bbda",  
    "CreatedAt": "2022-03-11T17:48:58+00:00",  
    "UpdatedAt": "2022-03-11T17:51:25+00:00"  
  }  
}
```

詳細については、「クラウドユーザーガイド」の「[添付ファイルの承諾](#)」を参照してください。

WAN

- API 詳細については、AWS CLI 「コマンドリファレンス[RejectAttachment](#)」の「」を参照してください。

start-route-analysis

次のコード例は、start-route-analysis を使用する方法を示しています。

AWS CLI

ルート分析を開始するには

次のstart-route-analysis例では、オプションの など、ソースと送信先間の分析を開始しますinclude-return-path。

```
aws networkmanager start-route-analysis \  
  --global-network-id global-network-00aa0aaa0b0aaa000 \  
  --source TransitGatewayAttachmentArn=arn:aws:ec2:us-east-1:503089527312:transit-  
gateway-attachment/tgw-attach-0d4a2d491bf68c093,IpAddress=10.0.0.0 \  
  --destination TransitGatewayAttachmentArn=arn:aws:ec2:us-  
west-1:503089527312:transit-gateway-attachment/tgw-  
attach-002577f30bb181742,IpAddress=11.0.0.0 \  
  --include-return-path
```

出力:

```
{  
  "RouteAnalysis": {  
    "GlobalNetworkId": "global-network-00aa0aaa0b0aaa000"  
    "OwnerId": "111122223333",  
    "RouteAnalysisId": "a1873de1-273c-470c-1a2bc2345678",  
    "StartTimestamp": 1695760154.0,  
    "Status": "RUNNING",  
    "Source": {  
      "TransitGatewayAttachmentArn": "arn:aws:ec2:us-  
east-1:111122223333:transit-gateway-attachment/tgw-attach-1234567890abcdef0",  
      "TransitGatewayArn": "arn:aws:ec2:us-east-1:111122223333:transit-  
gateway/tgw-abcdef01234567890",  
      "IpAddress": "10.0.0.0"  
    },  
    "Destination": {  
      "TransitGatewayAttachmentArn": "arn:aws:ec2:us-  
west-1:555555555555:transit-gateway-attachment/tgw-attach-021345abcdef6789",  
      "TransitGatewayArn": "arn:aws:ec2:us-west-1:111122223333:transit-  
gateway/tgw-09876543210fedcba0",  
      "IpAddress": "11.0.0.0"  
    },  
    "IncludeReturnPath": true,  
    "UseMiddleboxes": false  
  }  
}
```

詳細については、「Global AWS Networks for Transit Gateways ユーザーガイド」の「[Route Analyzer](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartRouteAnalysis](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを適用するには

次のtag-resource例では、タグNetwork=Northeastをデバイスに適用しますdevice-07f6fd08867abc123。

```
aws networkmanager tag-resource \  
  --resource-arn arn:aws:networkmanager::123456789012:device/global-  
network-01231231231231231/device-07f6fd08867abc123 \  
  --tags Key=Network,Value=Northeast \  
  --region us-west-2
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースからタグを削除する

次のuntag-resource例では、キーを持つタグをデバイス Network から削除しますdevice-07f6fd08867abc123。

```
aws networkmanager untag-resource \  
  --resource-arn arn:aws:networkmanager::123456789012:device/global-  
network-01231231231231231/device-07f6fd08867abc123 ]
```

```
--tag-keys Network \  
--region us-west-2
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-device

次の例は、update-device を使用する方法を説明しています。

AWS CLI

デバイスを更新するには

次のupdate-device例では、デバイスのサイト ID を指定device-07f6fd08867abc123してデバイスを更新します。

```
aws networkmanager update-device \  
--global-network-id global-network-01231231231231 \  
--device-id device-07f6fd08867abc123 \  
--site-id site-444555aaabbb11223 \  
--region us-west-2
```

出力:

```
{  
  "Device": {  
    "DeviceId": "device-07f6fd08867abc123",  
    "DeviceArn": "arn:aws:networkmanager::123456789012:device/global-network-01231231231231231231231231231231/device-07f6fd08867abc123",  
    "GlobalNetworkId": "global-network-01231231231231231231",  
    "Description": "NY office device",  
    "Type": "Office device",  
    "Vendor": "anycompany",  
    "Model": "abcabc",  
    "SerialNumber": "1234",  
    "SiteId": "site-444555aaabbb11223",  
    "CreatedAt": 1575554005.0,  
    "State": "UPDATING"  
  }  
}
```

```
}
```

詳細については、「Transit Gateway Network Manager ガイド」の [「デバイスの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateDevice](#)」の「」を参照してください。

update-global-network

次の例は、update-global-network を使用する方法を説明しています。

AWS CLI

グローバルネットワークを更新するには

次のupdate-global-network例では、グローバルネットワークの説明を更新しますglobal-network-01231231231231231。

```
aws networkmanager update-global-network \  
  --global-network-id global-network-01231231231231231 \  
  --description "Head offices" \  
  --region us-west-2
```

出力:

```
{  
  "GlobalNetwork": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-network/  
global-network-01231231231231231",  
    "Description": "Head offices",  
    "CreatedAt": 1575553525.0,  
    "State": "UPDATING"  
  }  
}
```

詳細については、「Transit Gateway Network Manager ガイド」の [「グローバルネットワーク」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateGlobalNetwork](#)」の「」を参照してください。

update-link

次の例は、update-link を使用する方法を説明しています。

AWS CLI

リンクを更新するには

次のupdate-link例では、リンク の帯域幅情報を更新しますlink-11112222aaaabbbb1。

```
aws networkmanager update-link \  
  --global-network-id global-network-01231231231231231 \  
  --link-id link-11112222aaaabbbb1 \  
  --bandwidth UploadSpeed=20,DownloadSpeed=20 \  
  --region us-west-2
```

出力:

```
{  
  "Link": {  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkArn": "arn:aws:networkmanager::123456789012:link/global-  
network-01231231231231231/link-11112222aaaabbbb1",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "SiteId": "site-444555aaaabbb11223",  
    "Description": "VPN Link",  
    "Type": "broadband",  
    "Bandwidth": {  
      "UploadSpeed": 20,  
      "DownloadSpeed": 20  
    },  
    "Provider": "AnyCompany",  
    "CreatedAt": 1575555811.0,  
    "State": "UPDATING"  
  }  
}
```

詳細については、「[Transit Gateway Network Manager ガイド](#)」の「[リンクの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateLink](#)」の「」を参照してください。

update-site

次のコード例は、update-site を使用する方法を示しています。

AWS CLI

サイトを更新するには

次のupdate-site例では、指定されたグローバルネットワークsite-444555aaabbb11223内のサイトの説明を更新します。

```
aws networkmanager update-site \  
  --global-network-id global-network-01231231231231231 \  
  --site-id site-444555aaabbb11223 \  
  --description "New York Office site" \  
  --region us-west-2
```

出力:

```
{  
  "Site": {  
    "SiteId": "site-444555aaabbb11223",  
    "SiteArn": "arn:aws:networkmanager::123456789012:site/global-  
network-01231231231231231/site-444555aaabbb11223",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "Description": "New York Office site",  
    "Location": {  
      "Latitude": "40.7128",  
      "Longitude": "-74.0060"  
    },  
    "CreatedAt": 1575554528.0,  
    "State": "UPDATING"  
  }  
}
```

詳細については、「Transit Gateway Network Manager ガイド」の[「サイトの使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSite](#)」の「」を参照してください。

を使用した Nimble Studio の例 AWS CLI

次のコード例は、Nimble Studio AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

get-eula

次のコード例は、get-eula を使用する方法を示しています。

AWS CLI

スタジオに関する情報を取得するには

次のget-eula例は、に関する情報を示していますEULA。

```
aws nimble get-eula \  
  --eula-id "EULAid"
```

出力:

```
{  
  "eula": {  
    "content": "https://www.mozilla.org/en-US/MPL/2.0/",  
    "createdAt": "2021-04-20T16:45:23+00:00",  
    "eulaId": "gJZLygd-Srq_5NNbSfiaLg",  
    "name": "Mozilla-FireFox",  
    "updatedAt": "2021-04-20T16:45:23+00:00"  
  }  
}
```

詳細については、「Amazon Nimble Studio ユーザーガイド」の「[を受け入れるEULA](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEula](#)」の「」を参照してください。

get-launch-profile-details

次の例は、get-launch-profile-details を使用する方法を説明しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のget-launch-profile-details例では、起動プロファイルの詳細を一覧表示します。

```
aws nimble get-launch-profile-details \  
  --studio-id "StudioID" \  
  --launch-profile-id "LaunchProfileID"
```

出力:

```
{  
  "launchProfile": {  
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/  
yeG71DwNQEiwNTRT7DrV7Q",  
    "createdAt": "2022-01-27T21:18:59+00:00",  
    "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",  
    "description": "The Launch Profile for the Render workers created by  
StudioBuilder.",  
    "ec2SubnetIds": [  
      "subnet-EXAMPLE11111"  
    ],  
    "launchProfileId": "yeG71DwNQEiwNTRT7DrV7Q",  
    "launchProfileProtocolVersions": [  
      "2021-03-31"  
    ],  
    "name": "RenderWorker-Default",  
    "state": "READY",  
    "statusCode": "LAUNCH_PROFILE_CREATED",  
    "statusMessage": "Launch Profile has been created",  
    "streamConfiguration": {  
      "clipboardMode": "ENABLED",
```

```
    "ec2InstanceTypes": [
      "g4dn.4xlarge",
      "g4dn.8xlarge"
    ],
    "maxSessionLengthInMinutes": 690,
    "maxStoppedSessionLengthInMinutes": 0,
    "streamingImageIds": [
      "Cw_jXnp1QcSSXhE2hkNRoQ",
      "YGXAqgoWTnCNSV8VP20sHQ"
    ]
  },
  "studioComponentIds": [
    "_hR_-RaAReS0jAnLakbX7Q",
    "vQ5w_TbIRayPkAZgcbyYRA",
    "ZQuMxN99Qfa_Js6ma9TwdA",
    "45Kj0SPPRzK20yvpCuQ6qw"
  ],
  "tags": {
    "resourceArn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q"
  },
  "updatedAt": "2022-01-27T21:19:13+00:00",
  "updatedBy": "AR0A3002NEHCCYRNDDIFT:i-00b98256b04d9e989",
  "validationResults": [
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_SUBNET_ASSOCIATION"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
    },
    {
      "state": "VALIDATION_SUCCESS",
```

```
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
    }
  ]
},
"streamingImages": [
  {
    "arn": "arn:aws:nimble:us-west-2:123456789012:streaming-image/
Cw_jXnp1QcSSXhE2hkNRoQ",
    "description": "Base windows image for NimbleStudio",
    "ec2ImageId": "ami-EXAMPLE11111",
    "eulaIds": [
      "gJZLygd-Srq_5NNbSfiaLg",
      "ggK2eIw6RQyt8PIee0lD3g",
      "a-D9Wc0VQCKUfxAinCDxaw",
      "RvoNmVXiSrS4LhLTb6ybkw",
      "wtp85BcSTa2NZeNRnMKdjw",
      "Rl-J0fM5S12hyIiwWIV6hw"
    ],
    "name": "NimbleStudioWindowsStreamImage",
    "owner": "amazon",
    "platform": "WINDOWS",
    "state": "READY",
    "streamingImageId": "Cw_jXnp1QcSSXhE2hkNRoQ",
    "tags": {
      "resourceArn": "arn:aws:nimble:us-west-2:123456789012:streaming-
image/Cw_jXnp1QcSSXhE2hkNRoQ"
    }
  },
  {
    "arn": "arn:aws:nimble:us-west-2:123456789012:streaming-image/
YGXAqgoWTnCNSV8VP20sHQ",
    "description": "Base linux image for NimbleStudio",
    "ec2ImageId": "ami-EXAMPLE11111",
    "eulaIds": [
      "gJZLygd-Srq_5NNbSfiaLg",
      "ggK2eIw6RQyt8PIee0lD3g",
      "a-D9Wc0VQCKUfxAinCDxaw",
      "RvoNmVXiSrS4LhLTb6ybkw",
      "wtp85BcSTa2NZeNRnMKdjw",
      "Rl-J0fM5S12hyIiwWIV6hw"
    ],
    "name": "NimbleStudioLinuxStreamImage",
```

```
    "owner": "amazon",
    "platform": "LINUX",
    "state": "READY",
    "streamingImageId": "YGXAqgoWTnCNSV8VP20sHQ",
    "tags": {
      "resourceArn": "arn:aws:nimble:us-west-2:123456789012:streaming-
image/YGXAqgoWTnCNSV8VP20sHQ"
    }
  ],
  "studioComponentSummaries": [
    {
      "description": "FSx for Windows",
      "name": "FSxWindows",
      "studioComponentId": "ZQuMxN99Qfa_Js6ma9TwdA",
      "subtype": "AMAZON_FSX_FOR_WINDOWS",
      "type": "SHARED_FILE_SYSTEM"
    },
    {
      "description": "Instance configuration studio component.",
      "name": "InstanceConfiguration",
      "studioComponentId": "vQ5w_TbIRayPkAZgcbyYRA",
      "subtype": "CUSTOM",
      "type": "CUSTOM"
    },
    {
      "name": "ActiveDirectory",
      "studioComponentId": "_hR_-RaAReS0jAnLakbX7Q",
      "subtype": "AWS_MANAGED_MICROSOFT_AD",
      "type": "ACTIVE_DIRECTORY"
    },
    {
      "description": "Render farm running Deadline",
      "name": "RenderFarm",
      "studioComponentId": "45Kj0SPPRzK20yvpCuQ6qw",
      "subtype": "CUSTOM",
      "type": "COMPUTE_FARM"
    }
  ]
}
```

詳細については、「Amazon Nimble Studio ユーザーガイド」の [「起動プロファイルの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLaunchProfileDetails](#)」の「」を参照してください。

get-launch-profile

次のコード例は、get-launch-profile を使用する方法を示しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のget-launch-profile例は、起動プロファイルに関する情報を示しています。

```
aws nimble get-launch-profile \  
  --studio-id "StudioID" \  
  --launch-profile-id "LaunchProfileID"
```

出力:

```
{  
  "launchProfile": {  
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/  
yeG7lDwNQEiwNTRT7DrV7Q",  
    "createdAt": "2022-01-27T21:18:59+00:00",  
    "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",  
    "description": "The Launch Profile for the Render workers created by  
StudioBuilder.",  
    "ec2SubnetIds": [  
      "subnet-EXAMPLE11111"  
    ],  
    "launchProfileId": "yeG7lDwNQEiwNTRT7DrV7Q",  
    "launchProfileProtocolVersions": [  
      "2021-03-31"  
    ],  
    "name": "RenderWorker-Default",  
    "state": "READY",  
    "statusCode": "LAUNCH_PROFILE_CREATED",  
    "statusMessage": "Launch Profile has been created",  
    "streamConfiguration": {  
      "clipboardMode": "ENABLED",  
      "ec2InstanceTypes": [  
        "g4dn.4xlarge",  
        "g4dn.8xlarge"  
      ]  
    }  
  }  
}
```

```
    ],
    "maxSessionLengthInMinutes": 690,
    "maxStoppedSessionLengthInMinutes": 0,
    "streamingImageIds": [
      "Cw_jXnp1QcSSXhE2hkNRoQ",
      "YGXAqgoWTnCNSV8VP20sHQ"
    ]
  },
  "studioComponentIds": [
    "_hR_-RaAReS0jAnLakbX7Q",
    "vQ5w_TbIRayPkAZgcbYRA",
    "ZQuMxN99Qfa_Js6ma9TwdA",
    "45Kj0SPPRzK20yvpCuQ6qw"
  ],
  "tags": {},
  "updatedAt": "2022-01-27T21:19:13+00:00",
  "updatedBy": "AR0A3002NEHCCYRNDDIFT:i-00b98256b04d9e989",
  "validationResults": [
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_SUBNET_ASSOCIATION"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
    }
  ]
}
```



```
}
```

詳細については、「Amazon Nimble Studio ユーザーガイド」の「[起動プロファイルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLaunchProfile](#)」の「」を参照してください。

get-studio

次の例は、get-studio を使用する方法を説明しています。

AWS CLI

スタジオに関する情報を取得するには

次のget-studio例では、AWS アカウントのスタジオを一覧表示します。

```
aws nimble get-studio \  
  --studio-id "StudioID"
```

出力:

```
{  
  "studio": {  
    "adminRoleArn": "arn:aws:iam::123456789012:role/studio-admin-role",  
    "arn": "arn:aws:nimble:us-west-2:123456789012:studio/stid-EXAMPLE11111",  
    "createdAt": "2022-01-27T20:29:35+00:00",  
    "displayName": "studio-name",  
    "homeRegion": "us-west-2",  
    "ssoClientId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "state": "READY",  
    "statusCode": "STUDIO_CREATED",  
    "statusMessage": "The studio has been created successfully ",  
    "studioEncryptionConfiguration": {  
      "keyType": "AWS_OWNED_KEY"  
    },  
    "studioId": "us-west-2:stid-EXAMPLE11111",  
    "studioName": "studio-name",  
    "studioUrl": "https://studio-name.nimblestudio.us-west-2.amazonaws.com",  
    "tags": {},  
    "updatedAt": "2022-01-27T20:29:37+00:00",  
    "userRoleArn": "arn:aws:iam::123456789012:role/studio-user-role"  
  }  
}
```

```
}  
}
```

詳細については、[「Amazon Nimble Studio ユーザーガイド」の「Amazon Nimble Studio とは」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetStudio](#)」の「」を参照してください。

list-eula-acceptances

次の例は、list-eula-acceptances を使用する方法を説明しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のlist-eula-acceptances例では、AWS アカウントEULAsで受け入れられたを一覧表示します。

```
aws nimble list-eula-acceptances \  
  --studio-id "StudioID"
```

出力:

```
{  
  "eulaAcceptances": [  
    {  
      "acceptedAt": "2022-01-28T17:44:35+00:00",  
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",  
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",  
      "eulaAcceptanceId": "V0JlpZQaSx6yHcUuX0qfQw",  
      "eulaId": "R1-J0fM5S12hyIiwWIV6hw"  
    },  
    {  
      "acceptedAt": "2022-01-28T17:44:35+00:00",  
      "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",  
      "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",  
      "eulaAcceptanceId": "YY_uDFW-SVibc627qbug0Q",  
      "eulaId": "RvoNmVXiSrS4LhLTb6ybkw"  
    },  
  ],  
}
```

```
    "acceptedAt": "2022-01-28T17:44:35+00:00",
    "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
    "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
    "eulaAcceptanceId": "ov087PnhQ4-MpttiL5uN6Q",
    "eulaId": "a-D9Wc0VQCKUfxAinCDxaw"
  },
  {
    "acceptedAt": "2022-01-28T17:44:35+00:00",
    "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
    "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
    "eulaAcceptanceId": "5YeXje4yR0amuTESGvqIAQ",
    "eulaId": "gJZLygd-Srq_5NNbSfiaLg"
  },
  {
    "acceptedAt": "2022-01-28T17:44:35+00:00",
    "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
    "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
    "eulaAcceptanceId": "W1sIn8PtScqeJEn8sxxhgw",
    "eulaId": "ggK2eIw6RQyt8PIee01D3g"
  },
  {
    "acceptedAt": "2022-01-28T17:44:35+00:00",
    "acceptedBy": "92677b4b19-e9fd012a-94ad-4f16-9866-c69a63ab6486",
    "accepteeId": "us-west-2:stid-nyoqq12fteqy1x48",
    "eulaAcceptanceId": "Zq9KNEQPRMWJ7FolSoQgUA",
    "eulaId": "wtp85BcSTa2NZeNRnMKdjw"
  }
]
}
```

詳細については、「Amazon Nimble Studio ユーザーガイド」の「[を受け入れるEULA](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListEulaAcceptances](#)」の「」を参照してください。

list-eulas

次の例は、list-eulas を使用する方法を説明しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のlist-eulas例では、AWS アカウントのを一覧表示EULAsします。

```
aws nimble list-eulas
```

出力:

```
{
  "eulas": [
    {
      "content": "https://www.mozilla.org/en-US/MPL/2.0/",
      "createdAt": "2021-04-20T16:45:23+00:00",
      "eulaId": "gJZLygd-Srq_5NNbSfiaLg",
      "name": "Mozilla-FireFox",
      "updatedAt": "2021-04-20T16:45:23+00:00"
    },
    {
      "content": "https://www.awsthinkbox.com/end-user-license-agreement",
      "createdAt": "2021-04-20T16:45:24+00:00",
      "eulaId": "RvoNmVXiSrS4LhLTb6ybkw",
      "name": "Thinkbox-Deadline",
      "updatedAt": "2021-04-20T16:45:24+00:00"
    },
    {
      "content": "https://www.videolan.org/legal.html",
      "createdAt": "2021-04-20T16:45:24+00:00",
      "eulaId": "Rl-J0fM5S12hyIiwWIV6hw",
      "name": "Videolan-VLC",
      "updatedAt": "2021-04-20T16:45:24+00:00"
    },
    {
      "content": "https://code.visualstudio.com/license",
      "createdAt": "2021-04-20T16:45:23+00:00",
      "eulaId": "ggK2eIw6RQyt8PIee0lD3g",
      "name": "Microsoft-VSCode",
      "updatedAt": "2021-04-20T16:45:23+00:00"
    },
    {
      "content": "https://darbyjohnston.github.io/DJV/legal.html#License",
      "createdAt": "2021-04-20T16:45:23+00:00",
      "eulaId": "wtp85BcSTa2NZeNRnMKdjw",
      "name": "DJV-DJV",
      "updatedAt": "2021-04-20T16:45:23+00:00"
    }
  ],
}
```

```
{
  "content": "https://www.sidefx.com/legal/license-agreement/",
  "createdAt": "2021-04-20T16:45:24+00:00",
  "eulaId": "uu2VDLo-QJeIGWWLBae_UA",
  "name": "SideFX-Houdini",
  "updatedAt": "2021-04-20T16:45:24+00:00"
},
{
  "content": "https://www.chaosgroup.com/eula",
  "createdAt": "2021-04-20T16:45:23+00:00",
  "eulaId": "L0HS4P3CRYKVXc2J2L07Vw",
  "name": "ChaosGroup-Vray",
  "updatedAt": "2021-04-20T16:45:23+00:00"
},
{
  "content": "https://www.foundry.com/eula",
  "createdAt": "2021-04-20T16:45:23+00:00",
  "eulaId": "SAuhfHmSAeUuq3wsMiMlw",
  "name": "Foundry-Nuke",
  "updatedAt": "2021-04-20T16:45:23+00:00"
},
{
  "content": "https://download.blender.org/release/GPL3-license.txt",
  "createdAt": "2021-04-20T16:45:23+00:00",
  "eulaId": "a-D9Wc0VQCKUfxAinCDxaw",
  "name": "BlenderFoundation-Blender",
  "updatedAt": "2021-04-20T16:45:23+00:00"
}
]
```

詳細については、「Amazon Nimble Studio ユーザーガイド」の「[を受け入れるEULA](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEulas](#)」の「」を参照してください。

list-launch-profiles

次の例は、list-launch-profiles を使用する方法を説明しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のlist-launch-profiles例では、AWS アカウントの起動プロファイルを一覧表示します。

```
aws nimble list-launch-profiles \  
  --studio-id "StudioID"
```

出力:

```
{  
  "launchProfiles": [  
    {  
      "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/  
yeG7lDwNQEiwNTRT7DrV7Q",  
      "createdAt": "2022-01-27T21:18:59+00:00",  
      "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",  
      "description": "The Launch Profile for the Render workers created by  
StudioBuilder.",  
      "ec2SubnetIds": [  
        "subnet-EXAMPLE11111"  
      ],  
      "launchProfileId": "yeG7lDwNQEiwNTRT7DrV7Q",  
      "launchProfileProtocolVersions": [  
        "2021-03-31"  
      ],  
      "name": "RenderWorker-Default",  
      "state": "READY",  
      "statusCode": "LAUNCH_PROFILE_CREATED",  
      "statusMessage": "Launch Profile has been created",  
      "streamConfiguration": {  
        "clipboardMode": "ENABLED",  
        "ec2InstanceTypes": [  
          "g4dn.4xlarge",  
          "g4dn.8xlarge"  
        ],  
        "maxSessionLengthInMinutes": 690,  
        "maxStoppedSessionLengthInMinutes": 0,  
        "streamingImageIds": [  
          "Cw_jXnp1QcSSXhE2hkNRoQ",  
          "YGXAqgoWTnCNSV8VP20sHQ"  
        ]  
      }  
    }  
  ]  
}
```

```
    ]
  },
  "studioComponentIds": [
    "_hR_-RaAReS0jAnLakbX7Q",
    "vQ5w_TbIRayPkAZgcbYRA",
    "ZQuMxN99Qfa_Js6ma9TwdA",
    "45Kj0SPPrzK20yvpCuQ6qw"
  ],
  "tags": {},
  "updatedAt": "2022-01-27T21:19:13+00:00",
  "updatedBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
  "validationResults": [
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_SUBNET_ASSOCIATION"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
    },
    {
      "state": "VALIDATION_SUCCESS",
      "statusCode": "VALIDATION_SUCCESS",
      "statusMessage": "The validation succeeded.",
      "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
    }
  ]
},
{
  "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
jDCIm1jRSaa9e44PZ3w7gg",
  "createdAt": "2022-01-27T21:19:26+00:00",
  "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
```

```
"description": "This Workstation Launch Profile was created by
StudioBuilder",
"ec2SubnetIds": [
  "subnet-046f4205ae535b2cc"
],
"launchProfileId": "jDCIm1jRSaa9e44PZ3w7gg",
"launchProfileProtocolVersions": [
  "2021-03-31"
],
"name": "Workstation-Default",
"state": "READY",
"statusCode": "LAUNCH_PROFILE_CREATED",
"statusMessage": "Launch Profile has been created",
"streamConfiguration": {
  "clipboardMode": "ENABLED",
  "ec2InstanceTypes": [
    "g4dn.4xlarge",
    "g4dn.8xlarge"
  ],
  "maxSessionLengthInMinutes": 690,
  "maxStoppedSessionLengthInMinutes": 0,
  "streamingImageIds": [
    "Cw_jXnp1QcSSXhE2hkNRoQ",
    "YGXAqgoWTnCNSV8VP20sHQ"
  ]
},
"studioComponentIds": [
  "_hR_-RaAReS0jAnLakbX7Q",
  "vQ5w_TbIRayPkAZgcbyYRA",
  "ZQuMxN99Qfa_Js6ma9TwdA",
  "yJSbsHXAQYwk9FXLNusX1Q",
  "45Kj0SPPrzK20yvpCuQ6qw"
],
"tags": {},
"updatedAt": "2022-01-27T21:19:40+00:00",
"updatedBy": "AR0A3002NEHCCYRNDIIFT:i-EXAMPLE11111",
"validationResults": [
  {
    "state": "VALIDATION_SUCCESS",
    "statusCode": "VALIDATION_SUCCESS",
    "statusMessage": "The validation succeeded.",
    "type": "VALIDATE_ACTIVE_DIRECTORY_STUDIO_COMPONENT"
  }
]
```



```
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SUBNET_ASSOCIATION"
    },
    {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_NETWORK_ACL_ASSOCIATION"
    },
    {
        "state": "VALIDATION_SUCCESS",
        "statusCode": "VALIDATION_SUCCESS",
        "statusMessage": "The validation succeeded.",
        "type": "VALIDATE_SECURITY_GROUP_ASSOCIATION"
    }
]
}
```

詳細については、「Amazon Nimble Studio ユーザーガイド」の「[起動プロファイルの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLaunchProfiles](#)」の「」を参照してください。

list-studio-components

次の例は、list-studio-components を使用する方法を説明しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のlist-studio-components例では、AWS アカウントのスタジオコンポーネントを一覧表示します。

```
aws nimble list-studio-components \
  --studio-id "StudioID"
```

出力:

```
{
  "studioComponents": [
    {
      "arn": "arn:aws:nimble:us-west-2:123456789012:studio-component/
ZQuMxN99Qfa_Js6ma9TwdA",
      "configuration": {
        "sharedFileSystemConfiguration": {
          "fileSystemId": "fs-EXAMPLE11111",
          "linuxMountPoint": "/mnt/fsxshare",
          "shareName": "share",
          "windowsMountDrive": "Z"
        }
      },
      "createdAt": "2022-01-27T21:15:34+00:00",
      "createdBy": "AROA3002NEHCCYRNDIIFT:i-EXAMPLE11111",
      "description": "FSx for Windows",
      "ec2SecurityGroupIds": [
        "sg-EXAMPLE11111"
      ],
      "name": "FSxWindows",
      "state": "READY",
      "statusCode": "STUDIO_COMPONENT_CREATED",
      "statusMessage": "Studio Component has been created",
      "studioComponentId": "ZQuMxN99Qfa_Js6ma9TwdA",
      "subtype": "AMAZON_FSX_FOR_WINDOWS",
      "tags": {},
      "type": "SHARED_FILE_SYSTEM",
      "updatedAt": "2022-01-27T21:15:35+00:00",
      "updatedBy": "AROA3002NEHCCYRNDIIFT:i-EXAMPLE11111"
    },
    ...
  ]
}
```

詳細については、[「Amazon Nimble Studio ユーザーガイド」の「が Amazon Nimble Studio と StudioBuilder 連携する方法」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStudioComponents](#)」の「」を参照してください。

list-studio-members

次のコード例は、list-studio-members を使用する方法を示しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のlist-studio-members例では、AWS アカウントで利用可能なスタジオメンバーを一覧表示します。

```
aws nimble list-studio-members \  
  --studio-id "StudioID"
```

出力:

```
{  
  "members": [  
    {  
      "identityStoreId": "d-EXAMPLE11111",  
      "persona": "ADMINISTRATOR",  
      "principalId": "EXAMPLE11111-e9fd012a-94ad-4f16-9866-c69a63ab6486"  
    }  
  ]  
}
```

詳細については、「Amazon Nimble Studio ユーザーガイド」の「[スタジオユーザーの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStudioMembers](#)」の「」を参照してください。

list-studios

次の例は、list-studios を使用する方法を説明しています。

AWS CLI

スタジオを一覧表示するには

次のlist-studios例では、AWS アカウントのスタジオを一覧表示します。

```
aws nimble list-studios
```

出力:

```
{
  "studios": [
    {
      "adminRoleArn": "arn:aws:iam::123456789012:role/studio-admin-role",
      "arn": "arn:aws:nimble:us-west-2:123456789012:studio/studio-id",
      "createdAt": "2022-01-27T20:29:35+00:00",
      "displayName": "studio-name",
      "homeRegion": "us-west-2",
      "ssoClientId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "state": "READY",
      "statusCode": "STUDIO_CREATED",
      "statusMessage": "The studio has been created successfully ",
      "studioEncryptionConfiguration": {
        "keyType": "AWS_OWNED_KEY"
      },
      "studioId": "us-west-2:studio-id",
      "studioName": "studio-name",
      "studioUrl": "https://studio-name.nimblestudio.us-west-2.amazonaws.com",
      "tags": {},
      "updatedAt": "2022-01-27T20:29:37+00:00",
      "userRoleArn": "arn:aws:iam::123456789012:role/studio-user-role"
    }
  ]
}
```

詳細については、[「Amazon Nimble Studio ユーザーガイド」の「Amazon Nimble Studio とは」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStudios](#)」の「」を参照してください。

OpenSearch を使用したサービス例 AWS CLI

次のコード例は、OpenSearch AWS Command Line Interface サービスで を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-elasticsearch-domain

次の例は、create-elasticsearch-domain を使用する方法を説明しています。

AWS CLI

Amazon Elasticsearch Service ドメインを作成するには

次のcreate-elasticsearch-domainコマンドは、内に新しい Amazon Elasticsearch Service ドメインを作成しVPC、単一のユーザーへのアクセスを制限します。Amazon ES は、指定されたサブネットとセキュリティグループ から VPC ID を推測しますIDs。

```
aws es create-elasticsearch-domain \
  --domain-name vpc-cli-example \
  --elasticsearch-version 6.2 \
  --elasticsearch-cluster-
config InstanceType=m4.large.elasticsearch,InstanceCount=1 \
  --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 \
  --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect":
"Allow", "Principal": { "AWS": "arn:aws:iam:123456789012:root" }, "Action": "es:*",
"Resource": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*" } ] }' \
  --vpc-options SubnetIds=subnet-1a2a3a4a,SecurityGroupIds=sg-2a3a4a5a
```

出力:

```
{
  "DomainStatus": {
    "ElasticsearchClusterConfig": {
      "DedicatedMasterEnabled": false,
      "InstanceCount": 1,
      "ZoneAwarenessEnabled": false,
      "InstanceType": "m4.large.elasticsearch"
    },
    "DomainId": "123456789012/vpc-cli-example",
    "CognitoOptions": {
      "Enabled": false
    }
  }
}
```

```

    },
    "VPCOptions": {
      "SubnetIds": [
        "subnet-1a2a3a4a"
      ],
      "VPCId": "vpc-3a4a5a6a",
      "SecurityGroupIds": [
        "sg-2a3a4a5a"
      ],
      "AvailabilityZones": [
        "us-west-1c"
      ]
    },
    "Created": true,
    "Deleted": false,
    "EBSOptions": {
      "VolumeSize": 10,
      "VolumeType": "standard",
      "EBSEnabled": true
    },
    "Processing": true,
    "DomainName": "vpc-cli-example",
    "SnapshotOptions": {
      "AutomatedSnapshotStartHour": 0
    },
    "ElasticsearchVersion": "6.2",
    "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":[\"es:*\"],\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*\"}]}",
    "AdvancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    },
    "EncryptionAtRestOptions": {
      "Enabled": false
    },
    "ARN": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example"
  }
}

```

詳細については、[Amazon Elasticsearch Service デベロッパーガイド](#)の「[Amazon Elasticsearch Service ドメインの作成と管理](#)」を参照してください。 Amazon Elasticsearch Service

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateElasticsearchDomain](#)」の「」を参照してください。

describe-elasticsearch-domain-config

次の例は、describe-elasticsearch-domain-config を使用する方法を説明しています。

AWS CLI

ドメイン設定の詳細を取得するには

次のdescribe-elasticsearch-domain-config例では、特定のドメインの設定の詳細と、個々のドメインコンポーネントのステータス情報を示します。

```
aws es describe-elasticsearch-domain-config \  
  --domain-name cli-example
```

出力:

```
{  
  "DomainConfig": {  
    "ElasticsearchVersion": {  
      "Options": "7.4",  
      "Status": {  
        "CreationDate": 1589395034.946,  
        "UpdateDate": 1589395827.325,  
        "UpdateVersion": 8,  
        "State": "Active",  
        "PendingDeletion": false  
      }  
    },  
    "ElasticsearchClusterConfig": {  
      "Options": {  
        "InstanceType": "c5.large.elasticsearch",  
        "InstanceCount": 1,  
        "DedicatedMasterEnabled": true,  
        "ZoneAwarenessEnabled": false,  
        "DedicatedMasterType": "c5.large.elasticsearch",  
        "DedicatedMasterCount": 3,  
        "WarmEnabled": true,  
        "WarmType": "ultrawarm1.medium.elasticsearch",  
        "WarmCount": 2  
      }  
    }  
  }  
}
```

```
    },
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "EBSOptions": {
    "Options": {
      "EBSEnabled": true,
      "VolumeType": "gp2",
      "VolumeSize": 10
    },
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "AccessPolicies": {
    "Options": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]}",
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
      "PendingDeletion": false
    }
  },
  "SnapshotOptions": {
    "Options": {
      "AutomatedSnapshotStartHour": 0
    },
    "Status": {
      "CreationDate": 1589395034.946,
      "UpdateDate": 1589395827.325,
      "UpdateVersion": 8,
      "State": "Active",
```



```
        "PendingDeletion": false
      }
    },
    "VPCOptions": {
      "Options": {},
      "Status": {
        "CreationDate": 1591210426.162,
        "UpdateDate": 1591210426.162,
        "UpdateVersion": 18,
        "State": "Active",
        "PendingDeletion": false
      }
    },
    "CognitoOptions": {
      "Options": {
        "Enabled": false
      },
      "Status": {
        "CreationDate": 1591210426.163,
        "UpdateDate": 1591210426.163,
        "UpdateVersion": 18,
        "State": "Active",
        "PendingDeletion": false
      }
    },
    "EncryptionAtRestOptions": {
      "Options": {
        "Enabled": true,
        "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
      },
      "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589395827.325,
        "UpdateVersion": 8,
        "State": "Active",
        "PendingDeletion": false
      }
    },
    "NodeToNodeEncryptionOptions": {
      "Options": {
        "Enabled": true
      },
      "Status": {
```

```
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589395827.325,
        "UpdateVersion": 8,
        "State": "Active",
        "PendingDeletion": false
    }
},
"AdvancedOptions": {
    "Options": {
        "rest.action.multi.allow_explicit_index": "true"
    },
    "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589395827.325,
        "UpdateVersion": 8,
        "State": "Active",
        "PendingDeletion": false
    }
},
"LogPublishingOptions": {
    "Options": {},
    "Status": {
        "CreationDate": 1591210426.164,
        "UpdateDate": 1591210426.164,
        "UpdateVersion": 18,
        "State": "Active",
        "PendingDeletion": false
    }
},
"DomainEndpointOptions": {
    "Options": {
        "EnforceHTTPS": true,
        "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
    },
    "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589395827.325,
        "UpdateVersion": 8,
        "State": "Active",
        "PendingDeletion": false
    }
},
"AdvancedSecurityOptions": {
    "Options": {
```

```
        "Enabled": true,
        "InternalUserDatabaseEnabled": true
    },
    "Status": {
        "CreationDate": 1589395034.946,
        "UpdateDate": 1589827485.577,
        "UpdateVersion": 14,
        "State": "Active",
        "PendingDeletion": false
    }
}
}
```

詳細については、[Amazon Elasticsearch Service デベロッパーガイド](#)の「[Amazon Elasticsearch Service ドメインの作成と管理](#)」を参照してください。Amazon Elasticsearch Service

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeElasticsearchDomainConfig](#)」の「」を参照してください。

describe-elasticsearch-domain

次の例は、describe-elasticsearch-domain を使用する方法を説明しています。

AWS CLI

単一のドメインの詳細を取得するには

次のdescribe-elasticsearch-domain例は、特定のドメインの設定の詳細を示しています。

```
aws es describe-elasticsearch-domain \
  --domain-name cli-example
```

出力:

```
{
  "DomainStatus": {
    "DomainId": "123456789012/cli-example",
    "DomainName": "cli-example",
    "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example",
    "Created": true,
    "Deleted": false,
```

```
    "Endpoint": "search-cli-example-1a2a3a4a5a6a7a8a9a0a.us-
east-1.es.amazonaws.com",
    "Processing": false,
    "UpgradeProcessing": false,
    "ElasticsearchVersion": "7.4",
    "ElasticsearchClusterConfig": {
      "InstanceType": "c5.large.elasticsearch",
      "InstanceCount": 1,
      "DedicatedMasterEnabled": true,
      "ZoneAwarenessEnabled": false,
      "DedicatedMasterType": "c5.large.elasticsearch",
      "DedicatedMasterCount": 3,
      "WarmEnabled": true,
      "WarmType": "ultrawarm1.medium.elasticsearch",
      "WarmCount": 2
    },
    "EBSOptions": {
      "EBSEnabled": true,
      "VolumeType": "gp2",
      "VolumeSize": 10
    },
    "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\": \"Allow\", \"Principal\": {\"AWS\": \"*\"}, \"Action\": \"es:*\", \"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/cli-example/*\"}]}",
    "SnapshotOptions": {
      "AutomatedSnapshotStartHour": 0
    },
    "CognitoOptions": {
      "Enabled": false
    },
    "EncryptionAtRestOptions": {
      "Enabled": true,
      "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
    },
    "NodeToNodeEncryptionOptions": {
      "Enabled": true
    },
    "AdvancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    },
    "ServiceSoftwareOptions": {
      "CurrentVersion": "R20200522",
      "NewVersion": "",
```

```

        "UpdateAvailable": false,
        "Cancellable": false,
        "UpdateStatus": "COMPLETED",
        "Description": "There is no software update available for this domain.",
        "AutomatedUpdateDate": 0.0
    },
    "DomainEndpointOptions": {
        "EnforceHTTPS": true,
        "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
    },
    "AdvancedSecurityOptions": {
        "Enabled": true,
        "InternalUserDatabaseEnabled": true
    }
}
}

```

詳細については、[Amazon Elasticsearch Service デベロッパーガイド](#)の「[Amazon Elasticsearch Service ドメインの作成と管理](#)」を参照してください。Amazon Elasticsearch Service

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeElasticsearchDomain](#)」の「」を参照してください。

describe-elasticsearch-domains

次のコード例は、describe-elasticsearch-domains を使用する方法を示しています。

AWS CLI

1 つ以上のドメインの詳細を取得するには

次のdescribe-elasticsearch-domains例では、1 つ以上のドメインの設定の詳細を示します。

```
aws es describe-elasticsearch-domains \
  --domain-names cli-example-1 cli-example-2
```

出力:

```
{
  "DomainStatusList": [{
```

```
"DomainId": "123456789012/cli-example-1",
"DomainName": "cli-example-1",
"ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-1",
"Created": true,
"Deleted": false,
"Endpoint": "search-cli-example-1-1a2a3a4a5a6a7a8a9a0a.us-
east-1.es.amazonaws.com",
"Processing": false,
"UpgradeProcessing": false,
"ElasticsearchVersion": "7.4",
"ElasticsearchClusterConfig": {
  "InstanceType": "c5.large.elasticsearch",
  "InstanceCount": 1,
  "DedicatedMasterEnabled": true,
  "ZoneAwarenessEnabled": false,
  "DedicatedMasterType": "c5.large.elasticsearch",
  "DedicatedMasterCount": 3,
  "WarmEnabled": true,
  "WarmType": "ultrawarm1.medium.elasticsearch",
  "WarmCount": 2
},
"EBSOptions": {
  "EBSEnabled": true,
  "VolumeType": "gp2",
  "VolumeSize": 10
},
"AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":"
\": \"Allow\", \"Principal\": {\"AWS\": \"*\"}, \"Action\": \"es:*\", \"Resource\":
\": \"arn:aws:es:us-east-1:123456789012:domain/cli-example-1/*\"}]}",
"SnapshotOptions": {
  "AutomatedSnapshotStartHour": 0
},
"CognitoOptions": {
  "Enabled": false
},
"EncryptionAtRestOptions": {
  "Enabled": true,
  "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/1a2a3a4a-1a2a-1a2a-1a2a-1a2a3a4a5a6a"
},
"NodeToNodeEncryptionOptions": {
  "Enabled": true
},
"AdvancedOptions": {
```

```
        "rest.action.multi.allow_explicit_index": "true"
    },
    "ServiceSoftwareOptions": {
        "CurrentVersion": "R20200522",
        "NewVersion": "",
        "UpdateAvailable": false,
        "Cancelable": false,
        "UpdateStatus": "COMPLETED",
        "Description": "There is no software update available for this
domain.",
        "AutomatedUpdateDate": 0.0
    },
    "DomainEndpointOptions": {
        "EnforceHTTPS": true,
        "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
    },
    "AdvancedSecurityOptions": {
        "Enabled": true,
        "InternalUserDatabaseEnabled": true
    }
},
{
    "DomainId": "123456789012/cli-example-2",
    "DomainName": "cli-example-2",
    "ARN": "arn:aws:es:us-east-1:123456789012:domain/cli-example-2",
    "Created": true,
    "Deleted": false,
    "Processing": true,
    "UpgradeProcessing": false,
    "ElasticsearchVersion": "7.4",
    "ElasticsearchClusterConfig": {
        "InstanceType": "r5.large.elasticsearch",
        "InstanceCount": 1,
        "DedicatedMasterEnabled": false,
        "ZoneAwarenessEnabled": false,
        "WarmEnabled": false
    },
    "EBSOptions": {
        "EBSEnabled": true,
        "VolumeType": "gp2",
        "VolumeSize": 10
    },
}
```

```
    "AccessPolicies": [{"Version": "2012-10-17", "Statement": [{"Effect": "Deny", "Principal": {"AWS": "*"}, "Action": "es:*", "Resource": "arn:aws:es:us-east-1:123456789012:domain/cli-example-2/*"}]}],
    "SnapshotOptions": {
      "AutomatedSnapshotStartHour": 0
    },
    "CognitoOptions": {
      "Enabled": false
    },
    "EncryptionAtRestOptions": {
      "Enabled": false
    },
    "NodeToNodeEncryptionOptions": {
      "Enabled": false
    },
    "AdvancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    },
    "ServiceSoftwareOptions": {
      "CurrentVersion": "",
      "NewVersion": "",
      "UpdateAvailable": false,
      "Cancellable": false,
      "UpdateStatus": "COMPLETED",
      "Description": "There is no software update available for this
domain.",
      "AutomatedUpdateDate": 0.0
    },
    "DomainEndpointOptions": {
      "EnforceHTTPS": false,
      "TLSSecurityPolicy": "Policy-Min-TLS-1-0-2019-07"
    },
    "AdvancedSecurityOptions": {
      "Enabled": false,
      "InternalUserDatabaseEnabled": false
    }
  ]
}
```

詳細については、[Amazon Elasticsearch Service デベロッパーガイド](#)の「[Amazon Elasticsearch Service ドメインの作成と管理](#)」を参照してください。Amazon Elasticsearch Service

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeElasticsearchDomains](#)」の「」を参照してください。

describe-reserved-elasticsearch-instances

次のコード例は、describe-reserved-elasticsearch-instances を使用する方法を示しています。

AWS CLI

すべてのリザーブドインスタンスを表示するには

次のdescribe-elasticsearch-domains例は、リージョンで予約したすべてのインスタンスの概要を示しています。

```
aws es describe-reserved-elasticsearch-instances
```

出力:

```
{
  "ReservedElasticsearchInstances": [{
    "FixedPrice": 100.0,
    "ReservedElasticsearchInstanceOfferingId":
    "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
    "ReservationName": "my-reservation",
    "PaymentOption": "PARTIAL_UPFRONT",
    "UsagePrice": 0.0,
    "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
    "RecurringCharges": [{
      "RecurringChargeAmount": 0.603,
      "RecurringChargeFrequency": "Hourly"
    }],
    "State": "payment-pending",
    "StartTime": 1522872571.229,
    "ElasticsearchInstanceCount": 3,
    "Duration": 31536000,
    "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",
    "CurrencyCode": "USD"
  }]
}
```

詳細については、[Amazon Elasticsearch Service デベロッパーガイド](#)の「[リザーブドインスタンス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReservedElasticsearchInstances](#)」の「」を参照してください。

list-domain-names

次のコード例は、list-domain-names を使用する方法を示しています。

AWS CLI

すべてのドメインを一覧表示するには

次のlist-domain-names例は、リージョン内のすべてのドメインの簡単な概要を示しています。

```
aws es list-domain-names
```

出力:

```
{
  "DomainNames": [{
    "DomainName": "cli-example-1"
  },
  {
    "DomainName": "cli-example-2"
  }
]
```

詳細については、[Amazon Elasticsearch Service デベロッパーガイド](#)の「[Amazon Elasticsearch Service ドメインの作成と管理](#)」を参照してください。 Amazon Elasticsearch Service

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDomainNames](#)」の「」を参照してください。

AWS OpsWorks を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS OpsWorks。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

assign-instance

次のコード例は、assign-instance を使用する方法を示しています。

AWS CLI

登録済みインスタンスをレイヤーに割り当てるには

次の例では、登録されたインスタンスをカスタムレイヤーに割り当てます。

```
aws opsworks --region us-east-1 assign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --layer-ids 26cf1d32-6876-42fa-bbf1-9cadc0bff938
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「登録済みインスタンスのレイヤーへの割り当て」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssignInstance](#)」の「」を参照してください。

assign-volume

次の例は、`assign-volume` を使用する方法を説明しています。

AWS CLI

登録済みボリュームをインスタンスに割り当てるには

次の例では、登録済みの Amazon Elastic Block Store (Amazon EBS) ボリュームをインスタンスに割り当てます。ボリュームはボリューム ID によって識別GUIDされます。ボリューム ID は、Amazon Elastic Compute Cloud (Amazon EC2) ボリューム ID ではなく、スタックにボリュームを登録するときに AWS OpsWorks が割り当てる です。を実行する前に`assign-volume`、まず `mount-volume` を実行してマウントポイントをボリューム`update-volume`に割り当てる必要があります。

```
aws opsworks --region us-east-1 assign-volume --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --volume-id 26cf1d32-6876-42fa-bbf1-9cad0bfff938
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「インスタンスへの Amazon EBS ボリュームの割り当て」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssignVolume](#)」の「」を参照してください。

associate-elastic-ip

次のコード例は、`associate-elastic-ip` を使用する方法を示しています。

AWS CLI

Elastic IP アドレスをインスタンスに関連付けるには

次の例では、Elastic IP アドレスを指定されたインスタンスに関連付けます。

```
aws opsworks --region us-east-1 associate-elastic-ip --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f --elastic-ip 54.148.130.96
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「リソース管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateElasticIp](#)」の「」を参照してください。

attach-elastic-load-balancer

次の例は、attach-elastic-load-balancer を使用する方法を説明しています。

AWS CLI

ロードバランサーをレイヤーにアタッチするには

次の例では、名前で識別されるロードバランサーを指定されたレイヤーにアタッチします。

```
aws opsworks --region us-east-1 attach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」のElastic Load Balancing」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AttachElasticLoadBalancer](#)」の「」を参照してください。

create-app

次のコード例は、create-app を使用する方法を示しています。

AWS CLI

例 1: アプリを作成するには

次の例では、GitHub リポジトリに保存されているコードimplePHPApp から S という名前のPHP アプリケーションを作成します。コマンドは、アプリケーションソース定義の短縮形式を使用します。

```
aws opsworks create-app \
  --region us-east-1 \
  --stack-id f6673d70-32e6-4425-8999-265dd002fec7 \
  --name SimplePHPApp \
  --type php \
  --app-source Type=git,Url=git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git,Revision=version1
```

出力:

```
{
  "AppId": "6cf5163c-a951-444f-a8f7-3716be75f2a2"
}
```

例 2: データベースがアタッチされたアプリケーションを作成するには

次の例では、パブリック S3 バケットの .zip アーカイブに保存されているコードから JSP アプリケーションを作成します。アプリケーションのデータストアとして機能する RDS DB インスタンスをアタッチします。アプリケーションソースとデータベースソースは、コマンドを実行するディレクトリにある別の JSON ファイルで定義されます。

```
aws opsworks create-app \
  --region us-east-1 \
  --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8 \
  --name SimpleJSP \
  --type java \
  --app-source file://appsource.json \
  --data-sources file://datasource.json
```

アプリケーションソース情報は `appsource.json` にあり、以下が含まれています。

```
{
  "Type": "archive",
  "Url": "https://s3.amazonaws.com/opsworks-demo-assets/simplejsp.zip"
}
```

データベースソース情報は `datasource.json` にあり、以下が含まれています。

```
[
  {
```

```
    "Type": "RdsDbInstance",
    "Arn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",
    "DatabaseName": "mydb"
  }
]
```

注: RDS DB インスタンスの場合は、まず `register-rds-db-instance` を使用してインスタンスをスタックに登録する必要があります。MySQL App Server インスタンスの場合は、`Type` を `OpsworksMySQLInstance` に設定します。これらのインスタンスは、AWS OpsWorks によって作成されるため、登録する必要はありません。

出力:

```
{
  "AppId": "26a61ead-d201-47e3-b55c-2a7c666942f8"
}
```

詳細については、AWS OpsWorks 「ユーザーガイド」の「アプリの追加」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateApp](#)」の「」を参照してください。

create-deployment

次のコード例は、`create-deployment` を使用する方法を示しています。

AWS CLI

例 1: アプリケーションをデプロイし、スタックコマンドを実行するには

次の例は、`create-deployment` コマンドを使用してアプリケーションをデプロイし、スタックコマンドを実行する方法を示しています。コマンドを指定する JSON オブジェクトの引用符 (") 文字にはすべてエスケープ文字 (\) が先行することに注意してください。エスケープ文字がないと、コマンドは無効な JSON エラーを返す可能性があります。

次の `create-deployment` 例では、指定されたスタックにアプリケーションをデプロイします。

```
aws opsworks create-deployment \
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \
```

```
--app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb  
--command "{\"Name\":\"deploy\"}"
```

出力:

```
{  
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"  
}
```

例 2: Rails アプリをデプロイしてデータベースを移行するには

次のcreate-deploymentコマンドは、Ruby on Rails アプリケーションを指定されたスタックにデプロイし、データベースを移行します。

```
aws opsworks create-deployment \  
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \  
  --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \  
  --command "{\"Name\":\"deploy\", \"Args\":{\"migrate\":[\"true\"]}]}"
```

出力:

```
{  
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"  
}
```

デプロイの詳細については、AWS OpsWorks [「ユーザーガイド」の「アプリケーションのデプロイ」](#)を参照してください。

例 3: レシピを実行する

次のcreate-deploymentコマンドは、指定されたスタックのインスタンスphpapp::appsetupでカスタムレシピを実行します。

```
aws opsworks create-deployment \  
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \  
  --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":[\"phpapp::appsetup\"]}]}"
```

出力:


```
{
  "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

詳細については、AWS OpsWorks 「[ユーザーガイド](#)」の「[スタックコマンドの実行](#)」を参照してください。

例 4: 依存関係をインストールする

次のcreate-deploymentコマンドは、パッケージや Ruby gem などの依存関係を、指定されたスタックのインスタンスにインストールします。

```
aws opsworks create-deployment \
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \
  --command "{\"Name\": \"install_dependencies\"}"
```

出力:

```
{
  "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

詳細については、AWS OpsWorks 「[ユーザーガイド](#)」の「[スタックコマンドの実行](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateDeployment](#)」の「」を参照してください。

create-instance

次の例は、create-instance を使用する方法を説明しています。

AWS CLI

インスタンスを作成するには

次のcreate-instanceコマンドは、指定されたスタックに myinstance1 という名前の m1.large Amazon Linux インスタンスを作成します。インスタンスは 1 つのレイヤーに割り当てられません。

```
aws opsworks --region us-east-1 create-instance --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --layer-ids 5c8c272a-f2d5-42e3-8245-5bf3927cb65b --hostname myinstance1 --instance-type m1.large --os "Amazon Linux"
```

自動生成された名前を使用するには、`get-hostname-suggestion` を呼び出します。これは `get-hostname-suggestion`、スタックの作成時に指定したテーマに基づいてホスト名を生成します。次に、その名前をホスト名引数に渡します。

出力:

```
{
  "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「レイヤーへのインスタンスの追加」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateInstance](#)」の「」を参照してください。

create-layer

次のコード例は、`create-layer` を使用する方法を示しています。

AWS CLI

レイヤーを作成するには

次の `create-layer` コマンドは、指定されたスタック `yPHPLayer` に `M` という名前の PHP App Server レイヤーを作成します。

```
aws opsworks create-layer --region us-east-1 --stack-id f6673d70-32e6-4425-8999-265dd002fec7 --type php-app --name MyPHPLayer --shortname myphplayer
```

出力:

```
{
```

```
"LayerId": "0b212672-6b4b-40e4-8a34-5a943cf2e07a"
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「レイヤーの作成方法」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateLayer](#)」の「」を参照してください。

create-server

次の例は、create-server を使用する方法を説明しています。

AWS CLI

サーバーを作成するには

次のcreate-server例では、デフォルトのリージョンautomate-06に という名前の新しい Chef Automate サーバーを作成します。デフォルトは、保持するバックアップの数、メンテナンスとバックアップの開始時間など、他のほとんどの設定で使用されます。create-server コマンドを実行する前に、「Opsworks [AWS OpsWorks for Chef Automate ユーザーガイド](#)」の「[Getting Started with](#) for Chef Automate」の前提条件を完了してください。AWS

```
aws opsworks-cm create-server \
  --engine "ChefAutomate" \
  --instance-profile-arn "arn:aws:iam::012345678901:instance-profile/aws-opsworks-
cm-ec2-role" \
  --instance-type "t2.medium" \
  --server-name "automate-06" \
  --service-role-arn "arn:aws:iam::012345678901:role/aws-opsworks-cm-service-role"
```

出力:

```
{
  "Server": {
    "AssociatePublicIpAddress": true,
    "BackupRetentionCount": 10,
    "CreatedAt": 2019-12-29T13:38:47.520Z,
    "DisableAutomatedBackup": FALSE,
```

```
"Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
"Engine": "ChefAutomate",
"EngineAttributes": [
  {
    "Name": "CHEF_AUTOMATE_ADMIN_PASSWORD",
    "Value": "1Example1"
  }
],
"EngineModel": "Single",
"EngineVersion": "2019-08",
"InstanceProfileArn": "arn:aws:iam::012345678901:instance-profile/aws-opsworks-cm-ec2-role",
"InstanceType": "t2.medium",
"PreferredBackupWindow": "Sun:02:00",
"PreferredMaintenanceWindow": "00:00",
"SecurityGroupIds": [ "sg-12345678" ],
"ServerArn": "arn:aws:iam::012345678901:instance/automate-06-1010V4UU2WRM2",
"ServerName": "automate-06",
"ServiceRoleArn": "arn:aws:iam::012345678901:role/aws-opsworks-cm-service-role",
"Status": "CREATING",
"SubnetIds": [ "subnet-12345678" ]
}
}
```

詳細については、AWS OpsWorks 「Chef Automate APIリファレンス」の[CreateServer](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateServer](#)」の「」を参照してください。

create-stack

次のコード例は、create-stack を使用する方法を示しています。

AWS CLI

スタックを作成するには

次のcreate-stackコマンドは、Stack CLI という名前のスタックを作成します。

```
aws opsworks create-stack --name "CLI Stack" --stack-region "us-east-1" --service-role-arn arn:aws:iam::123456789012:role/aws-opsworks-service-role --default-
```

```
instance-profile-arn arn:aws:iam::123456789012:instance-profile/aws-opsworks-ec2-role --region us-east-1
```

`service-role-arn` および `default-instance-profile-arn` パラメータが必要です。通常、最初のスタックを作成するときに `aws opsworks create-stack` を使用します。アカウントの Amazon リソースネーム (ARNs) を取得するには、IAMコンソールに移動し、ナビゲーションパネル `Roles` で `Roles` を選択し、ロールまたはプロファイルを選択し、`Summary` タブを選択します。

出力:

```
{
  "StackId": "f6673d70-32e6-4425-8999-265dd002fec7"
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「新しいスタックを作成する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateStack](#)」の「」を参照してください。

create-user-profile

次のコード例は、`create-user-profile` を使用する方法を示しています。

AWS CLI

ユーザープロファイルを作成するには

Identity AWS and Access Manager (IAM) ユーザーをインポート AWS OpsWorks するには、`aws opsworks create-user-profile` を呼び出し `create-user-profile` でユーザープロファイルを作成します。次の例では、Amazon リソースネーム () で識別される IAM ユーザーのユーザープロファイル `cli-user-test` を作成します。ARN。この例では、ユーザーに `cli-user-test` の SSH ユーザー名を割り当て `myusername`、セルフ管理を有効にします。これにより、ユーザーは SSH パブリックキーを指定できます。

```
aws opsworks --region us-east-1 create-user-profile --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test --ssh-username myusername --allow-self-management
```

出力:

```
{
  "IamUserArn": "arn:aws:iam::123456789102:user/cli-user-test"
}
```

ヒント: このコマンドは、アタッチされたポリシーによって付与されたアクセス許可を持つ IAM ユーザー AWS OpsWorks のみを インポートします。set-permissions コマンドを使用して、スタックごとの AWS OpsWorks アクセス許可を付与できます。

詳細情報

詳細については、AWS OpsWorks ユーザーガイド [AWS OpsWorks の「へのユーザーのインポート」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス CreateUserProfile](#)」の「」を参照してください。

delete-app

次の例は、delete-app を使用する方法を説明しています。

AWS CLI

アプリを削除するには

次の例では、アプリケーション ID で識別される指定されたアプリケーションを削除します。アプリケーション ID を取得するには、AWS OpsWorks コンソールのアプリケーションの詳細ページに移動するか、describe-apps コマンドを実行します。

```
aws opsworks delete-app --region us-east-1 --app-id 577943b9-2ec1-4baf-a7bf-1d347601edc5
```

出力: なし。

詳細情報

詳細については、AWS OpsWorks 「[ユーザーガイド](#)」の「[アプリ](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス DeleteApp](#)」の「」を参照してください。

delete-instance

次のコード例は、delete-instance を使用する方法を示しています。

AWS CLI

インスタンスを削除するには

次のdelete-instance例では、インスタンス ID で識別される指定されたインスタンスを削除します。インスタンス ID をを見つけるには、AWS OpsWorks コンソールでインスタンスの詳細ページを開くか、describe-instances コマンドを実行します。

インスタンスがオンラインの場合は、まず `stop-instance` を呼び出してインスタンスを停止し、インスタンスが停止するまで待つ必要があります。describe-instances を実行してインスタンスのステータスを確認します。

インスタンスの Amazon EBS ボリュームまたは Elastic IP アドレスを削除するには、それぞれ `--delete-volumes` または `--delete-elastic-ip` 引数を追加します。

```
aws opsworks delete-instance \  
  --region us-east-1 \  
  --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

このコマンドでは何も出力されません。

詳細については、AWS OpsWorks 「ユーザーガイド」の [AWS OpsWorks 「インスタンスの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteInstance](#)」の「」を参照してください。

delete-layer

次の例は、delete-layer を使用する方法を説明しています。

AWS CLI

レイヤーを削除するには

次の例では、レイヤー ID で識別される指定されたレイヤーを削除します。レイヤー ID を取得するには、AWS OpsWorks コンソールのレイヤーの詳細ページに移動するか、describe-layers コマンドを実行します。

注：レイヤーを削除する前に、`delete-instance` を使用してレイヤーのすべてのインスタンスを削除する必要があります。

```
aws opsworks delete-layer --region us-east-1 --layer-id a919454e-b816-4598-b29a-5796afb498ed
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の AWS OpsWorks 「インスタンスの削除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteLayer](#)」の「」を参照してください。

delete-stack

次の例は、`delete-stack` を使用する方法を説明しています。

AWS CLI

スタックを削除する

次の例では、スタック ID で識別される指定されたスタックを削除します。スタック ID を取得するには、AWS OpsWorks コンソールでスタック設定をクリックするか、`describe-stacks` コマンドを実行します。

注：レイヤーを削除する前に、`delete-app`、`delete-instance`、および `delete-layer` を使用して、スタックのすべてのアプリケーション、インスタンス、レイヤーを削除する必要があります。

```
aws opsworks delete-stack --region us-east-1 --stack-id 154a9d89-7e9e-433b-8de8-617e53756c84
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks ユーザーガイドの「スタックのシャットダウン」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteStack](#)」の「」を参照してください。

delete-user-profile

次のコード例は、delete-user-profile を使用する方法を示しています。

AWS CLI

ユーザープロフィールを削除し、 から IAM ユーザーを削除するには AWS OpsWorks

次の例では、Amazon リソースネーム (IAM) で識別される、指定された AWS Identity and Access Management () ユーザーのユーザープロフィールを削除しますARN。オペレーションはユーザーを から削除しますが AWS OpsWorks、IAM ユーザーを削除しません。API そのタスクには、IAM コンソールCLI、 、または を使用する必要があります。

```
aws opsworks --region us-east-1 delete-user-profile --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks ユーザーガイド AWS OpsWorks の「へのユーザーのインポート」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUserProfile](#)」の「」を参照してください。

deregister-elastic-ip

次の例は、deregister-elastic-ip を使用する方法を説明しています。

AWS CLI

スタックから Elastic IP アドレスの登録を解除するには

次の例では、IP アドレスによって識別される Elastic IP アドレスをスタックから登録解除します。

```
aws opsworks deregister-elastic-ip --region us-east-1 --elastic-ip 54.148.130.96
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「Elastic IP アドレスの登録解除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterElasticIp](#)」の「」を参照してください。

deregister-instance

次のコード例は、deregister-instance を使用する方法を示しています。

AWS CLI

スタックから登録済みインスタンスの登録を解除するには

次のderegister-instanceコマンドは、登録されたインスタンスをスタックから登録解除します。

```
aws opsworks --region us-east-1 deregister-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「登録済みインスタンスの登録解除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterInstance](#)」の「」を参照してください。

deregister-rds-db-instance

次の例は、deregister-rds-db-instance を使用する方法を説明しています。

AWS CLI

スタックから Amazon RDS DB インスタンスの登録を解除するには

次の例では、によって識別される RDS DB インスタンスをスタックARNから登録解除します。

```
aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn:aws:rds:us-west-2:123456789012:db:clitestdb
```

出力：なし。

詳細情報

詳細については、ASW OpsWorks 「ユーザーガイド」の「Amazon RDS インスタンスの登録解除」を参照してください。

インスタンス ID: clitestdb Master usernames: cliuser Master PWD: some23!pwd DB 名: mydb
aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn:aws:rds:us-west-2:645732743964:db:clitestdb

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterRdsDbInstance](#)」の「」を参照してください。

deregister-volume

次の例は、deregister-volume を使用する方法を説明しています。

AWS CLI

Amazon EBSボリュームの登録を解除するには

次の例では、EBSボリュームをスタックから登録解除します。ボリュームは、ボリューム ID によって識別GUIDされます。ボリューム ID は、ボリューム ID ではなく、スタックにボリュームを登録したときに AWS OpsWorks 割り当てられた EC2 です。

```
aws opsworks deregister-volume --region us-east-1 --volume-id 5c48ef52-3144-4bf5-beaa-fda4deb23d4d
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「Amazon EBS ボリュームの登録解除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterVolume](#)」の「」を参照してください。

describe-apps

次のコード例は、describe-apps を使用する方法を示しています。

AWS CLI

アプリを記述するには

次のdescribe-appsコマンドは、指定されたスタック内のアプリケーションについて説明します。

```
aws opsworks describe-apps \  
  --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a \  
  --region us-east-1
```

出力:

```
{  
  "Apps": [  
    {  
      "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",  
      "AppSource": {  
        "Url": "https://s3-us-west-2.amazonaws.com/opsworks-demo-assets/  
simplejsp.zip",  
        "Type": "archive"  
      },  
      "Name": "SimpleJSP",  
      "EnableSsl": false,  
      "SslConfiguration": {},  
      "AppId": "da1decc1-0dff-43ea-ad7c-bb667cd87c8b",  
      "Attributes": {  
        "RailsEnv": null,  
        "AutoBundleOnDeploy": "true",  
        "DocumentRoot": "ROOT"  
      },  
      "Shortname": "simplejsp",  
      "Type": "other",  
      "CreatedAt": "2013-08-01T21:46:54+00:00"  
    }  
  ]  
}
```

詳細については、AWS OpsWorks 「ユーザーガイド」の「アプリ」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeApps](#)」の「」を参照してください。

describe-commands

次のコード例は、describe-commands を使用する方法を示しています。

AWS CLI

コマンドを記述するには

次のdescribe-commandsコマンドは、指定されたインスタス内のコマンドについて説明します。

```
aws opsworks describe-commands \  
  --instance-id 8c2673b9-3fe5-420d-9cfa-78d875ee7687 \  
  --region us-east-1
```

出力:

```
{  
  "Commands": [  
    {  
      "Status": "successful",  
      "CompletedAt": "2013-07-25T18:57:47+00:00",  
      "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",  
      "DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",  
      "AcknowledgedAt": "2013-07-25T18:57:41+00:00",  
      "LogUrl": "https://s3.amazonaws.com/<bucket-name>/logs/008c1a91-  
ec59-4d51-971d-3adff54b00cc?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE  
&Expires=1375394373&Signature=HkXil6UuNfxTCC37EPQAa462E1E%3D&response-cache-  
control=private&response-content-encoding=gzip&response-content- type=text%2Fplain",  
      "Type": "undeploy",  
      "CommandId": "008c1a91-ec59-4d51-971d-3adff54b00cc",  
      "CreatedAt": "2013-07-25T18:57:34+00:00",  
      "ExitCode": 0  
    },  
    {  
      "Status": "successful",  
      "CompletedAt": "2013-07-25T18:55:40+00:00",  
      "InstanceId": "8c2673b9-3fe5-420d-9cfa-78d875ee7687",  
      "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
```

```
    "AcknowledgedAt": "2013-07-25T18:55:32+00:00",
    "LogUrl": "https://s3.amazonaws.com/<bucket-name>/
logs/899d3d64-0384-47b6-a586-33433aad117c?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Expires=1375394373&Signature=xMsJvtLuUqWmsr8s%2FAjVru0BtRs%3D&response-cache-
control=private&response-content-encoding=gzip&response-content-type=text%2Fplain",
    "Type": "deploy",
    "CommandId": "899d3d64-0384-47b6-a586-33433aad117c",
    "CreatedAt": "2013-07-25T18:55:29+00:00",
    "ExitCode": 0
  }
]
}
```

詳細については、「ユーザーガイド」の AWS OpsWorks 「ライフサイクルイベント」を参照してください。AWS OpsWorks

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeCommands](#)」の「」を参照してください。

describe-deployments

次のコード例は、describe-deployments を使用する方法を示しています。

AWS CLI

デプロイを記述するには

次のdescribe-deploymentsコマンドは、指定されたスタック内のデプロイについて説明します。

```
aws opsworks --region us-east-1 describe-deployments --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

出力:

```
{
  "Deployments": [
    {
      "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
      "Status": "successful",
      "CompletedAt": "2013-07-25T18:57:49+00:00",
```

```
"DeploymentId": "6ed0df4c-9ef7-4812-8dac-d54a05be1029",
"Command": {
  "Args": {},
  "Name": "undeploy"
},
"CreatedAt": "2013-07-25T18:57:34+00:00",
"Duration": 15,
"InstanceIds": [
  "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
  "9e588a25-35b2-4804-bd43-488f85ebe5b7"
]
},
{
  "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
  "Status": "successful",
  "CompletedAt": "2013-07-25T18:56:41+00:00",
  "IamUserArn": "arn:aws:iam::123456789012:user/someuser",
  "DeploymentId": "19d3121e-d949-4ff2-9f9d-94eac087862a",
  "Command": {
    "Args": {},
    "Name": "deploy"
  },
  "InstanceIds": [
    "8c2673b9-3fe5-420d-9cfa-78d875ee7687",
    "9e588a25-35b2-4804-bd43-488f85ebe5b7"
  ],
  "Duration": 72,
  "CreatedAt": "2013-07-25T18:55:29+00:00"
}
]
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「アプリケーションのデプロイ」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDeployments](#)」の「」を参照してください。

describe-elastic-ips

次の例は、describe-elastic-ips を使用する方法を説明しています。

AWS CLI

Elastic IP インスタンスを記述するには

次のdescribe-elastic-ipsコマンドは、指定されたインスタンスの Elastic IP アドレスについて説明します。

```
aws opsworks --region us-east-1 describe-elastic-ips --instance-id b62f3e04-e9eb-436c-a91f-d9e9a396b7b0
```

出力:

```
{
  "ElasticIps": [
    {
      "Ip": "192.0.2.0",
      "Domain": "standard",
      "Region": "us-west-2"
    }
  ]
}
```

詳細情報

詳細については、「AWS OpsWorks ユーザーガイド」の「インスタンス」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeElasticIps](#)」の「」を参照してください。

describe-elastic-load-balancers

次のコード例は、describe-elastic-load-balancers を使用する方法を示しています。

AWS CLI

スタックのエラスティックロードバランサーを記述するには

次のdescribe-elastic-load-balancersコマンドは、指定されたスタックのロードバランサーについて説明します。


```
aws opsworks --region us-west-2 describe-elastic-load-balancers --stack-id 6f4660e5-37a6-4e42-bfa0-1358ebd9c182
```

出力: この特定のスタックには 1 つのロードバランサーがあります。

```
{
  "ElasticLoadBalancers": [
    {
      "SubnetIds": [
        "subnet-60e4ea04",
        "subnet-66e1c110"
      ],
      "Ec2InstanceIds": [],
      "ElasticLoadBalancerName": "my-balancer",
      "Region": "us-west-2",
      "LayerId": "344973cb-bf2b-4cd0-8d93-51cd819bab04",
      "AvailabilityZones": [
        "us-west-2a",
        "us-west-2b"
      ],
      "VpcId": "vpc-b319f9d4",
      "StackId": "6f4660e5-37a6-4e42-bfa0-1358ebd9c182",
      "DnsName": "my-balancer-2094040179.us-west-2.elb.amazonaws.com"
    }
  ]
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「アプリ」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeElasticLoadBalancers](#)」の「」を参照してください。

describe-instances

次のコード例は、describe-instances を使用する方法を示しています。

AWS CLI

インスタンスを記述するには

次のdescribe-instancesコマンドは、指定されたスタック内のインスタンスについて説明します。

```
aws opsworks --region us-east-1 describe-instances --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

出力: 次の出力例は、2つのインスタンスを持つスタックの場合です。1つ目は登録済みEC2インスタンスで、2つ目は によって作成されました AWS OpsWorks。

```
{
  "Instances": [
    {
      "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcdded3fa0f",
      "PrivateDns": "ip-10-31-39-66.us-west-2.compute.internal",
      "LayerIds": [
        "26cf1d32-6876-42fa-bbf1-9cadcd0bff938"
      ],
      "EbsOptimized": false,
      "ReportedOs": {
        "Version": "14.04",
        "Name": "ubuntu",
        "Family": "debian"
      },
      "Status": "online",
      "InstanceId": "4d6d1710-ded9-42a1-b08e-b043ad7af1e2",
      "SshKeyName": "US-West-2",
      "InfrastructureClass": "ec2",
      "RootDeviceVolumeId": "vol-d08ec6c1",
      "SubnetId": "subnet-b8de0ddd",
      "InstanceType": "t1.micro",
      "CreatedAt": "2015-02-24T20:52:49+00:00",
      "AmiId": "ami-35501205",
      "Hostname": "ip-192-0-2-0",
      "Ec2InstanceId": "i-5cd23551",
      "PublicDns": "ec2-192-0-2-0.us-west-2.compute.amazonaws.com",
      "SecurityGroupIds": [
        "sg-c4d3f0a1"
      ],
      "Architecture": "x86_64",
      "RootDeviceType": "ebs",
      "InstallUpdatesOnBoot": true,
      "Os": "Custom",
      "VirtualizationType": "paravirtual",
    }
  ]
}
```

```
    "AvailabilityZone": "us-west-2a",
    "PrivateIp": "10.31.39.66",
    "PublicIp": "192.0.2.06",
    "RegisteredBy": "arn:aws:iam::123456789102:user/AWS/OpsWorks/OpsWorks-
EC2Register-i-5cd23551"
  },
  {
    "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcdded3fa0f",
    "PrivateDns": "ip-10-31-39-158.us-west-2.compute.internal",
    "SshHostRsaKeyFingerprint": "69:6b:7b:8b:72:f3:ed:23:01:00:05:bc:9f:a4:60:c1",
    "LayerIds": [
      "26cf1d32-6876-42fa-bbf1-9cad0bfff938"
    ],
    "EbsOptimized": false,
    "ReportedOs": {},
    "Status": "booting",
    "InstanceId": "9b137a0d-2f5d-4cc0-9704-13da4b31fdcb",
    "SshKeyName": "US-West-2",
    "InfrastructureClass": "ec2",
    "RootDeviceVolumeId": "vol-e09dd5f1",
    "SubnetId": "subnet-b8de0ddd",
    "InstanceProfileArn": "arn:aws:iam::123456789102:instance-profile/aws-
opsworks-ec2-role",
    "InstanceType": "c3.large",
    "CreatedAt": "2015-02-24T21:29:33+00:00",
    "AmiId": "ami-9fc29baf",
    "SshHostDsaKeyFingerprint": "fc:87:95:c3:f5:e1:3b:9f:d2:06:6e:62:9a:35:27:e8",
    "Ec2InstanceId": "i-8d2dca80",
    "PublicDns": "ec2-192-0-2-1.us-west-2.compute.amazonaws.com",
    "SecurityGroupIds": [
      "sg-b022add5",
      "sg-b122add4"
    ],
    "Architecture": "x86_64",
    "RootDeviceType": "ebs",
    "InstallUpdatesOnBoot": true,
    "Os": "Amazon Linux 2014.09",
    "VirtualizationType": "paravirtual",
    "AvailabilityZone": "us-west-2a",
    "Hostname": "custom11",
    "PrivateIp": "10.31.39.158",
    "PublicIp": "192.0.2.0"
  }
]
```

```
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「インスタンス」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstances](#)」の「」を参照してください。

describe-layers

次のコード例は、describe-layers を使用する方法を示しています。

AWS CLI

スタックのレイヤーを記述するには

次のdescribe-layersコマンドは、指定されたスタック内のレイヤーについて説明します。

```
aws opsworks --region us-east-1 describe-layers --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

出力:

```
{
  "Layers": [
    {
      "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
      "Type": "db-master",
      "DefaultSecurityGroupNames": [
        "AWS-OpsWorks-DB-Master-Server"
      ],
      "Name": "MySQL",
      "Packages": [],
      "DefaultRecipes": {
        "Undeploy": [],
        "Setup": [
          "opsworks_initial_setup",
          "ssh_host_keys",
          "ssh_users",
          "mysql::client",
          "dependencies",

```

```
        "ebs",
        "opsworks_ganglia::client",
        "mysql::server",
        "dependencies",
        "deploy::mysql"
    ],
    "Configure": [
        "opsworks_ganglia::configure-client",
        "ssh_users",
        "agent_version",
        "deploy::mysql"
    ],
    "Shutdown": [
        "opsworks_shutdown::default",
        "mysql::stop"
    ],
    "Deploy": [
        "deploy::default",
        "deploy::mysql"
    ]
  ],
  "CustomRecipes": {
    "Undeploy": [],
    "Setup": [],
    "Configure": [],
    "Shutdown": [],
    "Deploy": []
  },
  "EnableAutoHealing": false,
  "LayerId": "41a20847-d594-4325-8447-171821916b73",
  "Attributes": {
    "MySQLRootPasswordUbiquitous": "true",
    "RubygemsVersion": null,
    "RailsStack": null,
    "HaproxyHealthCheckMethod": null,
    "RubyVersion": null,
    "BundlerVersion": null,
    "HaproxyStatsPassword": null,
    "PassengerVersion": null,
    "MemcachedMemory": null,
    "EnableHaproxyStats": null,
    "ManageBundler": null,
    "NodejsVersion": null,
    "HaproxyHealthCheckUrl": null,
```

```
    "MysqlRootPassword": "*****FILTERED*****",
    "GangliaPassword": null,
    "GangliaUser": null,
    "HaproxyStatsUrl": null,
    "GangliaUrl": null,
    "HaproxyStatsUser": null
  },
  "Shortname": "db-master",
  "AutoAssignElasticIps": false,
  "CustomSecurityGroupIds": [],
  "CreatedAt": "2013-07-25T18:11:19+00:00",
  "VolumeConfigurations": [
    {
      "MountPoint": "/vol/mysql",
      "Size": 10,
      "NumberOfDisks": 1
    }
  ]
},
{
  "StackId": "38ee91e2-abdc-4208-a107-0b7168b3cc7a",
  "Type": "custom",
  "DefaultSecurityGroupNames": [
    "AWS-OpsWorks-Custom-Server"
  ],
  "Name": "TomCustom",
  "Packages": [],
  "DefaultRecipes": {
    "Undeploy": [],
    "Setup": [
      "opsworks_initial_setup",
      "ssh_host_keys",
      "ssh_users",
      "mysql::client",
      "dependencies",
      "ebs",
      "opsworks_ganglia::client"
    ],
    "Configure": [
      "opsworks_ganglia::configure-client",
      "ssh_users",
      "agent_version"
    ],
    "Shutdown": [
```

```
        "opsworks_shutdown::default"
    ],
    "Deploy": [
        "deploy::default"
    ]
},
"CustomRecipes": {
    "Undeploy": [],
    "Setup": [
        "tomcat::setup"
    ],
    "Configure": [
        "tomcat::configure"
    ],
    "Shutdown": [],
    "Deploy": [
        "tomcat::deploy"
    ]
},
"EnableAutoHealing": true,
"LayerId": "e6cbcd29-d223-40fc-8243-2eb213377440",
"Attributes": {
    "MysqlRootPasswordUbiquitous": null,
    "RubygemsVersion": null,
    "RailsStack": null,
    "HaproxyHealthCheckMethod": null,
    "RubyVersion": null,
    "BundlerVersion": null,
    "HaproxyStatsPassword": null,
    "PassengerVersion": null,
    "MemcachedMemory": null,
    "EnableHaproxyStats": null,
    "ManageBundler": null,
    "NodejsVersion": null,
    "HaproxyHealthCheckUrl": null,
    "MysqlRootPassword": null,
    "GangliaPassword": null,
    "GangliaUser": null,
    "HaproxyStatsUrl": null,
    "GangliaUrl": null,
    "HaproxyStatsUser": null
},
"Shortname": "tomcustom",
"AutoAssignElasticIps": false,
```

```
    "CustomSecurityGroupIds": [],
    "CreatedAt": "2013-07-25T18:12:53+00:00",
    "VolumeConfigurations": []
  }
]
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「レイヤー」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLayers](#)」の「」を参照してください。

describe-load-based-auto-scaling

次の例は、describe-load-based-auto-scaling を使用する方法を説明しています。

AWS CLI

レイヤーの負荷ベースのスケールリング設定を記述するには

次の例では、指定されたレイヤーの負荷ベースのスケールリング設定について説明します。レイヤーはレイヤー ID で識別されます。レイヤー ID は、レイヤーの詳細ページまたは `aws opsworks describe-layers` を実行して確認できます。

```
aws opsworks describe-load-based-auto-scaling --region us-east-1 --layer-ids 6bec29c9-c866-41a0-aba5-fa3e374ce2a1
```

出力：サンプルレイヤーには 1 つのロードベースのインスタンスがあります。

```
{
  "LoadBasedAutoScalingConfigurations": [
    {
      "DownScaling": {
        "IgnoreMetricsTime": 10,
        "ThresholdsWaitTime": 10,
        "InstanceCount": 1,
        "CpuThreshold": 30.0
      },
      "Enable": true,
      "UpScaling": {
        "IgnoreMetricsTime": 5,
```



```
    "ThresholdsWaitTime": 5,  
    "InstanceCount": 1,  
    "CpuThreshold": 80.0  
  },  
  "LayerId": "6bec29c9-c866-41a0-aba5-fa3e374ce2a1"  
}  
]  
}
```

詳細情報

詳細については、AWS OpsWorks ユーザーガイドの「自動ロードベースのスケージングの仕組み」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeLoadBasedAutoScaling](#)」の「」を参照してください。

describe-my-user-profile

次の例は、describe-my-user-profile を使用する方法を説明しています。

AWS CLI

ユーザーのプロファイルを取得するには

次の例は、コマンドを実行している AWS Identity and Access Management (IAM) ユーザーのプロファイルを取得する方法を示しています。

```
aws opsworks --region us-east-1 describe-my-user-profile
```

出力: 簡潔にするために、ユーザーのSSHパブリックキーのほとんどは省略記号 (...) に置き換えられます。

```
{  
  "UserProfile": {  
    "IamUserArn": "arn:aws:iam::123456789012:user/myusername",  
    "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQ...3LQ4aX9jpxQw== rsa-  
key-20141104",  
    "Name": "myusername",  
    "SshUsername": "myusername"  
  }  
}
```

詳細情報

詳細については、AWS OpsWorks ユーザーガイド [AWS OpsWorks の「へのユーザーのインポート」](#) を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeMyUserProfile](#)」の「」を参照してください。

describe-permissions

次のコード例は、describe-permissions を使用する方法を示しています。

AWS CLI

ユーザーのスタックごとの AWS OpsWorks アクセス許可レベルを取得するには

次の例は、指定されたスタックで AWS Identity and Access Management (IAM) ユーザーのアクセス許可レベルを取得する方法を示しています。

```
aws opsworks --region us-east-1 describe-permissions --iam-user-arn arn:aws:iam::123456789012:user/cli-user-test --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

出力:

```
{
  "Permissions": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "Level": "manage",
      "AllowSudo": true,
      "AllowSsh": true
    }
  ]
}
```

詳細情報

詳細については、AWS OpsWorks 「[ユーザーガイド](#)」の「[スタックあたりのアクセス許可レベルの付与](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePermissions](#)」の「」を参照してください。

describe-raid-arrays

次の例は、describe-raid-arrays を使用する方法を説明しています。

AWS CLI

RAID配列を記述するには

次の例では、指定されたスタックのインスタンスにアタッチされたRAID配列について説明します。

```
aws opsworks --region us-east-1 describe-raid-arrays --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

出力：1つのRAID配列を持つスタックの出力を次に示します。

```
{
  "RaidArrays": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "AvailabilityZone": "us-west-2a",
      "Name": "Created for php-app1",
      "NumberOfDisks": 2,
      "InstanceId": "9f14adbc-ced5-43b6-bf01-e7d0db6cf2f7",
      "RaidLevel": 0,
      "VolumeType": "standard",
      "RaidArrayId": "f2d4e470-5972-4676-b1b8-bae41ec3e51c",
      "Device": "/dev/md0",
      "MountPoint": "/mnt/workspace",
      "CreatedAt": "2015-02-26T23:53:09+00:00",
      "Size": 100
    }
  ]
}
```

詳細については、AWS OpsWorks 「ユーザーガイド」のEBS「ボリューム」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRaidArrays](#)」の「」を参照してください。

describe-rds-db-instances

次のコード例は、describe-rds-db-instances を使用する方法を示しています。

AWS CLI

スタックの登録済み Amazon RDSインスタンスを記述するには

次の例では、指定されたスタックに登録された Amazon RDSインスタンスについて説明します。

```
aws opsworks --region us-east-1 describe-rds-db-instances --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

出力：1つの登録済みRDSインスタンスを持つスタックの出力を次に示します。

```
{
  "RdsDbInstances": [
    {
      "Engine": "mysql",
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "MissingOnRds": false,
      "Region": "us-west-2",
      "RdsDbInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",
      "DbPassword": "*****FILTERED*****",
      "Address": "clitestdb.cd1q1k5uwd0k.us-west-2.rds.amazonaws.com",
      "DbUser": "cliuser",
      "DbInstanceIdentifier": "clitestdb"
    }
  ]
}
```

詳細については、AWS OpsWorks 「ユーザーガイド」の「リソース管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRdsDbInstances](#)」の「」を参照してください。

describe-stack-provisioning-parameters

次の例は、describe-stack-provisioning-parameters を使用する方法を説明しています。

AWS CLI

スタックのプロビジョニングパラメータを返すには

次のdescribe-stack-provisioning-parameters例では、指定されたスタックのプロビジョニングパラメータを返します。プロビジョニングパラメータには、ガスタック内のインスタンスでエージェントを管理するために OpsWorks 使用するエージェントのインストール場所やパブリックキーなどの設定が含まれます。

```
aws opsworks describe-stack-provisioning-parameters \  
  --stack-id 62744d97-6faf-4ecb-969b-a086fEXAMPLE
```

出力:

```
{  
  "AgentInstallerUrl": "https://opsworks-instance-agent-us-west-2.s3.amazonaws.com/ID_number/opsworks-agent-installer.tgz",  
  "Parameters": {  
    "agent_installer_base_url": "https://opsworks-instance-agent-us-west-2.s3.amazonaws.com",  
    "agent_installer_tgz": "opsworks-agent-installer.tgz",  
    "assets_download_bucket": "opsworks-instance-assets-us-west-2.s3.amazonaws.com",  
    "charlie_public_key": "-----BEGIN PUBLIC KEY-----PUBLIC_KEY_EXAMPLE\n-----  
END PUBLIC KEY-----",  
    "instance_service_endpoint": "opsworks-instance-service.us-west-2.amazonaws.com",  
    "instance_service_port": "443",  
    "instance_service_region": "us-west-2",  
    "instance_service_ssl_verify_peer": "true",  
    "instance_service_use_ssl": "true",  
    "ops_works_endpoint": "opsworks.us-west-2.amazonaws.com",  
    "ops_works_port": "443",  
    "ops_works_region": "us-west-2",  
    "ops_works_ssl_verify_peer": "true",  
    "ops_works_use_ssl": "true",  
    "verbose": "false",  
    "wait_between_runs": "30"  
  }  
}
```

```
}
```

詳細については、AWS OpsWorks 「ユーザーガイド」の「[スタックコマンドの実行](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStackProvisioningParameters](#)」の「」を参照してください。

describe-stack-summary

次の例は、describe-stack-summary を使用する方法を説明しています。

AWS CLI

スタックの設定を記述するには

次のdescribe-stack-summary コマンドは、指定されたスタックの設定の概要を返します。

```
aws opsworks --region us-east-1 describe-stack-summary --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

出力:

```
{
  "StackSummary": {
    "StackId": "8c428b08-a1a1-46ce-a5f8-feddc43771b8",
    "InstancesCount": {
      "Booting": 1
    },
    "Name": "CLITest",
    "AppsCount": 1,
    "LayersCount": 1,
    "Arn": "arn:aws:opsworks:us-west-2:123456789012:stack/8c428b08-a1a1-46ce-a5f8-feddc43771b8/"
  }
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「[スタック](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeStackSummary](#)」の「」を参照してください。

describe-stacks

次の例は、describe-stacks を使用する方法を説明しています。

AWS CLI

スタックを記述するには

次のdescribe-stacksコマンドは、アカウントのスタックについて説明します。

```
aws opsworks --region us-east-1 describe-stacks
```

出力:

```
{
  "Stacks": [
    {
      "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
      "StackId": "aeb7523e-7c8b-49d4-b866-03aae9d4fbcf",
      "DefaultRootDeviceType": "instance-store",
      "Name": "TomStack-sd",
      "ConfigurationManager": {
        "Version": "11.4",
        "Name": "Chef"
      },
      "UseCustomCookbooks": true,
      "CustomJson": "{\n  \"tomcat\": {\n    \"base_version\": 7,\n    \"java_opts\n\": \"-Djava.awt.headless=true -Xmx256m\"\n  },\n  \"datasources\": {\n    \"ROOT\":\n  \"jdbc/mydb\"\n  }\n}",
      "Region": "us-east-1",
      "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/aws-opsworks-ec2-role",
      "CustomCookbooksSource": {
        "Url": "git://github.com/example-repo/tomcustom.git",
        "Type": "git"
      },
      "DefaultAvailabilityZone": "us-east-1a",
      "HostnameTheme": "Layer_Dependent",
      "Attributes": {
        "Color": "rgb(45, 114, 184)"
      },
      "DefaultOs": "Amazon Linux",
      "CreatedAt": "2013-08-01T22:53:42+00:00"
    },
  ],
}
```

```
{
  "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
  "StackId": "40738975-da59-4c5b-9789-3e422f2cf099",
  "DefaultRootDeviceType": "instance-store",
  "Name": "MyStack",
  "ConfigurationManager": {
    "Version": "11.4",
    "Name": "Chef"
  },
  "UseCustomCookbooks": false,
  "Region": "us-east-1",
  "DefaultInstanceProfileArn": "arn:aws:iam::444455556666:instance-profile/aws-opsworks-ec2-role",
  "CustomCookbooksSource": {},
  "DefaultAvailabilityZone": "us-east-1a",
  "HostnameTheme": "Layer_Dependent",
  "Attributes": {
    "Color": "rgb(45, 114, 184)"
  },
  "DefaultOs": "Amazon Linux",
  "CreatedAt": "2013-10-25T19:24:30+00:00"
}
]
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「スタック」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStacks](#)」の「」を参照してください。

describe-timebased-auto-scaling

次の例は、describe-timebased-auto-scaling を使用する方法を説明しています。

AWS CLI

インスタンスの時間ベースのスケーリング設定を記述するには

次の例では、指定されたインスタンスの時間ベースのスケーリング設定について説明します。インスタンスは、インスタンス ID によって識別されます。インスタンス ID は、インスタンスの詳細ページまたは を実行して確認できます describe-instances。


```
aws opsworks describe-time-based-auto-scaling --region us-east-1 --instance-ids 701f2ffe-5d8e-4187-b140-77b75f55de8d
```

出力: この例では、1 つの時間ベースのインスタンスがあります。

```
{
  "TimeBasedAutoScalingConfigurations": [
    {
      "InstanceId": "701f2ffe-5d8e-4187-b140-77b75f55de8d",
      "AutoScalingSchedule": {
        "Monday": {
          "11": "on",
          "10": "on",
          "13": "on",
          "12": "on"
        },
        "Tuesday": {
          "11": "on",
          "10": "on",
          "13": "on",
          "12": "on"
        }
      }
    }
  ]
}
```

詳細情報

詳細については、AWS OpsWorks ユーザーガイドの「自動時間ベースのスケーリングの仕組み」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTimebasedAutoScaling](#)」の「」を参照してください。

describe-user-profiles

次の例は、describe-user-profiles を使用する方法を説明しています。

AWS CLI

ユーザープロフィールを記述するには

次のdescribe-user-profilesコマンドは、アカウントのユーザープロフィールについて説明します。

```
aws opsworks --region us-east-1 describe-user-profiles
```

出力:

```
{
  "UserProfiles": [
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/someuser",
      "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEak0uP7i80q3Cko...",
      "AllowSelfManagement": true,
      "Name": "someuser",
      "SshUsername": "someuser"
    },
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "AllowSelfManagement": true,
      "Name": "cli-user-test",
      "SshUsername": "myusername"
    }
  ]
}
```

詳細情報

詳細については、「ユーザーガイド」の AWS OpsWorks 「ユーザーの管理」を参照してください。AWS OpsWorks

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUserProfiles](#)」の「」を参照してください。

describe-volumes

次の例は、describe-volumes を使用する方法を説明しています。

AWS CLI

スタックのボリュームを記述するには

次の例では、スタックのEBSボリュームについて説明します。

```
aws opsworks --region us-east-1 describe-volumes --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

出力:

```
{
  "Volumes": [
    {
      "Status": "in-use",
      "AvailabilityZone": "us-west-2a",
      "Name": "CLITest",
      "InstanceId": "dfe18b02-5327-493d-91a4-c5c0c448927f",
      "VolumeType": "standard",
      "VolumeId": "56b66fbd-e1a1-4aff-9227-70f77118d4c5",
      "Device": "/dev/sdi",
      "Ec2VolumeId": "vol-295c1638",
      "MountPoint": "/mnt/myvolume",
      "Size": 1
    }
  ]
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「リソース管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVolumes](#)」の「」を参照してください。

detach-elastic-load-balancer

次の例は、detach-elastic-load-balancer を使用する方法を説明しています。

AWS CLI

ロードバランサーをレイヤーからデタッチするには

次の例では、名前で識別されるロードバランサーをレイヤーからデタッチします。

```
aws opsworks --region us-east-1 detach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」のElastic Load Balancing」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachElasticLoadBalancer](#)」の「」を参照してください。

disassociate-elastic-ip

次のコード例は、disassociate-elastic-ip を使用する方法を示しています。

AWS CLI

インスタンスから Elastic IP アドレスの関連付けを解除するには

次の例では、指定されたインスタンスから Elastic IP アドレスの関連付けを解除します。

```
aws opsworks --region us-east-1 disassociate-elastic-ip --elastic-ip 54.148.130.96
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「リソース管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateElasticIp](#)」の「」を参照してください。

get-hostname-suggestion

次のコード例は、get-hostname-suggestion を使用する方法を示しています。

AWS CLI

レイヤーの次のホスト名を取得するには

次の例では、指定されたレイヤーに対して次に生成されたホスト名を取得します。この例に使用されるレイヤーは、1つのインスタンスを持つ Java Application Server レイヤーです。スタックのホスト名テーマは、デフォルトの Layer_Dependent です。

```
aws opsworks --region us-east-1 get-hostname-suggestion --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

出力:

```
{  
  "Hostname": "java-app2",  
  "LayerId": "888c5645-09a5-4d0e-95a8-812ef1db76a4"  
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「新しいスタックを作成する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetHostnameSuggestion](#)」の「」を参照してください。

reboot-instance

次のコード例は、reboot-instance を使用する方法を示しています。

AWS CLI

インスタンスを再起動するには

次の例では、インスタンスを再起動します。

```
aws opsworks --region us-east-1 reboot-instance --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f
```

出力: なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「インスタンスの再起動」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootInstance](#)」の「」を参照してください。

register-elastic-ip

次のコード例は、register-elastic-ip を使用する方法を示しています。

AWS CLI

Elastic IP アドレスをスタックに登録するには

次の例では、IP アドレスで識別される Elastic IP アドレスを指定されたスタックに登録します。

注：Elastic IP アドレスは、スタックと同じリージョンにある必要があります。

```
aws opsworks register-elastic-ip --region us-east-1 --stack-  
id d72553d4-8727-448c-9b00-f024f0ba1b06 --elastic-ip 54.148.130.96
```

出力

```
{  
  "ElasticIp": "54.148.130.96"  
}
```

詳細情報

詳細については、OpsWorks 「ユーザーガイド」の「スタックでの Elastic IP アドレスの登録」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterElasticIp](#)」の「」を参照してください。

register-rds-db-instance

次のコード例は、register-rds-db-instance を使用する方法を示しています。

AWS CLI

Amazon RDS インスタンスをスタックに登録するには

次の例では、Amazon リソースネーム (ARN) で識別される Amazon RDS DB インスタンスを指定されたスタックに登録します。また、インスタンスのマスターユーザー名とパスワードも指定します。AWS OpsWorks はこれらの値のいずれも検証しません。どちらかが正しくない場合、アプリケーションはデータベースに接続できません。

```
aws opsworks register-rds-db-instance --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --rds-db-instance-arn arn:aws:rds:us-west-2:123456789012:db:clitestdb --db-user cliuser --db-password some23!pwd
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「スタックでの Amazon RDS インスタンスの登録」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterRdsDbInstance](#)」の「」を参照してください。

register-volume

次の例は、register-volume を使用する方法を説明しています。

AWS CLI

Amazon EBSボリュームをスタックに登録するには

次の例では、EBSボリューム ID で識別される Amazon ボリュームを指定されたスタックに登録します。

```
aws opsworks register-volume --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --ec-2-volume-id vol-295c1638
```

出力:

```
{  
  "VolumeId": "ee08039c-7cb7-469f-be10-40fb7f0c05e8"  
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「Amazon EBS ボリュームをスタックに登録する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterVolume](#)」の「」を参照してください。

register

次のコード例は、register を使用する方法を示しています。

AWS CLI

スタックにインスタンスを登録するには

次の例は、AWS Opsworks の外部で作成されたスタックにインスタンスを登録するさまざまな方法を示しています。登録するインスタンスregisterから、または別のワークステーションから実行できます。詳細については、AWS OpsWorks 「ユーザーガイド」の「Amazon インスタンスEC2とオンプレミスインスタンスの登録」を参照してください。

注: 簡潔にするために、例では region 引数を省略します。

Amazon EC2インスタンスを登録するには

EC2 インスタンスを登録することを示すには、引--infrastructure-class数を に設定しますec2。

次の例では、別のワークステーションから指定されたスタックにEC2インスタンスを登録します。インスタンスは、その EC2 ID である によって識別されますi-12345678。この例では、ワークステーションのデフォルトのSSHユーザー名を使用し、デフォルトのプライベートSSHキーなど、パスワードを必要としない認証手法を使用してインスタンスにログインしようとしません。これが失敗した場合、 はパスワードをregisterクエリします。

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb i-12345678
```

次の例では、別のワークステーションから指定されたスタックにEC2インスタンスを登録します。--ssh-username および --ssh-private-key引数を使用して、コマンドがインスタンスへのログインに使用するSSHユーザー名とプライベートキーファイルを明示的に指定します。ec2-userは Amazon Linux インスタンスの標準ユーザー名です。Ubuntu インスタンスubuntuで を使用します。

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --ssh-username ec2-user --ssh-private-key ssh_private_key i-12345678
```


次の例では、register コマンドを実行しているEC2インスタンスを登録します。でインスタンスにログインSSHし、インスタンス ID またはホスト名の代わりに --local引register数で を実行します。

```
aws opsworks register --infrastructure-class ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

オンプレミスインスタンスを登録するには

オンプレミスインスタンスを登録することを示すには、--infrastructure-class引数を に設定しますon-premises。

次の例では、既存のオンプレミスインスタンスを別のワークステーションから指定されたスタックに登録します。インスタンスは IP アドレス によって識別されます192.0.2.3。この例では、ワークステーションのデフォルトのSSHユーザー名を使用し、デフォルトのプライベートSSHキーなどのパスワードを必要としない認証手法を使用してインスタンスにログインしようとしません。これが失敗した場合、 はパスワードをregisterクエリします。

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb 192.0.2.3
```

次の例では、オンプレミスインスタンスを別のワークステーションから指定されたスタックに登録します。インスタンスはホスト名 で識別されますhost1。--override-... 引数は、 をホスト名webserver1として、 192.0.2.3 をインスタンスのパブリック IP アドレスとプライベート IP アドレス10.0.0.2としてそれぞれ表示するように指示 AWS OpsWorksします。

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --override-hostname webserver1 --override-public-ip 192.0.2.3 --override-private-ip 10.0.0.2 host1
```

次の例では、オンプレミスインスタンスを別のワークステーションから指定されたスタックに登録します。インスタンスは IP アドレスによって識別されます。 は、指定されたSSHユーザー名とプライベートキーファイルを使用してインスタンスregisterにログインします。

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --ssh-username admin --ssh-private-key ssh_private_key 192.0.2.3
```

次の例では、既存のオンプレミスインスタンスを別のワークステーションから指定されたスタックに登録します。コマンドは、SSHパスワードとインスタンスの IP アドレスを指定するカスタム SSH コマンド文字列を使用してインスタンスにログインします。

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --override-ssh "sshpass -p 'mypassword' ssh your-user@192.0.2.3"
```

次の例では、register コマンドを実行しているオンプレミスインスタンスに登録します。でインスタンスにログインSSHし、インスタンス ID またはホスト名の代わりに --local 引 register 数で を実行します。

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

出力: EC2 インスタンスに登録するための一般的な出力は次のとおりです。

```
Warning: Permanently added '52.11.41.206' (ECDSA) to the list of known hosts.
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100 6403k  100 6403k    0     0 2121k      0  0:00:03  0:00:03 --:--:-- 2121k
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Initializing AWS OpsWorks
environment
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Running on Ubuntu
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Checking if OS is supported
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Running on supported OS
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Setup motd
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing: ln -sf --backup /etc/
motd.opsworks-static /etc/motd
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Enabling multiverse repositories
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Customizing APT environment
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Installing system packages
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing: dpkg --configure -a
[Tue, 24 Feb 2015 20:48:37 +0000] opsworks-init: Executing with retry: apt-get
update
[Tue, 24 Feb 2015 20:49:13 +0000] opsworks-init: Executing: apt-get install -y ruby
ruby-dev libicu-dev libssl-dev libxslt-dev libxml2-dev libyaml-dev monit
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Using assets bucket from
environment: 'opsworks-instance-assets-us-east-1.s3.amazonaws.com'.
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Installing Ruby for the agent
```

```
[Tue, 24 Feb 2015 20:50:13 +0000] opsworks-init: Executing: /tmp/opsworks-agent-installer.YgGq8wF3UUre6yDy/opsworks-agent-installer/opsworks-agent/bin/installer_wrapper.sh -r -R opsworks-instance-assets-us-east-1.s3.amazonaws.com
[Tue, 24 Feb 2015 20:50:44 +0000] opsworks-init: Starting the installer
Instance successfully registered. Instance ID: 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
Connection to 52.11.41.206 closed.
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の AWS OpsWorks 「スタックでのインスタンスの登録」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[登録](#)」を参照してください。AWS CLI

set-load-based-auto-scaling

次の例は、set-load-based-auto-scaling を使用する方法を説明しています。

AWS CLI

レイヤーの負荷ベースのスケーリング設定を設定するには

次の例では、指定されたレイヤーの負荷ベースのスケーリングを有効にし、そのレイヤーの設定を設定します。create-instance レイヤーにロードベースのインスタンスを追加するには、を使用する必要があります。

```
aws opsworks --region us-east-1 set-load-based-auto-scaling --layer-id 523569ae-2faf-47ac-b39e-f4c4b381f36d --enable --up-scaling file://upscale.json --down-scaling file://downscale.json
```

この例では、アップスケーリングのしきい値設定を、という名前の作業ディレクトリの別のファイルに配置します。これにはupscale.json、以下が含まれます。

```
{
  "InstanceCount": 2,
  "ThresholdsWaitTime": 3,
  "IgnoreMetricsTime": 3,
  "CpuThreshold": 85,
  "MemoryThreshold": 85,
  "LoadThreshold": 85
}
```

この例では、ダウンスケーリングしきい値設定を という名前の作業ディレクトリの別のファイルに配置します。これには `downscale.json`、以下が含まれます。

```
{
  "InstanceCount": 2,
  "ThresholdsWaitTime": 3,
  "IgnoreMetricsTime": 3,
  "CpuThreshold": 35,
  "MemoryThreshold": 30,
  "LoadThreshold": 30
}
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks ユーザーガイドの「自動ロードベースのスケージングの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetLoadBasedAutoScaling](#)」の「」を参照してください。

set-permission

次の例は、`set-permission` を使用する方法を説明しています。

AWS CLI

スタックごとの AWS OpsWorks アクセス許可レベルを付与するには

を呼び出す AWS OpsWorks ことで AWS Identity and Access Management (IAM) ユーザーをにインポートする場合 `create-user-profile`、ユーザーには、アタッチされた IAM ポリシーによって付与されたアクセス許可のみが付与されます。ユーザーのポリシーを変更することで、AWS OpsWorks アクセス許可を付与できます。ただし、多くの場合、ユーザーをインポートし、`set-permission` コマンドを使用して、ユーザーがアクセスする必要があるスタックごとに標準のアクセス許可レベルの 1 つをユーザーに付与する方が簡単です。

次の例では、Amazon リソースネーム () で識別されるユーザーの指定されたスタックに対するアクセス許可を付与しますARN。この例では、スタックのインスタンスに対する `sudo` と `SSH` 権限を持つアクセス許可の管理レベルをユーザーに付与します。

```
aws opsworks set-permission --region us-east-1 --stack-id 71c7ca72-55ae-4b6a-8ee1-a8dcded3fa0f --level manage --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test --allow-ssh --allow-sudo
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「スタックあたりのアクセス許可を AWS OpsWorks ユーザーに付与する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetPermission](#)」の「」を参照してください。

set-time-based-auto-scaling

次の例は、set-time-based-auto-scaling を使用する方法を説明しています。

AWS CLI

レイヤーの時間ベースのスケーリング設定を設定するには

次の例では、指定されたインスタンスの時間ベースの設定を設定します。最初に create-instance を使用して、レイヤーにインスタンスを追加する必要があります。

```
aws opsworks --region us-east-1 set-time-based-auto-scaling --instance-id 69b6237c-08c0-4edb-a6af-78f3d01cedf2 --auto-scaling-schedule file://schedule.json
```

この例では、という名前の作業ディレクトリ内の別のファイルにスケジュールを配置します schedule.json。この例では、インスタンスは月曜日と火曜日の正午 UTC (協定世界時) 前後に数時間オンになっています。

```
{
  "Monday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  },
  "Tuesday": {
```

```
"10": "on",
"11": "on",
"12": "on",
"13": "on"
}
}
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks ユーザーガイドの「自動時間ベースのスケーリングの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetTimeBasedAutoScaling](#)」の「」を参照してください。

start-instance

次の例は、start-instance を使用する方法を説明しています。

AWS CLI

インスタンスを開始するには

次のstart-instanceコマンドは、指定された 24 時間 365 日インスタンスを開始します。

```
aws opsworks start-instance --instance-id f705ee48-9000-4890-8bd3-20eb05825aaf
```

出力：なし。describe-instances を使用して、インスタンスのステータスを確認します。

ヒント start-stack を呼び出すことで、1 つのコマンドでスタック内のすべてのオフラインインスタンスを開始できます。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「24/7 インスタンスの手動起動、停止、再起動」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartInstance](#)」の「」を参照してください。

start-stack

次の例は、start-stack を使用する方法を説明しています。

AWS CLI

スタックのインスタンスを開始するには

次の例では、スタックのすべての 24/7 インスタンスを起動します。特定のインスタンスを開始するには、を使用しますstart-instance。

```
aws opsworks --region us-east-1 start-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「インスタンスの開始」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartStack](#)」の「」を参照してください。

stop-instance

次の例は、stop-instance を使用する方法を説明しています。

AWS CLI

インスタンスを停止するには

次の例では、指定されたインスタンスを停止し、インスタンス ID で識別されます。インスタンス ID を取得するには、AWS OpsWorks コンソールのインスタンスの詳細ページに移動するか、describe-instances コマンドを実行します。

```
aws opsworks stop-instance --region us-east-1 --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

停止したインスタンスを再起動するには、を呼び出すstart-instanceか、を呼び出すことでインスタンスを削除しますdelete-instance。

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「インスタンスの停止」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopInstance](#)」の「」を参照してください。

stop-stack

次の例は、stop-stack を使用する方法を説明しています。

AWS CLI

スタックのインスタンスを停止するには

次の例では、スタックの 24/7 インスタンスをすべて停止しています。特定のインスタンスを停止するには、`stop-instance` を使用します。

```
aws opsworks --region us-east-1 stop-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

出力：出力なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「インスタンスの停止」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopStack](#)」の「」を参照してください。

unassign-instance

次のコード例は、unassign-instance を使用する方法を示しています。

AWS CLI

レイヤーから登録済みインスタンスの割り当てを解除するには

次のunassign-instanceコマンドは、アタッチされたレイヤーからインスタンスの割り当てを解除します。

```
aws opsworks --region us-east-1 unassign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks ユーザーガイドの「登録済みインスタンスの割り当て解除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UnassignInstance](#)」の「」を参照してください。

unassign-volume

次の例は、unassign-volume を使用する方法を説明しています。

AWS CLI

インスタンスからボリュームの割り当てを解除するには

次の例では、登録済みの Amazon Elastic Block Store (Amazon EBS) ボリュームをインスタンスから割り当て解除します。ボリュームはボリューム ID によって識別GUIDされます。ボリューム ID は、Amazon Elastic Compute Cloud (Amazon EC2) ボリューム ID ではなく、スタックにボリュームを登録するときに AWS OpsWorks が割り当てる です。

```
aws opsworks --region us-east-1 unassign-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「Amazon EBS ボリュームの割り当て解除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UnassignVolume](#)」の「」を参照してください。

update-app

次の例は、update-app を使用する方法を説明しています。

AWS CLI

アプリを更新するには

次の例では、指定されたアプリを更新して名前を変更します。

```
aws opsworks --region us-east-1 update-app --app-id 26a61ead-d201-47e3-b55c-2a7c666942f8 --name NewAppName
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「アプリの編集」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateApp](#)」の「」を参照してください。

update-elastic-ip

次のコード例は、update-elastic-ip を使用する方法を示しています。

AWS CLI

Elastic IP アドレス名を更新するには

次の例では、指定された Elastic IP アドレスの名前を更新します。

```
aws opsworks --region us-east-1 update-elastic-ip --elastic-ip 54.148.130.96 --name NewIPName
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「リソース管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateElasticIp](#)」の「」を参照してください。

update-instance

次の例は、update-instance を使用する方法を説明しています。

AWS CLI

インスタンスを更新するには

次の例では、指定されたインスタンスのタイプを更新します。

```
aws opsworks --region us-east-1 update-instance --instance-id dfc18b02-5327-493d-91a4-c5c0c448927f --instance-type c3.xlarge
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「インスタンス設定の編集」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateInstance](#)」の「」を参照してください。

update-layer

次の例は、update-layer を使用する方法を説明しています。

AWS CLI

レイヤーを更新するには

次の例では、Amazon EBS最適化インスタンスを使用するように指定されたレイヤーを更新します。

```
aws opsworks --region us-east-1 update-layer --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4 --use-efs-optimized-instances
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の OpsWorks 「レイヤーの設定の編集」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLayer](#)」の「」を参照してください。

update-my-user-profile

次のコード例は、update-my-user-profile を使用する方法を示しています。

AWS CLI

ユーザーのプロファイルを更新するには

次の例では、指定されたSSHパブリックキーを使用するようにdevelopmentユーザーのプロファイルを更新します。ユーザーのAWS認証情報は、credentials ファイル (~\.aws\credentials) 内のdevelopmentプロファイルによって表され、キーは作業ディレクトリ内の.pem ファイルにあります。

```
aws opsworks --region us-east-1 --profile development update-my-user-profile --ssh-public-key file://development_key.pem
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「AWS OpsWorks ユーザーガイド」の「ユーザー設定の編集」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMyUserProfile](#)」の「」を参照してください。

update-rds-db-instance

次の例は、update-rds-db-instance を使用する方法を説明しています。

AWS CLI

登録済みの Amazon RDS DB インスタンスを更新するには

次の例では、Amazon RDSインスタンスのマスターパスワード値を更新します。このコマンドは、RDSインスタンスのマスターパスワードを変更せず、に指定したパスワードのみを変更することに注意してください AWS OpsWorks。このパスワードがRDSインスタンスのパスワードと一致しない場合、アプリケーションはデータベースに接続できません。

```
aws opsworks --region us-east-1 update-rds-db-instance --db-password 123456789
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「スタックでの Amazon RDS インスタンスの登録」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRdsDbInstance](#)」の「」を参照してください。

update-volume

次のコード例は、update-volume を使用する方法を示しています。

AWS CLI

登録済みボリュームを更新するには

次の例では、登録済みの Amazon Elastic Block Store (Amazon EBS) ボリュームのマウントポイントを更新します。ボリュームはボリューム ID によって識別GUIDされます。ボリューム ID は、Amazon Elastic Compute Cloud (Amazon EC2) ボリューム ID ではなく、スタックに登録するときにはボリューム AWS OpsWorks に割り当てます。

```
aws opsworks --region us-east-1 update-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df --mount-point /mnt/myvol
```

出力：なし。

詳細情報

詳細については、AWS OpsWorks ユーザーガイドの「インスタンスへの Amazon EBSボリュームの割り当て」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVolume](#)」の「」を参照してください。

AWS OpsWorks CM を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS OpsWorks CM。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-node

次の例は、associate-node を使用する方法を説明しています。

AWS CLI

ノードを関連付けるには

次のassociate-nodeコマンドは、という名前の Chef Automate サーバー*i-44de882p*を関連付けます。つまり*automate-06*、*automate-06*サーバーはノードを管理し、アソシエイトノードコマンドによってノードにインストールされた*chef-client*エージェントソフトウェアを介してレシピコマンドをノードに通信します。有効なノード名はEC2インスタンス IDsです。:

```
aws opsworks-cm associate-node --server-name "automate-06" --node-name "i-43de882p"
--engine-attributes "Name=CHEF_ORGANIZATION,Value='MyOrganization'
Name=CHEF_NODE_PUBLIC_KEY,Value='Public_key_contents'"
```

コマンドによって返される出力は次のようになります。出力:

```
{
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0N0Fymh7E/1Hum0BPsemPQFE6dcGaiFk"
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の AWS OpsWorks 「For Chef Automate でのノードの自動追加」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateNode](#)」の「」を参照してください。

create-backup

次の例は、create-backup を使用する方法を説明しています。

AWS CLI

バックアップを作成するには

次のcreate-backupコマンドは、us-east-1リージョンautomate-06内の という名前の Chef Automate サーバーの手動バックアップを開始します。コマンドは、--descriptionパラメータのバックアップに説明メッセージを追加します。

```
aws opsworks-cm create-backup \  
  --server-name 'automate-06' \  
  --description "state of my infrastructure at launch"
```

出力には、新しいバックアップに関する次のような情報が表示されます。

出力:

```
{  
  "Backups": [  
    {  
      "BackupArn": "string",  
      "BackupId": "automate-06-20160729133847520",  
      "BackupType": "MANUAL",  
      "CreatedAt": 2016-07-29T13:38:47.520Z,  
      "Description": "state of my infrastructure at launch",  
      "Engine": "Chef",  
      "EngineModel": "Single",  
      "EngineVersion": "12",  
      "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/  
automate-06-1010V4UU2WRM2",
```

```

        "InstanceType": "m4.large",
        "KeyPair": "",
        "PreferredBackupWindow": "",
        "PreferredMaintenanceWindow": "",
        "S3LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
        "SecurityGroupIds": [ "sg-1a24c270" ],
        "ServerName": "automate-06",
        "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-
service-role.1114810729735",
        "Status": "OK",
        "StatusDescription": "",
        "SubnetIds": [ "subnet-49436a18" ],
        "ToolsVersion": "string",
        "UserArn": "arn:aws:iam::1019881987024:user/opsworks-user"
    }
],
}

```

詳細については、AWS OpsWorks 「ユーザーガイド」の「Chef Automate Server AWS OpsWorks の のバックアップと復元」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateBackup](#)」の「」を参照してください。

create-server

次の例は、create-server を使用する方法を説明しています。

AWS CLI

サーバーを作成するには

次のcreate-server例では、デフォルトのリージョンautomate-06に という名前の新しい Chef Automate サーバーを作成します。デフォルトは、保持するバックアップの数、メンテナンスとバックアップの開始時間など、他のほとんどの設定で使用されます。create-server コマンドを実行する前に、「Opsworks [AWS OpsWorks for Chef Automate ユーザーガイド](#)」の「[Getting Started with](#) for Chef Automate」の前提条件を完了してください。AWS

```

aws opsworks-cm create-server \
  --engine "Chef" \
  --engine-model "Single" \

```



```
--engine-version "12" \  
--server-name "automate-06" \  
--instance-profile-arn "arn:aws:iam::1019881987024:instance-profile/aws-  
opsworks-cm-ec2-role" \  
--instance-type "t2.medium" \  
--key-pair "amazon-test" \  
--service-role-arn "arn:aws:iam::044726508045:role/aws-opsworks-cm-service-role"
```

出力には、新しいサーバーに関する次のような情報が表示されます。

```
{  
  "Server": {  
    "BackupRetentionCount": 10,  
    "CreatedAt": 2016-07-29T13:38:47.520Z,  
    "DisableAutomatedBackup": FALSE,  
    "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",  
    "Engine": "Chef",  
    "EngineAttributes": [  
      {  
        "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",  
        "Value": "1Password1"  
      }  
    ],  
    "EngineModel": "Single",  
    "EngineVersion": "12",  
    "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/aws-  
opsworks-cm-ec2-role",  
    "InstanceType": "t2.medium",  
    "KeyPair": "amazon-test",  
    "MaintenanceStatus": "",  
    "PreferredBackupWindow": "Sun:02:00",  
    "PreferredMaintenanceWindow": "00:00",  
    "SecurityGroupIds": [ "sg-1a24c270" ],  
    "ServerArn": "arn:aws:iam::1019881987024:instance/  
automate-06-1010V4UU2WRM2",  
    "ServerName": "automate-06",  
    "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-  
role",  
    "Status": "CREATING",  
    "StatusReason": "",  
    "SubnetIds": [ "subnet-49436a18" ]  
  }  
}
```

詳細については、AWS OpsWorks 「Chef Automate APIリファレンス」の[UpdateServer](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateServer](#)」の「」を参照してください。

delete-backup

次の例は、delete-backup を使用する方法を説明しています。

AWS CLI

バックアップを削除するには

次のdelete-backupコマンドは、バックアップ ID で識別される Chef Automate サーバーの手動バックアップまたは自動バックアップを削除します。このコマンドは、保存できるバックアップの最大数に近づいている場合や、Amazon S3 ストレージコストを最小限に抑える場合に便利です。

```
aws opsworks-cm delete-backup --backup-id "automate-06-2016-11-19T23:42:40.240Z"
```

出力は、バックアップの削除が成功したかどうかを示します。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「For Chef Automate Server AWS OpsWorks の のバックアップと復元」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBackup](#)」の「」を参照してください。

delete-server

次のコード例は、delete-server を使用する方法を示しています。

AWS CLI

サーバーを削除するには

次のdelete-serverコマンドは、サーバー名で識別される Chef Automate サーバーを削除します。サーバーが削除されると、DescribeServerリクエストによって返されなくなります。

```
aws opsworks-cm delete-server --server-name "automate-06"
```

出力は、サーバーの削除が成功したかどうかを示します。

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「Chef Automate Server AWS OpsWorks の」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteServer](#)」の「」を参照してください。

describe-account-attributes

次のコード例は、describe-account-attributes を使用する方法を示しています。

AWS CLI

アカウントの属性を記述するには

次のdescribe-account-attributes コマンドは、アカウントの AWS OpsWorks for Chef Automate リソースの使用状況に関する情報を返します。

```
aws opsworks-cm describe-account-attributes
```

コマンドによって返される各アカウント属性エントリの出力は次のようになります。出力:

```
{
  "Attributes": [
    {
      "Maximum": 5,
      "Name": "ServerLimit",
      "Used": 2
    }
  ]
}
```

詳細情報

詳細については、AWS OpsWorks 「Chef Automate APIリファレンス」の DescribeAccountAttributes 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountAttributes](#)」の「」を参照してください。

describe-backups

次のコード例は、describe-backups を使用する方法を示しています。

AWS CLI

バックアップを記述するには

次のdescribe-backupsコマンドは、デフォルトリージョンのアカウントに関連付けられているすべてのバックアップに関する情報を返します。

```
aws opsworks-cm describe-backups
```

コマンドによって返される各バックアップエントリの出力は次のようになります。

出力:

```
{
  "Backups": [
    {
      "BackupArn": "string",
      "BackupId": "automate-06-20160729133847520",
      "BackupType": "MANUAL",
      "CreatedAt": "2016-07-29T13:38:47.520Z",
      "Description": "state of my infrastructure at launch",
      "Engine": "Chef",
      "EngineModel": "Single",
      "EngineVersion": "12",
      "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
      "InstanceType": "m4.large",
      "KeyPair": "",
      "PreferredBackupWindow": "",
      "PreferredMaintenanceWindow": "",
      "S3LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
      "SecurityGroupIds": [ "sg-1a24c270" ],
      "ServerName": "automate-06",
```

```

        "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-
service-role.1114810729735",
        "Status": "Successful",
        "StatusDescription": "",
        "SubnetIds": [ "subnet-49436a18" ],
        "ToolsVersion": "string",
        "UserArn": "arn:aws:iam::1019881987024:user/opsworks-user"
    }
],
}

```

詳細については、AWS OpsWorks 「ユーザーガイド」の「Chef Automate Server AWS OpsWorks の のバックアップと復元」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeBackups](#)」の「」を参照してください。

describe-events

次のコード例は、describe-events を使用する方法を示しています。

AWS CLI

イベントを記述するには

次のdescribe-events例では、指定された Chef Automate サーバーに関連付けられているすべてのイベントに関する情報を返します。

```

aws opsworks-cm describe-events \
  --server-name 'automate-06'

```

コマンドによって返される各イベントエントリの出力は、次の例に似ています。

```

{
  "ServerEvents": [
    {
      "CreatedAt": 2016-07-29T13:38:47.520Z,
      "LogUrl": "https://s3.amazonaws.com/<bucket-name>/
automate-06-20160729133847520",
      "Message": "Updates successfully installed.",
      "ServerName": "automate-06"
    }
  ]
}

```

```
]
}
```

詳細については、AWS OpsWorks 「ユーザーガイド」の「[一般的なトラブルシューティングのヒント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

describe-node-association-status

次の例は、describe-node-association-status を使用する方法を説明しています。

AWS CLI

ノードの関連付けステータスを記述するには

次のdescribe-node-association-statusコマンドは、ノードを という名前の Chef Automate サーバーに関連付けるリクエストのステータスを返します automate-06。

```
aws opsworks-cm describe-node-association-status --server-  
name "automate-06" --node-association-status-token "AfLJKL+/  
GoKLZJBdDQEx0065CDi57b1Qe9nKM8joSok0pQ9xr8DqApBN9/106sLdSvlfDEKkEx+eoCHvjowHa0s="
```

コマンドによって返される各アカウント属性エントリの出力は次のようになります。出力:

```
{  
  "NodeAssociationStatus": "IN_PROGRESS"  
}
```

詳細情報

詳細については、AWS OpsWorks 「Chef Automate APIリファレンス」の DescribeNodeAssociationStatus 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNodeAssociationStatus](#)」の「」を参照してください。

describe-servers

次の例は、describe-servers を使用する方法を説明しています。

AWS CLI

サーバーを記述するには

次のdescribe-serversコマンドは、アカウントに関連付けられているすべてのサーバーと、デフォルトのリージョンに関する情報を返します。

```
aws opsworks-cm describe-servers
```

コマンドによって返される各サーバーエントリの出力は次のようになります。出力:

```
{
  "Servers": [
    {
      "BackupRetentionCount": 8,
      "CreatedAt": 2016-07-29T13:38:47.520Z,
      "DisableAutomatedBackup": FALSE,
      "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
      "Engine": "Chef",
      "EngineAttributes": [
        {
          "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",
          "Value": "1Password1"
        }
      ],
      "EngineModel": "Single",
      "EngineVersion": "12",
      "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
      "InstanceType": "m4.large",
      "KeyPair": "",
      "MaintenanceStatus": "SUCCESS",
      "PreferredBackupWindow": "03:00",
      "PreferredMaintenanceWindow": "Mon:09:00",
      "SecurityGroupIds": [ "sg-1a24c270" ],
      "ServerArn": "arn:aws:iam::1019881987024:instance/automate-06-1010V4UU2WRM2",
      "ServerName": "automate-06",
      "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
      "Status": "HEALTHY",
      "StatusReason": "",
      "SubnetIds": [ "subnet-49436a18" ]
    }
  ]
}
```

```
]
}
```

詳細情報

詳細については、AWS OpsWorks 「for Chef Automate API Guide DescribeServers」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeServers](#)」の「」を参照してください。

disassociate-node

次のコード例は、disassociate-node を使用する方法を示しています。

AWS CLI

ノードの関連付けを解除するには

次のdisassociate-nodeコマンドは、という名前のノードの関連付けを解除しi-44de882p、という名前の Chef Automate サーバーによる管理からノードを削除しますautomate-06。有効なノード名はEC2インスタンス IDsです。：

```
aws opsworks-cm disassociate-node --server-name "automate-06" --node-  
name "i-43de882p" --engine-attributes "Name=CHEF_ORGANIZATION,Value='MyOrganization'  
Name=CHEF_NODE_PUBLIC_KEY,Value='Public_key_contents'"
```

コマンドによって返される出力は次のようになります。出力:

```
{  
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//  
rHRqHDWXxwVoNBxcEy4V7R0NOFymh7E/1Hum0BPsemPQFE6dcGaiFk"  
}
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「Chef Automate Server AWS OpsWorks の」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateNode](#)」の「」を参照してください。

restore-server

次のコード例は、restore-server を使用する方法を示しています。

AWS CLI

サーバーを復元するには

次のrestore-serverコマンドは、ID が のバックアップから、デフォルトのリージョン automate-06で という名前の Chef Automate サーバーをインプレース復元します automate-06-2016-11-22T16:13:27.998Z。サーバーを復元すると、Chef Automate サーバーが指定したバックアップの実行時に管理していたノードへの接続が復元されます。

```
aws opsworks-cm restore-server --backup-id 「automate-06-2016-11-22T16:13:27.998Z」 --server-name 「automate-06」
```

出力はコマンド ID のみです。出力:

```
(None)
```

詳細情報

詳細については、AWS OpsWorks 「ユーザーガイド」の「Chef Automate Server AWS OpsWorks の失敗の復元」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreServer](#)」の「」を参照してください。

start-maintenance

次のコード例は、start-maintenance を使用する方法を示しています。

AWS CLI

メンテナンスを開始するには

次のstart-maintenance例では、デフォルトのリージョンで指定された Chef Automate または Puppet Enterprise サーバーでメンテナンスを手動で開始します。このコマンドは、以前の自動メンテナンスの試行が失敗し、メンテナンスの失敗の根本的な原因が解決された場合に役立ちます。

```
aws opsworks-cm start-maintenance \  
--server-name 'automate-06'
```

出力:

```
{  
  "Server": {  
    "AssociatePublicIpAddress": true,  
    "BackupRetentionCount": 10,  
    "ServerName": "automate-06",  
    "CreatedAt": 1569229584.842,  
    "CloudFormationStackArn": "arn:aws:cloudformation:us-  
west-2:123456789012:stack/aws-opsworks-cm-instance-automate-06-1606611794746/  
EXAMPLE0-31de-11eb-bdb0-0a5b0a1353b8",  
    "DisableAutomatedBackup": false,  
    "Endpoint": "automate-06-EXAMPLEv8gjk5f.us-west-2.opsworks-cm.io",  
    "Engine": "ChefAutomate",  
    "EngineModel": "Single",  
    "EngineAttributes": [],  
    "EngineVersion": "2020-07",  
    "InstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/aws-  
opsworks-cm-ec2-role",  
    "InstanceType": "m5.large",  
    "PreferredMaintenanceWindow": "Sun:01:00",  
    "PreferredBackupWindow": "Sun:15:00",  
    "SecurityGroupIds": [  
      "sg-EXAMPLE"  
    ],  
    "ServiceRoleArn": "arn:aws:iam::123456789012:role/service-role/aws-opsworks-  
cm-service-role",  
    "Status": "UNDER_MAINTENANCE",  
    "SubnetIds": [  
      "subnet-EXAMPLE"  
    ],  
    "ServerArn": "arn:aws:opsworks-cm:us-west-2:123456789012:server/  
automate-06/0148382d-66b0-4196-8274-d1a2b6dff8d1"  
  }  
}
```

詳細については、AWS OpsWorks 「ユーザーガイド」の「[システムメンテナンス \(Puppet Enterprise サーバー\)](#)」または「[システムメンテナンス \(Chef Automate サーバー\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartMaintenance](#)」の「」を参照してください。

update-server-engine-attributes

次の例は、update-server-engine-attributes を使用する方法を説明しています。

AWS CLI

サーバーエンジン属性を更新するには

次のupdate-server-engine-attributesコマンドは、 という名前の Chef Automate サーバーのCHEF_PIVOTAL_KEYエンジン属性の値を更新しますautomate-06。現在、他のエンジン属性の値を変更することはできません。

```
aws opsworks-cm update-server-engine-attributes \  
  --attribute-name CHEF_PIVOTAL_KEY \  
  --attribute-value "new key value" \  
  --server-name "automate-06"
```

出力には、更新されたサーバーに関する次のような情報が表示されます。

```
{  
  "Server": {  
    "BackupRetentionCount": 2,  
    "CreatedAt": 2016-07-29T13:38:47.520Z,  
    "DisableAutomatedBackup": FALSE,  
    "Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",  
    "Engine": "Chef",  
    "EngineAttributes": [  
      {  
        "Name": "CHEF_PIVOTAL_KEY",  
        "Value": "new key value"  
      }  
    ],  
    "EngineModel": "Single",  
    "EngineVersion": "12",  
    "InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/  
automate-06-1010V4UU2WRM2",  
    "InstanceType": "m4.large",  
    "KeyPair": "",
```

```
    "MaintenanceStatus": "SUCCESS",
    "PreferredBackupWindow": "Mon:09:15",
    "PreferredMaintenanceWindow": "03:00",
    "SecurityGroupIds": [ "sg-1a24c270" ],
    "ServerArn": "arn:aws:iam::1019881987024:instance/
automate-06-1010V4UU2WRM2",
    "ServerName": "automate-06",
    "ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
    "Status": "HEALTHY",
    "StatusReason": "",
    "SubnetIds": [ "subnet-49436a18" ]
  }
}
```

詳細については、AWS OpsWorks 「Chef Automate APIリファレンス」
の[UpdateServerEngineAttributes](#) 「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateServerEngineAttributes](#)」の「」を参照してください。

update-server

次の例は、update-server を使用する方法を説明しています。

AWS CLI

サーバーを更新するには

次のupdate-serverコマンドは、デフォルトのリージョンで指定された Chef Automate サーバーのメンテナンス開始時刻を更新します。--preferred-maintenance-window パラメータを追加して、サーバーメンテナンスの開始日時を月曜日の午前 9 時 15 分に変更します。UTC.:

```
aws opsworks-cm update-server \  
  --server-name "automate-06" \  
  --preferred-maintenance-window "Mon:09:15"
```

出力には、更新されたサーバーに関する次のような情報が表示されます。

```
{  
  "Server": {
```

```
"BackupRetentionCount": 8,
"CreatedAt": 2016-07-29T13:38:47.520Z,
"DisableAutomatedBackup": TRUE,
"Endpoint": "https://opsworks-cm.us-east-1.amazonaws.com",
"Engine": "Chef",
"EngineAttributes": [
  {
    "Name": "CHEF_DELIVERY_ADMIN_PASSWORD",
    "Value": "1Password1"
  }
],
"EngineModel": "Single",
"EngineVersion": "12",
"InstanceProfileArn": "arn:aws:iam::1019881987024:instance-profile/
automate-06-1010V4UU2WRM2",
"InstanceType": "m4.large",
"KeyPair": "",
"MaintenanceStatus": "OK",
"PreferredBackupWindow": "Mon:09:15",
"PreferredMaintenanceWindow": "03:00",
"SecurityGroupIds": [ "sg-1a24c270" ],
"ServerArn": "arn:aws:iam::1019881987024:instance/
automate-06-1010V4UU2WRM2",
"ServerName": "automate-06",
"ServiceRoleArn": "arn:aws:iam::1019881987024:role/aws-opsworks-cm-service-
role.1114810729735",
"Status": "HEALTHY",
"StatusReason": "",
"SubnetIds": [ "subnet-49436a18" ]
}
}
```

詳細については、AWS OpsWorks 「Chef Automate APIリファレンス」の[UpdateServer](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateServer](#)」の「」を参照してください。

を使用した組織の例 AWS CLI

次のコード例は、Organizations AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-handshake

次の例は、accept-handshake を使用する方法を説明しています。

AWS CLI

別のアカウントからハンドシェイクを受け入れるには

組織の所有者である Bill は、以前 Juan のアカウントを自分の組織に招待しました。次の例は、Juan のアカウントがハンドシェイクを受け入れ、招待に同意したことを示しています。

```
aws organizations accept-handshake --handshake-id h-examplehandshakeid111
```

出力は、次のように表示されます。

```
{
  "Handshake": {
    "Action": "INVITE",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "RequestedTimestamp": 1481656459.257,
    "ExpirationTimestamp": 1482952459.257,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "juan@example.com",
```

```
        "Type": "EMAIL"
      }
    ],
    "Resources": [
      {
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@amazon.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Org Master Account"
          },
          {
            "Type": "ORGANIZATION_FEATURE_SET",
            "Value": "ALL"
          }
        ],
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid"
      },
      {
        "Type": "EMAIL",
        "Value": "juan@example.com"
      }
    ],
    "State": "ACCEPTED"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptHandshake](#)」の「」を参照してください。

attach-policy

次のコード例は、attach-policy を使用する方法を示しています。

AWS CLI

root、OU、またはアカウントにポリシーをアタッチするには

例 1

次の例は、サービスコントロールポリシー (SCP) を OU にアタッチする方法を示しています。

```
aws organizations attach-policy
  --policy-id p-examplepolicyid111
  --target-id ou-examplerootid111-exampleoid111
```

例 2

次の例は、サービスコントロールポリシーをアカウントに直接アタッチする方法を示しています。

```
aws organizations attach-policy
  --policy-id p-examplepolicyid111
  --target-id 333333333333
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachPolicy](#)」の「」を参照してください。

cancel-handshake

次の例は、cancel-handshake を使用する方法を説明しています。

AWS CLI

別のアカウントから送信されたハンドシェイクをキャンセルするには

ビルは以前に、スーザンのアカウントに組織に参加する招待を送信しました。スーザンが承諾する前に、彼は考えを変え、招待をキャンセルすることにしました。次の例は、Bill のキャンセルを示しています。

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

出力には、状態が `CANCELED` になったことを示すハンドシェイクオブジェクトが含まれます。

```
{
  "Handshake": {
    "Id": "h-examplehandshakeid111",
    "State": "CANCELED",
    "Action": "INVITE",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
```



```
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "susan@example.com",
        "Type": "EMAIL"
      }
    ],
    "Resources": [
      {
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid",
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@example.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Master Account"
          },
          {
            "Type": "ORGANIZATION_FEATURE_SET",
            "Value": "CONSOLIDATED_BILLING"
          }
        ]
      },
      {
        "Type": "EMAIL",
        "Value": "anika@example.com"
      },
      {
        "Type": "NOTES",
        "Value": "This is a request for Susan's account to
join Bob's organization."
      }
    ],
    "RequestedTimestamp": 1.47008383521E9,
    "ExpirationTimestamp": 1.47137983521E9
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelHandshake](#)」の「」を参照してください。

create-account

次のコード例は、create-account を使用する方法を示しています。

AWS CLI

自動的に組織の一部となるメンバーアカウントを作成するには

次の例は、組織のメンバーアカウントを作成する方法を示しています。メンバーアカウントは、「プロダクションアカウント」という名前と E メールアドレス (susan@example.com) で構成されます。roleName パラメータが指定されていない OrganizationAccountAccessRole ため、組織は のデフォルト名を使用して IAM ロールを自動的に作成します。また、IamUserAccessToBilling パラメータが指定されていない ALLOW ため、アカウント請求データにアクセスするための十分なアクセス許可を持つ IAM ユーザーまたはロールを許可する設定は、 のデフォルト値に設定されます。組織はスーザンに「ようこそ」の AWS E メールを自動的に送信します。

```
aws organizations create-account --email susan@example.com --account-name "Production Account"
```

出力には、ステータスが現在の IN_PROGRESS 状態であることを示すリクエストオブジェクトが含まれます。

```
{
  "CreateAccountStatus": {
    "State": "IN_PROGRESS",
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

後で、パラメータの値として Id レスポンス値を コマンドに提供 describe-create-account-status することで、リクエストの現在のステータスを create-account-request-id クエリできます。

詳細については、「Organizations ユーザーガイド」の「組織で AWS アカウントを作成する」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAccount](#)」の「」を参照してください。

create-organization

次のコード例は、create-organization を使用する方法を示しています。

AWS CLI

例 1: 新しい組織を作成するには

Bill は、アカウント 111111111111 の認証情報を使用して組織を作成したいと考えています。次の例は、このアカウントが新しい組織のマスターアカウントになることを示しています。Bill は機能セットを指定していないため、新しい組織ではデフォルトですべての機能が有効になり、サービスコントロールポリシーがルート上で有効になります。

```
aws organizations create-organization
```

出力には、新しい組織に関する詳細を含む組織オブジェクトが含まれます。

```
{
  "Organization": {
    "AvailablePolicyTypes": [
      {
        "Status": "ENABLED",
        "Type": "SERVICE_CONTROL_POLICY"
      }
    ],
    "MasterAccountId": "111111111111",
    "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
    "MasterAccountEmail": "bill@example.com",
    "FeatureSet": "ALL",
    "Id": "o-exampleorgid",
    "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid"
  }
}
```

例 2: 一括決済機能のみを有効にした新しい組織を作成するには

次の例では、一括決済機能のみをサポートする組織を作成します。

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

出力には、新しい組織に関する詳細を含む組織オブジェクトが含まれます。

```
{
  "Organization": {
    "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
    "AvailablePolicyTypes": [],
    "Id": "o-exampleorgid",
    "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
    "MasterAccountEmail": "bill@example.com",
    "MasterAccountId": "111111111111",
    "FeatureSet": "CONSOLIDATED_BILLING"
  }
}
```

詳細については、「AWS Organizations ユーザーガイド」の「Creating an Organization」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateOrganization](#)」の「」を参照してください。

create-organizational-unit

次の例は、create-organizational-unit を使用する方法を説明しています。

AWS CLI

ルート OU または親 OU に OU を作成するには

次の例は、AccountingOU という名前の OU を作成する方法を示しています。

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

出力には、新しい OU の詳細を含む organizationalUnit オブジェクトが含まれます。

```
{
  "OrganizationalUnit": {
    "Id": "ou-examplerootid111-exampleouid111",
    "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
  }
}
```

```

        "Name": "AccountingOU"
    }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateOrganizationalUnit](#)」の「」を参照してください。

create-policy

次のコード例は、create-policy を使用する方法を示しています。

AWS CLI

例 1: ポリシーのテキストソースファイルを使用してJSONポリシーを作成するには

次の例は、という名前のサービスコントロールポリシー (SCP) を作成する方法を示しています AllowAllS3Actions。ポリシーの内容は、policy.json というローカルコンピューター上のファイルから取得されます。

```

aws organizations create-policy --content file://policy.json --
name AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows
delegation of all S3 actions"

```

出力には、新しいポリシーの詳細を含むポリシーオブジェクトが含まれます。

```

{
  "Policy": {
    "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\\\"Allow\\\",\\\"Action\":[\\\"s3:*\\\"],\\\"Resource\":[\\\"*\\\"]}]}",
    "PolicySummary": {
      "Arn": "arn:aws:organizations::o-exampleorgid:policy/
service_control_policy/p-examplepolicyid111",
      "Description": "Allows delegation of all S3 actions",
      "Name": "AllowAllS3Actions",
      "Type": "SERVICE_CONTROL_POLICY"
    }
  }
}

```

例 2: ポリシーをパラメータとしてJSONポリシーを作成するには

次の例はSCP、パラメータにポリシーコンテンツをJSON文字列として埋め込むことで、今回同じを作成する方法を示しています。文字列は、パラメータ内でリテラルとして扱われるように、二重引用符の前にバックスラッシュを付けてエスケープする必要があります。パラメータ自体も二重引用符で囲みます。

```
aws organizations create-policy --content "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"*\"]}]}\" --name AllowAllS3Actions --type SERVICE_CONTROL_POLICY --description Allows delegation of all S3 actions
```

Organizations でのポリシーの作成と使用の詳細については、「AWS Organizations ユーザーガイド」の「AWS Organizations のポリシーの管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePolicy](#)」の「」を参照してください。

decline-handshake

次の例は、decline-handshake を使用する方法を説明しています。

AWS CLI

別のアカウントから送信されたハンドシェイクを拒否するには

次の例は、アカウント 222222222222 の所有者である管理者である Susan が、Bill の組織への招待を拒否したことを示しています。DeclineHandshake オペレーションはハンドシェイクオブジェクトを返し、状態が `DECLINED` になったことを示します。

```
aws organizations decline-handshake --handshake-id h-examplehandshakeid111
```

出力には、の新しい状態を示すハンドシェイクオブジェクトが含まれます `DECLINED`。

```
{
  "Handshake": {
    "Id": "h-examplehandshakeid111",
    "State": "DECLINED",
    "Resources": [
      {
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid",
```

```

        "Resources": [
            {
                "Type": "MASTER_EMAIL",
                "Value": "bill@example.com"
            },
            {
                "Type": "MASTER_NAME",
                "Value": "Master Account"
            }
        ],
        {
            "Type": "EMAIL",
            "Value": "susan@example.com"
        },
        {
            "Type": "NOTES",
            "Value": "This is an invitation to Susan's account
to join the Bill's organization."
        }
    ],
    "Parties": [
        {
            "Type": "EMAIL",
            "Id": "susan@example.com"
        },
        {
            "Type": "ORGANIZATION",
            "Id": "o-exampleorgid"
        }
    ],
    "Action": "INVITE",
    "RequestedTimestamp": 1470684478.687,
    "ExpirationTimestamp": 1471980478.687,
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeclineHandshake](#)」の「」を参照してください。

delete-organization

次の例は、delete-organization を使用する方法を説明しています。

AWS CLI

組織を削除するには

次の例は、組織を削除する方法を示しています。この操作を実行するには、組織のマスターアカウントの管理者である必要があります。この例では、以前に組織からすべてのメンバーアカウント、OUs、およびポリシーを削除したことを前提としています。

```
aws organizations delete-organization
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteOrganization](#)」の「」を参照してください。

delete-organizational-unit

次のコード例は、delete-organizational-unit を使用する方法を示しています。

AWS CLI

OU を削除するには

次の例は、OU を削除する方法を示しています。この例では、以前に OU OUsからすべてのアカウントと他のアカウントを削除したことを前提としています。

```
aws organizations delete-organizational-unit --organizational-unit-id ou-examplerootid111-exampleoid111
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteOrganizationalUnit](#)」の「」を参照してください。

delete-policy

次の例は、delete-policy を使用する方法を説明しています。

AWS CLI

ポリシーを削除するには

次の例は、組織からポリシーを削除する方法を示しています。この例では、ポリシーをすべてのエンティティから事前にデタッチしたことを前提としています。

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicy](#)」の「」を参照してください。

describe-account

次のコード例は、describe-account を使用する方法を示しています。

AWS CLI

アカウントの詳細を取得するには

次の例は、アカウントの詳細をリクエストする方法を示しています。

```
aws organizations describe-account --account-id 555555555555
```

出力には、アカウントに関する詳細を含むアカウントオブジェクトが表示されます。

```
{
  "Account": {
    "Id": "555555555555",
    "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/555555555555",
    "Name": "Beta account",
    "Email": "anika@example.com",
    "JoinedMethod": "INVITED",
    "JoinedTimeStamp": 1481756563.134,
    "Status": "ACTIVE"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccount](#)」の「」を参照してください。

describe-create-account-status

次のコード例は、describe-create-account-status を使用する方法を示しています。

AWS CLI

アカウントの作成リクエストに関する最新のステータスを取得するには

次の例は、以前のリクエストの最新ステータスをリクエストして組織内にアカウントを作成する方法を示しています。指定された `--request-id` は、`create-account` への元の呼び出しのレスポンスから取得されます。アカウント作成リクエストは、Organizations がアカウントの作成を正常に完了したことをステータスフィールドで表示します。

コマンド:

```
aws organizations describe-create-account-status --create-account-request-id car-examplecreateaccountrequestid111
```

出力:

```
{
  "CreateAccountStatus": {
    "State": "SUCCEEDED",
    "AccountId": "555555555555",
    "AccountName": "Beta account",
    "RequestedTimestamp": 1470684478.687,
    "CompletedTimestamp": 1470684532.472,
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCreateAccountStatus](#)」の「」を参照してください。

describe-handshake

次の例は、`describe-handshake` を使用する方法を説明しています。

AWS CLI

ハンドシェイクに関する情報を取得するには

次の例は、ハンドシェイクの詳細をリクエストする方法を示しています。ハンドシェイク ID は `InviteAccountToOrganization`、への元の呼び出し、

または `ListHandshakesForAccount` または `への呼び出しから取得されま`
`ずListHandshakesForOrganization`。

```
aws organizations describe-handshake --handshake-id h-examplehandshakeid111
```

出力には、リクエストされたハンドシェイクに関するすべての詳細を含むハンドシェイクオブジェクトが含まれます。

```
{
  "Handshake": {
    "Id": "h-examplehandshakeid111",
    "State": "OPEN",
    "Resources": [
      {
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid",
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@example.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Master Account"
          }
        ]
      },
      {
        "Type": "EMAIL",
        "Value": "anika@example.com"
      }
    ],
    "Parties": [
      {
        "Type": "ORGANIZATION",
        "Id": "o-exampleorgid"
      },
      {
        "Type": "EMAIL",
        "Id": "anika@example.com"
      }
    ],
    "Action": "INVITE",
  }
}
```

```
        "RequestedTimestamp": 1470158698.046,  
        "ExpirationTimestamp": 1471454698.046,  
        "Arn": "arn:aws:organizations::111111111111:handshake/o-  
exampleorgid/invite/h-examplehandshakeid111"  
    }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeHandshake](#)」の「」を参照してください。

describe-organization

次の例は、describe-organization を使用する方法を説明しています。

AWS CLI

現在の組織に関する情報を取得するには

次の例は、組織の詳細をリクエストする方法を示しています。

```
aws organizations describe-organization
```

出力には、組織に関する詳細を含む組織オブジェクトが含まれます。

```
{  
  "Organization": {  
    "MasterAccountArn": "arn:aws:organizations::111111111111:account/o-  
exampleorgid/111111111111",  
    "MasterAccountEmail": "bill@example.com",  
    "MasterAccountId": "111111111111",  
    "Id": "o-exampleorgid",  
    "FeatureSet": "ALL",  
    "Arn": "arn:aws:organizations::111111111111:organization/o-  
exampleorgid",  
    "AvailablePolicyTypes": [  
      {  
        "Status": "ENABLED",  
        "Type": "SERVICE_CONTROL_POLICY"  
      }  
    ]  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOrganization](#)」の「」を参照してください。

describe-organizational-unit

次のコード例は、describe-organizational-unit を使用する方法を示しています。

AWS CLI

OU に関する情報を取得するには

次のdescribe-organizational-unit例では、OU の詳細をリクエストします。

```
aws organizations describe-organizational-unit \  
--organizational-unit-id ou-examplerootid111-exampleoid111
```

出力:

```
{  
  "OrganizationalUnit": {  
    "Name": "Accounting Group",  
    "Arn": "arn:aws:organizations::123456789012:ou/o-exampleorgid/ou-  
examplerootid111-exampleoid111",  
    "Id": "ou-examplerootid111-exampleoid111"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOrganizationalUnit](#)」の「」を参照してください。

describe-policy

次の例は、describe-policy を使用する方法を説明しています。

AWS CLI

ポリシーに関する情報を取得するには

次の例は、ポリシーに関する情報をリクエストする方法を示しています。

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

出力には、ポリシーの詳細を含むポリシーオブジェクトが含まれます。

```
{
  "Policy": {
    "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n    {\n      \"Effect\": \"Allow\",\n      \"Action\": \"*\",\n      \"Resource\": \"*\"\n    }]\n}",
    "PolicySummary": {
      "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/service_control_policy/p-examplepolicyid111",
      "Type": "SERVICE_CONTROL_POLICY",
      "Id": "p-examplepolicyid111",
      "AwsManaged": false,
      "Name": "AllowAllS3Actions",
      "Description": "Enables admins to delegate S3 permissions"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePolicy](#)」の「」を参照してください。

detach-policy

次の例は、detach-policy を使用する方法を説明しています。

AWS CLI

root、OU、またはアカウントからポリシーをデタッチするには

次のコード例は、OU からポリシーをデタッチする方法を示しています。

```
aws organizations detach-policy --target-id ou-examplerootid111-exampleouid111 --policy-id p-examplepolicyid111
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetachPolicy](#)」の「」を参照してください。

disable-policy-type

次の例は、disable-policy-type を使用する方法を説明しています。

AWS CLI

ルートのポリシータイプを無効にするには

次の例は、ルートでサービスコントロールポリシー (SCP) ポリシータイプを無効にする方法を示しています。

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-type SERVICE_CONTROL_POLICY
```

出力は、PolicyTypes レスポンス要素に SERVICE_CONTROL_ が含まれなくなったことを示しますPOLICY。

```
{
  "Root": {
    "PolicyTypes": [],
    "Name": "Root",
    "Id": "r-examplerootid111",
    "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-examplerootid111"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisablePolicyType](#)」の「」を参照してください。

enable-all-features

次のコード例は、enable-all-features を使用する方法を示しています。

AWS CLI

組織内のすべての機能を有効にするには

この例では、管理者が組織内のすべての招待アカウントに、組織内のすべての有効機能を承認するように求めています。AWS 組織は、招待されたすべてのメンバーアカウントに登録されているアドレスに E メールを送信し、送信されたハンドシェイクを受け入れることで、所有者にすべての機能の変更を承認するように求めます。招待されたすべてのメンバーアカウントがハンドシェイクを受け入れると、組織管理者はすべての機能に対する変更を確定し、適切なアクセス許可を持つメンバーアカウントはポリシーを作成してルート、OUs、およびアカウントに適用できます。

aws organizations enable-all-features

出力はハンドシェイクオブジェクトで、招待されたすべてのメンバーアカウントに送信され、承認されます。

```
{
  "Handshake": {
    "Action": "ENABLE_ALL_FEATURES",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-exampleorgid/enable_all_features/h-examplehandshakeid111",
    "ExpirationTimestamp": 1.483127868609E9,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "id": "o-exampleorgid",
        "type": "ORGANIZATION"
      }
    ],
    "requestedTimestamp": 1.481831868609E9,
    "resources": [
      {
        "type": "ORGANIZATION",
        "value": "o-exampleorgid"
      }
    ],
    "state": "REQUESTED"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableAllFeatures](#)」の「」を参照してください。

enable-policy-type

次の例は、enable-policy-type を使用する方法を説明しています。

AWS CLI

ルートでポリシータイプの使用を有効にするには

次の例は、ルートでサービスコントロールポリシー (SCP) ポリシータイプを有効にする方法を示しています。


```
aws organizations enable-policy-type --root-id r-examplerootid111 --policy-type SERVICE_CONTROL_POLICY
```

出力には、有効SCPsになった policyTypes レスポンス要素を含むルートオブジェクトが表示されます。

```
{
  "Root": {
    "PolicyTypes": [
      {
        "Status": "ENABLED",
        "Type": "SERVICE_CONTROL_POLICY"
      }
    ],
    "Id": "r-examplerootid111",
    "Name": "Root",
    "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-examplerootid111"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[EnablePolicyType](#)」の「」を参照してください。

invite-account-to-organization

次のコード例は、invite-account-to-organization を使用する方法を示しています。

AWS CLI

組織にアカウントを招待するには

次の例は、bill@example.com が所有するマスターアカウントで、juan@example.com が所有するアカウントを組織への参加に招待しています。

```
aws organizations invite-account-to-organization --target '{"Type": "EMAIL", "Id": "juan@example.com"}' --notes "This is a request for Juan's account to join Bill's organization."
```

出力には、招待されたアカウントに送信される内容を示すハンドシェイク構造が含まれます。

```
{
  "Handshake": {
    "Action": "INVITE",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "ExpirationTimestamp": 1482952459.257,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "juan@example.com",
        "Type": "EMAIL"
      }
    ],
    "RequestedTimestamp": 1481656459.257,
    "Resources": [
      {
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@amazon.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Org Master Account"
          },
          {
            "Type": "ORGANIZATION_FEATURE_SET",
            "Value": "FULL"
          }
        ],
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid"
      },
      {
        "Type": "EMAIL",
        "Value": "juan@example.com"
      }
    ],
    "State": "OPEN"
  }
}
```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[InviteAccountToOrganization](#)」の「」を参照してください。

leave-organization

次のコード例は、leave-organization を使用する方法を示しています。

AWS CLI

組織をメンバーアカウントとして残すには

次の例は、現在メンバーである組織から退出することをリクエストしているメンバーアカウントの管理者を示しています。

```
aws organizations leave-organization
```

- API 詳細については、AWS CLI 「コマンドリファレンス[LeaveOrganization](#)」の「」を参照してください。

list-accounts-for-parent

次の例は、list-accounts-for-parent を使用する方法を説明しています。

AWS CLI

指定された親ルートまたは OU 内のすべてのアカウントのリストを取得するには

次の例は、OU 内のアカウントのリストをリクエストする方法を示しています。

```
aws organizations list-accounts-for-parent --parent-id ou-examplerootid111-exampleouid111
```

出力には、アカウントサマリーオブジェクトのリストが含まれます。

```
{  
  "Accounts": [  
    {
```

```
        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333",
        "JoinedMethod": "INVITED",
        "JoinedTimestamp": 1481835795.536,
        "Id": "333333333333",
        "Name": "Development Account",
        "Email": "juan@example.com",
        "Status": "ACTIVE"
    },
    {
        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/444444444444",
        "JoinedMethod": "INVITED",
        "JoinedTimestamp": 1481835812.143,
        "Id": "444444444444",
        "Name": "Test Account",
        "Email": "anika@example.com",
        "Status": "ACTIVE"
    }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccountsForParent](#)」の「」を参照してください。

list-accounts

次のコード例は、list-accounts を使用する方法を示しています。

AWS CLI

組織内のすべてのアカウントのリストを取得するには

次の例は、組織内のアカウントのリストをリクエストする方法を示しています。

```
aws organizations list-accounts
```

出力には、アカウントサマリーオブジェクトのリストが含まれます。

```
{
  "Accounts": [
    {
```

```
    "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
    "JoinedMethod": "INVITED",
    "JoinedTimestamp": 1481830215.45,
    "Id": "111111111111",
    "Name": "Master Account",
    "Email": "bill@example.com",
    "Status": "ACTIVE"
  },
  {
    "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/222222222222",
    "JoinedMethod": "INVITED",
    "JoinedTimestamp": 1481835741.044,
    "Id": "222222222222",
    "Name": "Production Account",
    "Email": "alice@example.com",
    "Status": "ACTIVE"
  },
  {
    "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333",
    "JoinedMethod": "INVITED",
    "JoinedTimestamp": 1481835795.536,
    "Id": "333333333333",
    "Name": "Development Account",
    "Email": "juan@example.com",
    "Status": "ACTIVE"
  },
  {
    "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/444444444444",
    "JoinedMethod": "INVITED",
    "JoinedTimestamp": 1481835812.143,
    "Id": "444444444444",
    "Name": "Test Account",
    "Email": "anika@example.com",
    "Status": "ACTIVE"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccounts](#)」の「」を参照してください。

list-children

次のコード例は、list-children を使用する方法を示しています。

AWS CLI

親 OU またはルートOUsの子アカウントと を取得するには

次の例では、そのアカウント 444444444444 を含むルートまたは OU を一覧表示する方法を示します。

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-examplerootid111-exampleouid111
```

出力には、親に含まれる OUs 2 つの子が表示されます。

```
{
  "Children": [
    {
      "Id": "ou-examplerootid111-exampleouid111",
      "Type": "ORGANIZATIONAL_UNIT"
    },
    {
      "Id": "ou-examplerootid111-exampleouid222",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListChildren](#)」の「」を参照してください。

list-create-account-status

次の例は、list-create-account-status を使用する方法を説明しています。

AWS CLI

例 1: 現在の組織で行われたアカウント作成リクエストのリストを取得するには

次の例は、正常に完了した組織のアカウント作成リクエストのリストをリクエストする方法を示しています。

```
aws organizations list-create-account-status --states SUCCEEDED
```

出力には、各リクエストに関する情報を含むオブジェクトの配列が含まれます。

```
{
  "CreateAccountStatuses": [
    {
      "AccountId": "4444444444444444",
      "AccountName": "Developer Test Account",
      "CompletedTimeStamp": 1481835812.143,
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "State": "SUCCEEDED"
    }
  ]
}
```

例 2: 現在の組織で行われた進行中のアカウント作成リクエストのリストを取得するには

次の例では、組織の進行中のアカウント作成リクエストのリストを取得します。

```
aws organizations list-create-account-status --states IN_PROGRESS
```

出力には、各リクエストに関する情報を含むオブジェクトの配列が含まれます。

```
{
  "CreateAccountStatuses": [
    {
      "State": "IN_PROGRESS",
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "AccountName": "Production Account"
    }
  ]
}
```

```
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCreateAccountStatus](#)」の「」を参照してください。

list-handshakes-for-account

次の例は、list-handshakes-for-account を使用する方法を説明しています。

AWS CLI

アカウントに送信されたハンドシェイクのリストを取得するには

次の例は、オペレーションの呼び出しに使用された認証情報のアカウントに関連付けられているすべてのハンドシェイクのリストを取得する方法を示しています。

```
aws organizations list-handshakes-for-account
```

出力には、現在の状態を含む各ハンドシェイクに関する情報を含むハンドシェイク構造のリストが含まれます。

```
{
  "Handshake": {
    "Action": "INVITE",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "ExpirationTimestamp": 1482952459.257,
    "Id": "h-examplehandshakeid111",
    "Parties": [
      {
        "Id": "o-exampleorgid",
        "Type": "ORGANIZATION"
      },
      {
        "Id": "juan@example.com",
        "Type": "EMAIL"
      }
    ],
    "RequestedTimestamp": 1481656459.257,
    "Resources": [
```



```
{
  "Resources": [
    {
      "Type": "MASTER_EMAIL",
      "Value": "bill@amazon.com"
    },
    {
      "Type": "MASTER_NAME",
      "Value": "Org Master Account"
    },
    {
      "Type": "ORGANIZATION_FEATURE_SET",
      "Value": "FULL"
    }
  ],
  "Type": "ORGANIZATION",
  "Value": "o-exampleorgid"
},
{
  "Type": "EMAIL",
  "Value": "juan@example.com"
}
],
"State": "OPEN"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListHandshakesForAccount](#)」の「」を参照してください。

list-handshakes-for-organization

次の例は、list-handshakes-for-organization を使用する方法を説明しています。

AWS CLI

組織に関連付けられているハンドシェイクのリストを取得するには

次の例は、現在の組織に関連付けられているハンドシェイクのリストを取得する方法を示しています。

```
aws organizations list-handshakes-for-organization
```

出力には 2 つのハンドシェイクが表示されます。1 つ目は Juan のアカウントへの招待で、の状態を示します OPEN。2 つ目は Anika の アカウントへの招待で、の状態を示します ACCEPTED。

```
{
  "Handshakes": [
    {
      "Action": "INVITE",
      "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
      "ExpirationTimestamp": 1482952459.257,
      "Id": "h-examplehandshakeid111",
      "Parties": [
        {
          "Id": "o-exampleorgid",
          "Type": "ORGANIZATION"
        },
        {
          "Id": "juan@example.com",
          "Type": "EMAIL"
        }
      ],
      "RequestedTimestamp": 1481656459.257,
      "Resources": [
        {
          "Resources": [
            {
              "Type": "MASTER_EMAIL",
              "Value": "bill@amazon.com"
            },
            {
              "Type": "MASTER_NAME",
              "Value": "Org Master
Account"
            }
          ],
          "Type":
"ORGANIZATION_FEATURE_SET",
          "Value": "FULL"
        }
      ],
      "Type": "ORGANIZATION",
      "Value": "o-exampleorgid"
    }
  ]
}
```

```

        },
        {
            "Type": "EMAIL",
            "Value": "juan@example.com"
        },
        {
            "Type": "NOTES",
            "Value": "This is an invitation to Juan's
account to join Bill's organization."
        }
    ],
    "State": "OPEN"
},
{
    "Action": "INVITE",
    "State": "ACCEPTED",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "ExpirationTimestamp": 1.471797437427E9,
    "Id": "h-examplehandshakeid222",
    "Parties": [
        {
            "Id": "o-exampleorgid",
            "Type": "ORGANIZATION"
        },
        {
            "Id": "anika@example.com",
            "Type": "EMAIL"
        }
    ],
    "RequestedTimestamp": 1.469205437427E9,
    "Resources": [
        {
            "Resources": [
                {
                    "Type": "MASTER_EMAIL",
                    "Value": "bill@example.com"
                },
                {
                    "Type": "MASTER_NAME",
                    "Value": "Master Account"
                }
            ]
        },
        {
            "Type": "ORGANIZATION",

```

```

    "Value": "o-exampleorgid"
  },
  {
    "Type": "EMAIL",
    "Value": "anika@example.com"
  },
  {
    "Type": "NOTES",
    "Value": "This is an invitation to Anika's
account to join Bill's organization."
  }
]
}
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListHandshakesForOrganization](#)」の「」を参照してください。

list-organizational-units-for-parent

次の例は、list-organizational-units-for-parent を使用する方法を説明しています。

AWS CLI

親 OU またはルートOUs内の のリストを取得するには

次の例は、指定されたルートOUsで のリストを取得する方法を示しています。

```
aws organizations list-organizational-units-for-parent --parent-id r-  
examplerootid111
```

出力は、指定されたルートに 2 つの が含まれていることを示しOUs、それぞれの詳細を示します。

```

{
  "OrganizationalUnits": [
    {
      "Name": "AccountingDepartment",
      "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-  
examplerootid111/ou-examplerootid111-exampleouid111"
    },
  ],
}

```

```
    {
      "Name": "ProductionDepartment",
      "Arn": "arn:aws:organizations:o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid222"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOrganizationalUnitsForParent](#)」の「」を参照してください。

list-parents

次の例は、list-parents を使用する方法を説明しています。

AWS CLI

アカウントOUsまたは子 OU の親またはルートを一覧表示するには

次の例では、そのアカウント 444444444444 を含むルートまたは親 OU を一覧表示する方法を示します。

```
aws organizations list-parents --child-id 444444444444
```

出力は、指定されたアカウントが指定された ID の OU にあることを示します。

```
{
  "Parents": [
    {
      "Id": "ou-examplerootid111-exampleouid111",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListParents](#)」の「」を参照してください。

list-policies-for-target

次のコード例は、list-policies-for-target を使用する方法を示しています。

AWS CLI

アカウントに直接SCPsアタッチされた のリストを取得するには

次の例は、Filter パラメータで指定された、アカウントに直接アタッチされているすべてのサービスコントロールポリシー (SCPs) のリストを取得する方法を示しています。

```
aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-id 444444444444
```

出力には、ポリシーの概要情報を含むポリシー構造のリストが含まれます。このリストには、OU階層内の場所からの継承のためにアカウントに適用されるポリシーは含まれません。

```
{
  "Policies": [
    {
      "Type": "SERVICE_CONTROL_POLICY",
      "Name": "AllowAllEC2Actions",
      "AwsManaged": false,
      "Id": "p-examplepolicyid222",
      "Arn": "arn:aws:organizations::o-exampleorgid:policy/service_control_policy/p-examplepolicyid222",
      "Description": "Enables account admins to delegate permissions for any EC2 actions to users and roles in their accounts."
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPoliciesForTarget](#)」の「」を参照してください。

list-policies

次の例は、list-policies を使用する方法を説明しています。

AWS CLI

特定のタイプの組織のすべてのポリシーのリストを取得するには

次の例はSCPs、フィルターパラメータで指定された のリストを取得する方法を示しています。

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

出力には、ポリシーのリストと概要情報が含まれます。

```
{
  "Policies": [
    {
      "Type": "SERVICE_CONTROL_POLICY",
      "Name": "AllowAllS3Actions",
      "AwsManaged": false,
      "Id": "p-examplepolicyid111",
      "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid111",
      "Description": "Enables account admins to delegate
permissions for any S3 actions to users and roles in their accounts."
    },
    {
      "Type": "SERVICE_CONTROL_POLICY",
      "Name": "AllowAllEC2Actions",
      "AwsManaged": false,
      "Id": "p-examplepolicyid222",
      "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid222",
      "Description": "Enables account admins to delegate
permissions for any EC2 actions to users and roles in their accounts."
    },
    {
      "AwsManaged": true,
      "Description": "Allows access to every operation",
      "Type": "SERVICE_CONTROL_POLICY",
      "Id": "p-FullAWSAccess",
      "Arn": "arn:aws:organizations::aws:policy/
service_control_policy/p-FullAWSAccess",
      "Name": "FullAWSAccess"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicies](#)」の「」を参照してください。

list-roots

次のコード例は、list-roots を使用する方法を示しています。

AWS CLI

組織内のルートの一覧を取得するには

この例では、組織のルートの一覧を取得する方法を示します。

```
aws organizations list-roots
```

出力には、概要情報を含むルート構造の一覧が含まれます。

```
{
  "Roots": [
    {
      "Name": "Root",
      "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
      "Id": "r-examplerootid111",
      "PolicyTypes": [
        {
          "Status": "ENABLED",
          "Type": "SERVICE_CONTROL_POLICY"
        }
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRoots](#)」の「」を参照してください。

list-targets-for-policy

次の例は、list-targets-for-policy を使用する方法を説明しています。

AWS CLI

ポリシーがアタッチされているルートOUs、およびアカウントの一覧を取得するには

次の例は、指定されたポリシーがアタッチされているルートOUs、およびアカウントの一覧を取得する方法を示しています。

```
aws organizations list-targets-for-policy --policy-id p-FuLLAWSAccess
```


出力には、ポリシーがアタッチされているルート、、OUsおよびアカウントに関する概要情報を含むアタッチメントオブジェクトのリストが含まれます。

```
{
  "Targets": [
    {
      "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
      "Name": "Root",
      "TargetId": "r-examplerootid111",
      "Type": "ROOT"
    },
    {
      "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333;",
      "Name": "Developer Test Account",
      "TargetId": "333333333333",
      "Type": "ACCOUNT"
    },
    {
      "Arn": "arn:aws:organizations::111111111111:ou/o-
exampleorgid/ou-examplerootid111-exampleouid111",
      "Name": "Accounting",
      "TargetId": "ou-examplerootid111-exampleouid111",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTargetsForPolicy](#)」の「」を参照してください。

move-account

次のコード例は、move-account を使用する方法を示しています。

AWS CLI

ルート間または間でアカウントを移動するには OUs

次の例は、組織内のマスターアカウントをルートから OU に移動する方法を示しています。

```
aws organizations move-account --account-id 333333333333 --source-parent-id r-  
examplerootid111 --destination-parent-id ou-examplerootid111-exampleoid111
```

- API 詳細については、AWS CLI 「コマンドリファレンス[MoveAccount](#)」の「」を参照してください。

remove-account-from-organization

次のコード例は、remove-account-from-organization を使用する方法を示しています。

AWS CLI

マスターアカウントとして組織からアカウントを削除するには

次の例は、組織からアカウントを削除する方法を示しています。

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveAccountFromOrganization](#)」の「」を参照してください。

update-organizational-unit

次のコード例は、update-organizational-unit を使用する方法を示しています。

AWS CLI

OU の名前を変更するには

この例では、OU の名前を変更する方法を示します。この例では、OU の名前が「AccountingOU」に変更されます。

```
aws organizations update-organizational-unit --organizational-unit-id ou-  
examplerootid111-exampleoid111 --name AccountingOU
```

出力には新しい名前が表示されます。

```
{  
  "OrganizationalUnit": {
```

```

        "Id": "ou-examplerootid111-exampleoid111"
        "Name": "AccountingOU",
        "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleoid111"
    }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateOrganizationalUnit](#)」の「」を参照してください。

update-policy

次の例は、update-policy を使用する方法を説明しています。

AWS CLI

例 1: ポリシーの名前を変更するには

次のupdate-policy例では、ポリシーの名前を変更し、新しい説明を提供します。

```

aws organizations update-policy \
  --policy-id p-examplepolicyid111 \
  --name Renamed-Policy \
  --description "This description replaces the original."

```

出力には新しい名前と説明が表示されます。

```

{
  "Policy": {
    "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": {\n\n    \"Effect\": \"Allow\",\n    \"Action\": \"ec2:*\",\n    \"Resource\": \"*\"\n  }\n}\n",
    "PolicySummary": {
      "Id": "p-examplepolicyid111",
      "AwsManaged": false,
      "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/
service_control_policy/p-examplepolicyid111",
      "Description": "This description replaces the original.",
      "Name": "Renamed-Policy",
      "Type": "SERVICE_CONTROL_POLICY"
    }
  }
}

```

```
}
```

例 2: ポリシーのJSONテキストコンテンツを置き換えるには

次の例は、SCP前の例の のJSONテキストを、 の代わりに S3 を許可する新しいJSONポリシーテキスト文字列に置き換える方法を示していますEC2。

```
aws organizations update-policy \  
  --policy-id p-examplepolicyid111 \  
  --content "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":\"Allow\",\  
  \"Action\":\"s3:*\",\"Resource\":\"*\"}}"
```

出力には新しいコンテンツが表示されます。

```
{  
  "Policy": {  
    "Content": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":  
  \"Allow\", \"Action\": \"s3:*\", \"Resource\": \"*\" } }",  
    "PolicySummary": {  
      "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/  
service_control_policy/p-examplepolicyid111",  
      "AwsManaged": false;  
      "Description": "This description replaces the original.",  
      "Id": "p-examplepolicyid111",  
      "Name": "Renamed-Policy",  
      "Type": "SERVICE_CONTROL_POLICY"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePolicy](#)」の「」を参照してください。

AWS Outposts を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Outposts。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

get-outpost-instance-types

次の例は、get-outpost-instance-types を使用する方法を説明しています。

AWS CLI

Outpost でインスタンスタイプを取得するには

次のget-outpost-instance-types例では、指定された Outpost のインスタンスタイプを取得します。

```
aws outposts get-outpost-instance-types \  
  --outpost-id op-0ab23c4567EXAMPLE
```

出力:

```
{  
  "InstanceTypes": [  
    {  
      "InstanceType": "c5d.large"  
    },  
    {  
      "InstanceType": "i3en.24xlarge"  
    },  
    {  
      "InstanceType": "m5d.large"  
    },  
    {  
      "InstanceType": "r5d.large"  
    }  
  ],  
  "OutpostId": "op-0ab23c4567EXAMPLE",  
  "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/  
op-0ab23c4567EXAMPLE"
```

```
}
```

詳細については、[「Outposts ユーザーガイド」の「Outpost でインスタンスを起動するAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOutpostInstanceTypes](#)」の「」を参照してください。

get-outpost

次のコード例は、get-outpost を使用する方法を示しています。

AWS CLI

Outpost の詳細を取得するには

次のget-outpost例では、指定された Outpost の詳細を表示します。

```
aws outposts get-outpost \  
  --outpost-id op-0ab23c4567EXAMPLE
```

出力:

```
{  
  "Outpost": {  
    "OutpostId": "op-0ab23c4567EXAMPLE",  
    "OwnerId": "123456789012",  
    "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/  
op-0ab23c4567EXAMPLE",  
    "SiteId": "os-0ab12c3456EXAMPLE",  
    "Name": "EXAMPLE",  
    "LifecycleStatus": "ACTIVE",  
    "AvailabilityZone": "us-west-2a",  
    "AvailabilityZoneId": "usw2-az1",  
    "Tags": {}  
  }  
}
```

詳細については、[「Outposts ユーザーガイド」の「Outposts の使用」](#)を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOutpost](#)」の「」を参照してください。

list-outposts

次のコード例は、list-outposts を使用する方法を示しています。

AWS CLI

Outposts を一覧表示するには

次のlist-outposts例では、AWS アカウントの Outposts を一覧表示します。

```
aws outposts list-outposts
```

出力:

```
{
  "Outposts": [
    {
      "OutpostId": "op-0ab23c4567EXAMPLE",
      "OwnerId": "123456789012",
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE",
      "SiteId": "os-0ab12c3456EXAMPLE",
      "Name": "EXAMPLE",
      "Description": "example",
      "LifecycleStatus": "ACTIVE",
      "AvailabilityZone": "us-west-2a",
      "AvailabilityZoneId": "usw2-az1",
      "Tags": {
        "Name": "EXAMPLE"
      }
    },
    {
      "OutpostId": "op-4fe3dc21baEXAMPLE",
      "OwnerId": "123456789012",
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-4fe3dc21baEXAMPLE",
      "SiteId": "os-0ab12c3456EXAMPLE",
      "Name": "EXAMPLE2",
      "LifecycleStatus": "ACTIVE",
```

```
        "AvailabilityZone": "us-west-2a",
        "AvailabilityZoneId": "usw2-az1",
        "Tags": {}
    }
]
}
```

詳細については、[「Outposts ユーザーガイド」の「Outposts の使用」](#)を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOutposts](#)」の「」を参照してください。

list-sites

次のコード例は、list-sites を使用する方法を示しています。

AWS CLI

サイトを一覧表示するには

次のlist-sites例では、AWS アカウントで使用可能な Outpost サイトを一覧表示します。

```
aws outposts list-sites
```

出力:

```
{
  "Sites": [
    {
      "SiteId": "os-0ab12c3456EXAMPLE",
      "AccountId": "123456789012",
      "Name": "EXAMPLE",
      "Description": "example",
      "Tags": {}
    }
  ]
}
```

詳細については、[「Outposts ユーザーガイド」の「Outposts の使用」](#)を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSites](#)」の「」を参照してください。

AWS Payment Cryptography を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Payment Cryptography。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-alias

次の例は、create-alias を使用する方法を説明しています。

AWS CLI

キーのエイリアスを作成するには

次のcreate-alias例では、キーのエイリアスを作成します。

```
aws payment-cryptography create-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiflw2h
```

出力:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",
```

```
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
    kwapwa6qaifllw2h"
  }
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[エイリアスについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAlias](#)」の「」を参照してください。

create-key

次の例は、create-key を使用する方法を説明しています。

AWS CLI

キーを作成するには

次のcreate-key例では、CVV/CVV2 値の生成と検証に使用できる 2KEY つのTDESキーを生成します。

```
aws payment-cryptography create-key \
  --exportable \
  --key-
attributes KeyAlgorithm=TDES_2KEY, KeyUsage=TR31_C0_CARD_VERIFICATION_KEY, KeyClass=SYMMETRIC
```

出力:

```
{
  "Key": {
    "CreateTimestamp": "1686800690",
    "Enabled": true,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
    kwapwa6qaifllw2h",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_2KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": false,
        "DeriveKey": false,
        "Encrypt": false,
```

```
        "Generate": true,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": false,  
        "Verify": true,  
        "Wrap": false  
    },  
    "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"  
},  
"KeyCheckValue": "F2E50F",  
"KeyCheckValueAlgorithm": "ANSI_X9_24",  
"KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",  
"KeyState": "CREATE_COMPLETE",  
"UsageStartTimestamp": "1686800690"  
}  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーの生成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateKey](#)」の「」を参照してください。

delete-alias

次のコード例は、delete-alias を使用する方法を示しています。

AWS CLI

エイリアスを削除するには

次のdelete-alias例では、エイリアスを削除します。キーには影響しません。

```
aws payment-cryptography delete-alias \  
  --alias-name alias/sampleAlias1
```

このコマンドでは何も出力されません。

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[エイリアスについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAlias](#)」の「」を参照してください。

delete-key

次のコード例は、delete-key を使用する方法を示しています。

AWS CLI

キーを削除するには

次のdelete-key例では、デフォルトの待機期間である 7 日後にキーの削除をスケジュールします。

```
aws payment-cryptography delete-key \  
  --key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/  
  kwapwa6qaifl1w2h
```

出力:

```
{  
  "Key": {  
    "CreateTimestamp": "1686801198",  
    "DeletePendingTimestamp": "1687405998",  
    "Enabled": true,  
    "Exportable": true,  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaifl1w2h",  
    "KeyAttributes": {  
      "KeyAlgorithm": "TDES_2KEY",  
      "KeyClass": "SYMMETRIC_KEY",  
      "KeyModesOfUse": {  
        "Decrypt": false,  
        "DeriveKey": false,  
        "Encrypt": false,  
        "Generate": true,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": false,  
        "Verify": true,  
        "Wrap": false  
      },  
      "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"  
    },  
    "KeyCheckValue": "F2E50F",  
    "KeyCheckValueAlgorithm": "ANSI_X9_24",  
  }  
}
```

```

    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "DELETE_PENDING",
    "UsageStartTimestamp": "1686801190"
  }
}

```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteKey](#)」の「」を参照してください。

export-key

次の例は、export-key を使用する方法を説明しています。

AWS CLI

キーをエクスポートするには

次のexport-key例では、キーをエクスポートします。

```

aws payment-cryptography export-key \
  --export-key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/1co3w6gask7zgu2l \
  --key-material '{"Tr34KeyBlock": { \
    "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-cryptography:us-west-2:123456789012:key/ftobshq7pvioc5fx", \
    "ExportToken": "export-token-cu4lg26ofcziixny", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "WrappingKeyCertificate": file://wrapping-key-certificate.pem }}'

```

wrapping-key-certificate.pem の内容:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ01SQ1ZZS8xMXFUK2svVz1RUDJQOE1V
```

出力:

```

{
  "WrappedKey": {
    "KeyMaterial":
      "308205A106092A864886F70D010702A08205923082058E020101310D300B06096086480165030402013082031F

```

```
    "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
  }
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の[「エクスポートキー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ExportKey](#)」の「」を参照してください。

get-alias

次のコード例は、get-alias を使用する方法を示しています。

AWS CLI

エイリアスを取得するには

次のget-alias例では、エイリアスに関連付けられたキーARNの を返します。

```
aws payment-cryptography get-alias \  
  --alias-name alias/sampleAlias1
```

出力:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaif1lw2h"  
  }  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の[「エイリアスについて」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAlias](#)」の「」を参照してください。

get-key

次の例は、get-key を使用する方法を説明しています。

AWS CLI

キーのメタデータを取得するには

次のget-key例では、エイリアスに関連付けられたキーのメタデータを返します。このオペレーションは暗号化マテリアルを返しません。

```
aws payment-cryptography get-key \  
  --key-identifier alias/sampleAlias1
```

出力:

```
{  
  "Key": {  
    "CreateTimestamp": "1686800690",  
    "DeletePendingTimestamp": "1687405998",  
    "Enabled": true,  
    "Exportable": true,  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaifllw2h",  
    "KeyAttributes": {  
      "KeyAlgorithm": "TDES_2KEY",  
      "KeyClass": "SYMMETRIC_KEY",  
      "KeyModesOfUse": {  
        "Decrypt": false,  
        "DeriveKey": false,  
        "Encrypt": false,  
        "Generate": true,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": false,  
        "Verify": true,  
        "Wrap": false  
      },  
      "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"  
    },  
    "KeyCheckValue": "F2E50F",  
    "KeyCheckValueAlgorithm": "ANSI_X9_24",  
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",  
    "KeyState": "DELETE_PENDING",  
    "UsageStartTimestamp": "1686801190"  
  }  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーの取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetKey](#)」の「」を参照してください。

get-parameters-for-export

次のコード例は、get-parameters-for-export を使用する方法を示しています。

AWS CLI

エクスポートプロセスを初期化するには

次のget-parameters-for-export例では、キーペアを生成し、キーに署名してから、証明書と証明書ルートを返します。

```
aws payment-cryptography get-parameters-for-export \  
  --signing-key-algorithm RSA_2048 \  
  --key-material-type TR34_KEY_BLOCK
```

出力:

```
{  
  "ExportToken": "export-token-ep5cwyzone7oya53",  
  "ParametersValidUntilTimestamp": "1687415640",  
  "SigningKeyAlgorithm": "RSA_2048",  
  "SigningKeyCertificate":  
  
  "MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCMC  
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdB  
BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMCVVMxCzAJBgNVBAGTAldBMRAwDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z  
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGFT  
YXpvbi5jb20wZGZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE  
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUHVvXUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
```



```

"SigningKeyCertificateChain":
"NIICiTCCAfICCD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAd
BqkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAdBqkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAARhhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}

```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[エクスポートキー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParametersForExport](#)」の「」を参照してください。

get-parameters-for-import

次の例は、get-parameters-for-import を使用する方法を説明しています。

AWS CLI

インポートプロセスを初期化するには

次のget-parameters-for-import例では、キーペアを生成し、キーに署名してから、証明書と証明書ルートを返します。

```

aws payment-cryptography get-parameters-for-import \
  --key-material-type TR34_KEY_BLOCK \
  --wrapping-key-algorithm RSA_2048

```

出力:

```

{
  "ImportToken": "import-token-qgmafpa7nt2kfbb",

```

```

"ParametersValidUntilTimestamp": "1687415640",
"WrappingKeyAlgorithm": "RSA_2048",
"WrappingKeyCertificate":
"MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
"WrappingKeyCertificateChain":
"NIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}

```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「キーの[インポート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParametersForImport](#)」の「」を参照してください。

get-public-key-certificate

次の例は、get-public-key-certificate を使用する方法を説明しています。

AWS CLI

パブリックキーを返すには

次のget-public-key-certificate例では、キーペアのパブリックキー部分を返します。

```
aws payment-cryptography get-public-key-certificate \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiFlLw2h
```

出力:

```
{
  "KeyCertificate":
  "MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
  BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySwTC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
  "KeyCertificateChain":
  "MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
  BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySwTC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
```

```
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーペアに関連付けられたパブリックキー/証明書を取得する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPublicKeyCertificate](#)」の「」を参照してください。

import-key

次の例は、import-key を使用する方法を説明しています。

AWS CLI

TR-34 キーをインポートするには

次のimport-key例では、TR-34 キーをインポートします。

```
aws payment-cryptography import-key \
  --key-material='{ "Tr34KeyBlock": {" \
    CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-west-2:123456789012:key/rmm5wn2q564nijnm", \
    "ImportToken": "import-token-5ott6ho5nts7bbc", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "SigningKeyCertificate": file://signing-key-certificate.pem, \
    "WrappedKeyBlock": file://wrapped-key-block.pem } }'
```

signing-key-certificate.pem の内容:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ01RYWVCK25IbE1WZU1PR1ZiNjU1Q2Jz
```

wrapped-key-block.pem の内容:

```
3082059806092A864886F70D010702A082058930820585020101310D300B06096086480165030402013082031606
```

出力:

```
{
  "Key": {
    "CreateTimestamp": "2023-06-09T16:56:27.621000-07:00",
```

```
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
bzmvgyx dg3sktwdx",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_2KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": false,
        "DeriveKey": false,
        "Encrypt": false,
        "Generate": true,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": false,
        "Verify": true,
        "Wrap": false
      },
      "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
    },
    "KeyCheckValue": "D9B20E",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "EXTERNAL",
    "KeyState": "CREATE_COMPLETE",
    "UsageStartTimestamp": "2023-06-09T16:56:27.621000-07:00"
  }
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「キーの[インポート](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportKey](#)」の「」を参照してください。

list-aliases

次の例は、list-aliases を使用する方法を説明しています。

AWS CLI

エイリアスのリストを取得するには

次のlist-aliases例は、このリージョンのアカウント内のすべてのエイリアスを示しています。

```
aws payment-cryptography list-aliases
```

出力:

```
{
  "Aliases": [
    {
      "AliasName": "alias/sampleAlias1",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/kwapwa6qaiif1lw2h"
    },
    {
      "AliasName": "alias/sampleAlias2",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/kwapwa6qaiif1lw2h"
    }
  ]
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[エイリアスについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAliases](#)」の「」を参照してください。

list-keys

次のコード例は、list-keys を使用する方法を示しています。

AWS CLI

キーのリストを取得するには

次のlist-keys例は、このリージョンのアカウントのすべてのキーを示しています。

```
aws payment-cryptography list-keys
```

出力:

```
{
  "Keys": [
    {
```

```
    "CreateTimestamp": "1666506840",
    "Enabled": false,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiifllw2h",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_3KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": true,
        "DeriveKey": false,
        "Encrypt": true,
        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": true,
        "Verify": false,
        "Wrap": true
      },
      "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
    },
    "KeyCheckValue": "369D",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "CREATE_COMPLETE",
    "UsageStopTimestamp": "1666938840"
  }
]
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーを一覧表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListKeys](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

キーのタグのリストを取得するには

次のlist-tags-for-resource例では、キーのタグを取得します。

```
aws payment-cryptography list-tags-for-resource \  
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "BIN",  
      "Value": "20151120"  
    },  
    {  
      "Key": "Project",  
      "Value": "Production"  
    }  
  ]  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[APIオペレーションによるキータグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

restore-key

次の例は、restore-key を使用する方法を説明しています。

AWS CLI

削除が予定されているキーを復元するには

次のrestore-key例では、キーの削除をキャンセルします。

```
aws payment-cryptography restore-key \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h
```


出力:

```
{
  "Key": {
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyAttributes": {
      "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyAlgorithm": "TDES_3KEY",
      "KeyModesOfUse": {
        "Encrypt": false,
        "Decrypt": false,
        "Wrap": false,
        "Unwrap": false,
        "Generate": true,
        "Sign": false,
        "Verify": true,
        "DeriveKey": false,
        "NoRestrictions": false
      }
    },
    "KeyCheckValue": "",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "Enabled": false,
    "Exportable": true,
    "KeyState": "CREATE_COMPLETE",
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "CreateTimestamp": "1686800690",
    "UsageStopTimestamp": "1687405998"
  }
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreKey](#)」の「」を参照してください。

start-key-usage

次のコード例は、start-key-usage を使用する方法を示しています。

AWS CLI

キーを有効にするには

次のstart-key-usage例では、キーの使用を有効にします。

```
aws payment-cryptography start-key-usage \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h
```

出力:

```
{  
  "Key": {  
    "CreateTimestamp": "1686800690",  
    "Enabled": true,  
    "Exportable": true,  
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
alsuwxug3pgy6xh",  
    "KeyAttributes": {  
      "KeyAlgorithm": "TDES_3KEY",  
      "KeyClass": "SYMMETRIC_KEY",  
      "KeyModesOfUse": {  
        "Decrypt": true,  
        "DeriveKey": false,  
        "Encrypt": true,  
        "Generate": false,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": true,  
        "Verify": false,  
        "Wrap": true  
      },  
      "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"  
    },  
    "KeyCheckValue": "369D",  
    "KeyCheckValueAlgorithm": "ANSI_X9_24",  
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",  
    "KeyState": "CREATE_COMPLETE",  
    "UsageStartTimestamp": "1686800690"  
  }  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーの有効化と無効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartKeyUsage](#)」の「」を参照してください。

stop-key-usage

次のコード例は、stop-key-usage を使用する方法を示しています。

AWS CLI

キーを無効にするには

次のstop-key-usage例では、キーを無効にします。

```
aws payment-cryptography stop-key-usage \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h
```

出力:

```
{  
  "Key": {  
    "CreateTimestamp": "1686800690",  
    "Enabled": true,  
    "Exportable": true,  
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
alsuwfxug3pgy6xh",  
    "KeyAttributes": {  
      "KeyAlgorithm": "TDES_3KEY",  
      "KeyClass": "SYMMETRIC_KEY",  
      "KeyModesOfUse": {  
        "Decrypt": true,  
        "DeriveKey": false,  
        "Encrypt": true,  
        "Generate": false,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": true,  
        "Verify": false,  
        "Wrap": true  
      }  
    }  
  },  
}
```

```
        "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
    },
    "KeyCheckValue": "369D",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "CREATE_COMPLETE",
    "UsageStartTimestamp": "1686800690"
}
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キーの有効化と無効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopKeyUsage](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

キーにタグを付けるには

次のtag-resource例では、キーにタグを付けます。

```
aws payment-cryptography tag-resource \
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiflw2h \
  --tags Key=sampleTag,Value=sampleValue
```

このコマンドでは何も出力されません。

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キータグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

キーからタグを削除するには

次の`untag-resource`例では、キーからタグを削除します。

```
aws payment-cryptography untag-resource \  
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaiFlLw2h \  
  --tag-keys sampleTag
```

このコマンドでは何も出力されません。

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[キータグの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-alias

次の例は、`update-alias` を使用する方法を説明しています。

AWS CLI

エイリアスを更新するには

次の`update-alias`例では、エイリアスを別のキーに関連付けます。

```
aws payment-cryptography update-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
tqv5yij6wtxx64pi
```

出力:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
tqv5yij6wtxx64pi "  
  }  
}
```

```
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[エイリアスについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAlias](#)」の「」を参照してください。

AWS Payment Cryptography を使用したデータプレーンの例 AWS CLI

次のコード例は、AWS Payment Cryptography Data Plane AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

decrypt-data

次の例は、decrypt-data を使用する方法を説明しています。

AWS CLI

暗号文を復号するには

次のdecrypt-data例では、対称キーを使用して暗号文データを復号化します。このオペレーションでは、キーが KeyModesOfUseに設定Decryptされ、KeyUsageに設定されている必要がありますTR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY。

```
aws payment-cryptography-data decrypt-data \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h \  
  --cipher-text 33612AB9D6929C3A828EB6030082B2BD \  
  --output-type text
```

```
--decryption-attributes 'Symmetric={Mode=CBC}'
```

出力:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
  kwapwa6qaifl1w2h",
  "KeyCheckValue": "71D7AE",
  "PlainText": "31323334313233343132333431323334"
}
```

詳細については、AWS「[Payment Cryptography ユーザーガイド](#)」の「[データを復号する](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス DecryptData](#)」の「」を参照してください。

encrypt-data

次の例は、encrypt-data を使用する方法を説明しています。

AWS CLI

データを暗号化するには

次のencrypt-data例では、対称キーを使用してプレーンテキストデータを暗号化します。このオペレーションでは、キーが KeyModesOfUseに設定Encryptされ、KeyUsageに設定されている必要がありますTR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY。

```
aws payment-cryptography-data encrypt-data \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
  kwapwa6qaifl1w2h \
  --plain-text 31323334313233343132333431323334 \
  --encryption-attributes 'Symmetric={Mode=CBC}'
```

出力:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
  kwapwa6qaifl1w2h",
  "KeyCheckValue": "71D7AE",
  "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

```
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の[「データの暗号化」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EncryptData](#)」の「」を参照してください。

generate-card-validation-data

次の例は、generate-card-validation-data を使用する方法を説明しています。

AWS CLI

を生成するには CVV

次のgenerate-card-validation-data例では、CVV/ を生成しますCVV2。

```
aws payment-cryptography-data generate-card-validation-data \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaifllw2h \  
  --primary-account-number=171234567890123 \  
  --generation-attributes CardVerificationValue2={CardExpiryDate=0123}
```

出力:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaifllw2h",  
  "KeyCheckValue": "CADD1",  
  "ValidationData": "801"  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の[「カードデータの生成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateCardValidationData](#)」の「」を参照してください。

generate-mac

次のコード例は、generate-mac を使用する方法を示しています。

AWS CLI

を生成するには MAC

次のgenerate-card-validation-data例では、アルゴリズム HMAC_SHA256 とHMAC暗号化キーを使用してカードデータ認証用のハッシュベースのメッセージ認証コード (HMAC) を生成します。キーは `KeyUsage`、`TR31_M7_HMAC_KEY` は `KeyModesOfUse` に設定されている必要がありますGenerate。

```
aws payment-cryptography-data generate-mac \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaifllw2h \  
  --message-  
data "3b313038383439303031303733393431353d32343038323236303030373030303f33" \  
  --generation-attributes Algorithm=HMAC_SHA256
```

出力:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifllw2h,  
  "KeyCheckValue": "2976E7",  
  "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[生成MAC](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateMac](#)」の「」を参照してください。

generate-pin-data

次の例は、generate-pin-data を使用する方法を説明しています。

AWS CLI

を生成するには PIN

次のgenerate-card-validation-data例では、Visa PINスキームPINを使用して新しいランダムを生成します。

```
aws payment-cryptography-data generate-pin-data \  
  --generation-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2 \  
  --encryption-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt \  
  --primary-account-number 171234567890123 \  
  --pin-block-format ISO_FORMAT_0 \  
  --generation-attributes VisaPin={PinVerificationKeyIndex=1}
```

出力:

```
{  
  "GenerationKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2",  
  "GenerationKeyCheckValue": "7F2363",  
  "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt",  
  "EncryptionKeyCheckValue": "7CC9E2",  
  "EncryptedPinBlock": "AC17DC148BDA645E",  
  "PinData": {  
    "VerificationValue": "5507"  
  }  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の [PIN「データの生成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GeneratePinData](#)」の「」を参照してください。

re-encrypt-data

次の例は、re-encrypt-data を使用する方法を説明しています。

AWS CLI

別のキーを使用してデータを再暗号化するには

次のre-encrypt-data例では、AES対称キーを使用して暗号化された暗号テキストを復号し、トランザクションあたりの派生一意キー (DUKPT) キーを使用して再暗号化します。

```
aws payment-cryptography-data re-encrypt-data \  

```

```

--incoming-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/hyv7ymboitd4vfy \
--outgoing-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/jl6ythkcvzesbxen \
--cipher-
text 4D2B0BDBA192D5AEFEAA5B3EC28E4A65383C313FFA25140101560F75FE1B99F27192A90980AB9334
\
--incoming-encryption-
attributes "Dukpt={Mode=ECB,KeySerialNumber=012345678911111}" \
--outgoing-encryption-attributes '{"Symmetric": {"Mode": "ECB"}}'

```

出力:

```

{
  "CipherText":
    "F94959DA30EEFF0C035483C6067667CF6796E3C1AD28C2B61F9CFEB772A8DD41C0D6822931E0D3B1",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
jl6ythkcvzesbxen",
  "KeyCheckValue": "2E8CD9"
}

```

詳細については、AWS「[Payment Cryptography ユーザーガイド](#)」の「[データの暗号化と復号化](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス `ReEncryptData`](#)」の「」を参照してください。

translate-pin-data

次のコード例は、translate-pin-data を使用する方法を示しています。

AWS CLI

PINデータを翻訳するには

次のtranslate-pin-data例では、を 0 ISO PINブロックを使用したPEKTDDES暗号化PINからDUKPTアルゴリズムを使用した 4 AES ISO PINブロックに変換します。

```

aws payment-cryptography-data translate-pin-data \
--encrypted-pin-block "AC17DC148BDA645E" \
--incoming-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}' \

```

```
--incoming-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt \
--outgoing-key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/4pmyquwjs3yj4vwe \
--outgoing-translation-attributes
IsoFormat4="{PrimaryAccountNumber=171234567890123}" \
--outgoing-dukpt-attributes KeySerialNumber="FFFF9876543210E00008"
```

出力:

```
{
  "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt",
  "KeyCheckValue": "7CC9E2"
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の [PIN「データの変換」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TranslatePinData](#)」の「」を参照してください。

verify-auth-request-cryptogram

次の例は、verify-auth-request-cryptogram を使用する方法を説明しています。

AWS CLI

認証リクエストを検証するには

次のverify-auth-request-cryptogram例では、認証リクエストの暗号 () を検証します ARQC。

```
aws payment-cryptography-data verify-auth-request-cryptogram \
--auth-request-cryptogram FGE1BD1E6037FB3E \
--auth-response-attributes '{"ArpcMethod1": {"AuthResponseCode": "1111"}}' \
--key-identifier arn:aws:payment-cryptography:us-west-2:111122223333:key/pboipdfzd4mdklya \
--major-key-derivation-mode "EMV_OPTION_A" \
--session-key-derivation-attributes '{"EmvCommon": {"ApplicationTransactionCounter": "1234", "PanSequenceNumber": "01", "PrimaryAccountNumber": "471234567890123"}}' \
```

```
--transaction-data "123456789ABCDEF"
```

出力:

```
{
  "AuthResponseValue": "D899B8C6FBF971AA",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya",
  "KeyCheckValue": "985792"
}
```

詳細については、「Payment Cryptography ユーザーガイド」の「[認証リクエスト \(ARQC\) 暗号文の検証](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[VerifyAuthRequestCryptogram](#)」の「」を参照してください。

verify-card-validation-data

次のコード例は、verify-card-validation-data を使用する方法を示しています。

AWS CLI

を検証するには CVV

次のverify-card-validation-data例では、 の CVV/CVV2 を検証しますPAN。

```
aws payment-cryptography-data verify-card-validation-data \
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi \
  --primary-account-number=171234567890123 \
  --verification-attributes CardVerificationValue2={CardExpiryDate=0123} \
  --validation-data 801
```

出力:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
  "KeyCheckValue": "CADD1"
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[カードデータの検証](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[VerifyCardValidationData](#)」の「」を参照してください。

verify-mac

次の例は、verify-mac を使用する方法を説明しています。

AWS CLI

を検証するには MAC

次のverify-mac例では、アルゴリズム HMAC_SHA256 とHMAC暗号化キーを使用して、カードデータ認証用のハッシュベースのメッセージ認証コード (HMAC) を検証します。

```
aws payment-cryptography-data verify-mac \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qno15lghrzunce6 \  
  --message-  
data "3b343038383439303031303733393431353d32343038323236303030373030303f33" \  
  --verification-attributes='Algorithm=HMAC_SHA256' \  
  --mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

出力:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qno15lghrzunce6,  
  "KeyCheckValue": "2976E7",  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の「[VerifyMAC](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[VerifyMac](#)」の「」を参照してください。

verify-pin-data

次の例は、verify-pin-data を使用する方法を説明しています。

AWS CLI

を検証するには PIN

次のverify-pin-data例では、PINの を検証しますPAN。

```
aws payment-cryptography-data verify-pin-data \  
  --verification-key-identifier arn:aws:payment-cryptography:us-  
east-2:111122223333:key/37y2tsl45p5zjbh2 \  
  --encryption-key-identifier arn:aws:payment-cryptography:us-  
east-2:111122223333:key/ivi5ksfsuplneuyt \  
  --primary-account-number 171234567890123 \  
  --pin-block-format ISO_FORMAT_0 \  
  --verification-attributes  
  VisaPin="{PinVerificationKeyIndex=1,VerificationValue=5507}" \  
  --encrypted-pin-block AC17DC148BDA645E
```

出力:

```
{  
  "VerificationKeyArn": "arn:aws:payment-cryptography:us-  
east-2:111122223333:key/37y2tsl45p5zjbh2",  
  "VerificationKeyCheckValue": "7F2363",  
  "EncryptionKeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
ivi5ksfsuplneuyt",  
  "EncryptionKeyCheckValue": "7CC9E2",  
}
```

詳細については、AWS 「Payment Cryptography ユーザーガイド」の [PIN「データの検証」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [VerifyPinData](#)」の「」を参照してください。

を使用した Amazon Pinpoint の例 AWS CLI

次のコード例は、Amazon Pinpoint AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-app

次のコード例は、create-app を使用する方法を示しています。

AWS CLI

例 1: アプリケーションを作成するには

次の create-app の例は、新しいアプリケーション (プロジェクト) を作成します。

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp
```

出力:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {}  
  }  
}
```

例 2: タグが付けられたアプリケーションを作成するには

次の create-app の例は、新しいアプリケーション (プロジェクト) を作成し、タグ (キーと値) をアプリケーションに関連付けます。

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```


出力:

```
{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {
      "Stack": "Test"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateApp](#)」の「」を参照してください。

create-sms-template

次のコード例は、create-sms-template を使用する方法を示しています。

AWS CLI

SMSチャネルを介して送信されるメッセージのメッセージテンプレートを作成します。

次のcreate-sms-template例では、SMSメッセージテンプレートを作成します。

```
aws pinpoint create-sms-template \
  --template-name TestTemplate \
  --sms-template-request file://myfile.json \
  --region us-east-1
```

myfile.json の内容:

```
{
  "Body": "hello\n how are you?\n food is good",
  "TemplateDescription": "Test SMS Template"
}
```

出力:

```
{
```

```
"CreateTemplateMessageBody": {
  "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
  "Message": "Created",
  "RequestID": "8c36b17f-a0b0-400f-ac21-29e9b62a975d"
}
}
```

詳細については、[Amazon Pinpoint ユーザーガイド](#)の「[Amazon Pinpoint メッセージテンプレート](#)」を参照してください。Amazon Pinpoint

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSmsTemplate](#)」の「」を参照してください。

delete-app

次のコード例は、delete-app を使用する方法を示しています。

AWS CLI

アプリケーションを削除するには

次の delete-app の例は、アプリケーション (プロジェクト) を削除します。

```
aws pinpoint delete-app \
  --application-id 810c7aab86d42fb2b56c8c966example
```

出力:

```
{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {}
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteApp](#)」の「」を参照してください。

get-apns-channel

次のコード例は、get-apns-channel を使用する方法を示しています。

AWS CLI

APNsアプリケーションのステータスと設定に関する情報を取得するには

次のget-apns-channel例では、アプリケーションのステータスとAPNs設定に関する情報を取得します。

```
aws pinpoint get-apns-channel \  
  --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \  
  --region us-east-1
```

出力:

```
{  
  "APNSChannelResponse": {  
    "ApplicationId": "9ab1068eb0a6461c86cce7f27ce0efd7",  
    "CreationDate": "2019-05-09T21:54:45.082Z",  
    "DefaultAuthenticationMethod": "CERTIFICATE",  
    "Enabled": true,  
    "HasCredential": true,  
    "HasTokenKey": false,  
    "Id": "apns",  
    "IsArchived": false,  
    "LastModifiedDate": "2019-05-09T22:04:01.067Z",  
    "Platform": "APNS",  
    "Version": 2  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApnsChannel](#)」の「」を参照してください。

get-app

次のコード例は、get-app を使用する方法を示しています。

AWS CLI

アプリケーション (プロジェクト) に関する情報を取得するには

次のget-app例では、アプリケーション (プロジェクト) に関する情報を取得します。

```
aws pinpoint get-app \  
  --application-id 810c7aab86d42fb2b56c8c966example \  
  --region us-east-1
```

出力:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {  
      "Year": "2019",  
      "Stack": "Production"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApp](#)」の「」を参照してください。

get-apps

次のコード例は、get-apps を使用する方法を示しています。

AWS CLI

すべてのアプリケーションに関する情報を取得するには

次のget-apps例では、すべてのアプリケーション (プロジェクト) に関する情報を取得します。

```
aws pinpoint get-apps
```

出力:

```
{
  "ApplicationsResponse": {
    "Item": [
      {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
          "Year": "2019",
          "Stack": "Production"
        }
      },
      {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/42d8c7eb0990a57ba1d5476a3example",
        "Id": "42d8c7eb0990a57ba1d5476a3example",
        "Name": "AnyCompany",
        "tags": {}
      },
      {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/80f5c382b638ffe5ad12376bbexample",
        "Id": "80f5c382b638ffe5ad12376bbexample",
        "Name": "ExampleCorp_Test",
        "tags": {
          "Year": "2019",
          "Stack": "Test"
        }
      }
    ],
    "NextToken":
    "eyJJDCmVhdGlvbkRhdGUiOiIyMDE5LTA3LTE2VDE0jM40jUzljkwM1oiLCJBY2NvdW50SWQiOiI1MTIzOTcxODM4Nz"
  }
}
```

NextToken レスポンス値の存在は、使用可能な出力が多いことを示します。コマンドを再度呼び出し、その値をNextToken入力パラメータとして指定します。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetApps](#)」の「」を参照してください。

get-campaign

次の例は、get-campaign を使用する方法を説明しています。

AWS CLI

キャンペーンのステータス、設定、その他の設定に関する情報を取得するには

次のget-campaign例では、キャンペーンのステータス、設定、その他の設定に関する情報を取得します。

```
aws pinpoint get-campaign \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --campaign-id a1e63c6cc0eb43ed826ffcc3cc90b30d \  
  --region us-east-1
```

出力:

```
{  
  "CampaignResponse": {  
    "AdditionalTreatments": [],  
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/a1e63c6cc0eb43ed826ffcc3cc90b30d",  
    "CreationDate": "2019-10-08T18:40:16.581Z",  
    "Description": " ",  
    "HoldoutPercent": 0,  
    "Id": "a1e63c6cc0eb43ed826ffcc3cc90b30d",  
    "IsPaused": false,  
    "LastModifiedDate": "2019-10-08T18:40:16.581Z",  
    "Limits": {  
      "Daily": 0,  
      "MaximumDuration": 60,  
      "MessagesPerSecond": 50,  
      "Total": 0  
    },  
    "MessageConfiguration": {  
      "EmailMessage": {  
        "FromAddress": "sender@example.com",  
        "HtmlBody": "<!DOCTYPE html>\n <html lang=\"en\">\n <head>\n <meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" />\n</head>\n<body>Hello</body>\n</html>",
```

```
        "Title": "PinpointDemo"
      }
    },
    "Name": "MyCampaign",
    "Schedule": {
      "IsLocalTime": false,
      "StartTime": "IMMEDIATE",
      "Timezone": "utc"
    },
    "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
    "SegmentVersion": 1,
    "State": {
      "CampaignStatus": "COMPLETED"
    },
    "tags": {},
    "TemplateConfiguration": {},
    "Version": 1
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCampaign](#)」の「」を参照してください。

get-campaigns

次の例は、get-campaigns を使用する方法を説明しています。

AWS CLI

アプリケーションに関連付けられているすべてのキャンペーンのステータス、設定、およびその他の設定に関する情報を取得するには

次のget-campaigns例では、アプリケーションに関連付けられているすべてのキャンペーンのステータス、設定、およびその他の設定に関する情報を取得します。

```
aws pinpoint get-campaigns \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

出力:

```
{
```

```

"CampaignsResponse": {
  "Item": [
    {
      "AdditionalTreatments": [],
      "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
      "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/
campaigns/7e1280344c8f4a9aa40a00b006fe44f1",
      "CreationDate": "2019-10-08T18:40:22.905Z",
      "Description": " ",
      "HoldoutPercent": 0,
      "Id": "7e1280344c8f4a9aa40a00b006fe44f1",
      "IsPaused": false,
      "LastModifiedDate": "2019-10-08T18:40:22.905Z",
      "Limits": {},
      "MessageConfiguration": {
        "EmailMessage": {
          "FromAddress": "sender@example.com",
          "HtmlBody": "<!DOCTYPE html>\n  <html lang=\"en
\n  <head>\n  <meta http-equiv=\"Content-Type\" content=\"text/html;
charset=utf-8\" />\n</head>\n<body>Hello</body>\n</html>",
          "Title": "PinpointDemo Test"
        }
      },
      "Name": "MyCampaign1",
      "Schedule": {
        "IsLocalTime": false,
        "QuietTime": {},
        "StartTime": "IMMEDIATE",
        "Timezone": "UTC"
      },
      "SegmentId": "b66c9e42f71444b2aa2e0fffc1df28f60",
      "SegmentVersion": 1,
      "State": {
        "CampaignStatus": "COMPLETED"
      },
      "tags": {},
      "TemplateConfiguration": {},
      "Version": 1
    },
    {
      "AdditionalTreatments": [],
      "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",

```



```

    "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/6e0b7591a90841d2b5d93fa11143e5a7/campaigns/
a1e63c6cc0eb43ed826ffcc3cc90b30d",
    "CreationDate": "2019-10-08T18:40:16.581Z",
    "Description": " ",
    "HoldoutPercent": 0,
    "Id": "a1e63c6cc0eb43ed826ffcc3cc90b30d",
    "IsPaused": false,
    "LastModifiedDate": "2019-10-08T18:40:16.581Z",
    "Limits": {
      "Daily": 0,
      "MaximumDuration": 60,
      "MessagesPerSecond": 50,
      "Total": 0
    },
    "MessageConfiguration": {
      "EmailMessage": {
        "FromAddress": "sender@example.com",
        "HtmlBody": "<!DOCTYPE html>\n  <html lang=\"en
\n  <head>\n  <meta http-equiv=\"Content-Type\" content=\"text/html;
charset=utf-8\" />\n</head>\n<body>Demo</body>\n</html>",
        "Title": "PinpointDemo"
      }
    },
    "Name": "MyCampaign2",
    "Schedule": {
      "IsLocalTime": false,
      "StartTime": "IMMEDIATE",
      "Timezone": "utc"
    },
    "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
    "SegmentVersion": 1,
    "State": {
      "CampaignStatus": "COMPLETED"
    },
    "tags": {},
    "TemplateConfiguration": {},
    "Version": 1
  }
]
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCampaigns](#)」の「」を参照してください。

get-channels

次の例は、get-channels を使用する方法を説明しています。

AWS CLI

アプリケーションの各チャンネルの履歴とステータスに関する情報を取得するには

次のget-channels例では、アプリケーションの各チャンネルの履歴とステータスに関する情報を取得します。

```
aws pinpoint get-channels \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

出力:

```
{  
  "ChannelsResponse": {  
    "Channels": {  
      "GCM": {  
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
        "CreationDate": "2019-10-08T18:28:23.182Z",  
        "Enabled": true,  
        "HasCredential": true,  
        "Id": "gcm",  
        "IsArchived": false,  
        "LastModifiedDate": "2019-10-08T18:28:23.182Z",  
        "Version": 1  
      },  
      "SMS": {  
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
        "CreationDate": "2019-10-08T18:39:18.511Z",  
        "Enabled": true,  
        "Id": "sms",  
        "IsArchived": false,  
        "LastModifiedDate": "2019-10-08T18:39:18.511Z",  
        "Version": 1  
      }  
    }  
  }  
}
```

```
    "EMAIL": {
      "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
      "CreationDate": "2019-10-08T18:27:23.990Z",
      "Enabled": true,
      "Id": "email",
      "IsArchived": false,
      "LastModifiedDate": "2019-10-08T18:27:23.990Z",
      "Version": 1
    },
    "IN_APP": {
      "Enabled": true,
      "IsArchived": false,
      "Version": 0
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetChannels](#)」の「」を参照してください。

get-email-channel

次のコード例は、get-email-channel を使用する方法を示しています。

AWS CLI

アプリケーション用の E メールチャネルのステータスと設定に関する情報を取得するには

次のget-email-channel例では、アプリケーションの E メールチャネルのステータスと設定を取得します。

```
aws pinpoint get-email-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

出力:

```
{
  "EmailChannelResponse": {
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
```

```

    "CreationDate": "2019-10-08T18:27:23.990Z",
    "Enabled": true,
    "FromAddress": "sender@example.com",
    "Id": "email",
    "Identity": "arn:aws:ses:us-east-1:AIDACKCEVSQ6C2EXAMPLE:identity/
sender@example.com",
    "IsArchived": false,
    "LastModifiedDate": "2019-10-08T18:27:23.990Z",
    "MessagesPerSecond": 1,
    "Platform": "EMAIL",
    "RoleArn": "arn:aws:iam::AIDACKCEVSQ6C2EXAMPLE:role/pinpoint-events",
    "Version": 1
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEmailChannel](#)」の「」を参照してください。

get-endpoint

次の例は、get-endpoint を使用する方法を説明しています。

AWS CLI

アプリケーションの特定のエンドポイントの設定と属性に関する情報を取得するには

次の get-endpoint の例は、アプリケーションの指定されたエンドポイントの設定と属性に関する情報を取得します。

```

aws pinpoint get-endpoint \
  --application-id 611e3e3cdd47474c9c1399a505665b91 \
  --endpoint-id testendpoint \
  --region us-east-1

```

出力:

```

{
  "EndpointResponse": {
    "Address": "+11234567890",
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
    "Attributes": {},

```

```
"ChannelType": "SMS",
"CohortId": "63",
"CreationDate": "2019-01-28T23:55:11.534Z",
"EffectiveDate": "2021-08-06T00:04:51.763Z",
"EndpointStatus": "ACTIVE",
"Id": "testendpoint",
"Location": {
  "Country": "USA"
},
"Metrics": {
  "SmsDelivered": 1.0
},
"OptOut": "ALL",
"RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",
"User": {
  "UserAttributes": {
    "Age": [
      "24"
    ]
  },
  "UserId": "testuser"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEndpoint](#)」の「」を参照してください。

get-gcm-channel

次のコード例は、get-gcm-channel を使用する方法を示しています。

AWS CLI

GCMアプリケーションのステータスと設定に関する情報を取得するには

次のget-gcm-channel例では、アプリケーションのステータスとGCM設定に関する情報を取得します。

```
aws pinpoint get-gcm-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

出力:

```
{
  "GCMChannelResponse": {
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
    "CreationDate": "2019-10-08T18:28:23.182Z",
    "Enabled": true,
    "HasCredential": true,
    "Id": "gcm",
    "IsArchived": false,
    "LastModifiedDate": "2019-10-08T18:28:23.182Z",
    "Platform": "GCM",
    "Version": 1
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGcmChannel](#)」の「」を参照してください。

get-sms-channel

次のコード例は、get-sms-channel を使用する方法を示しています。

AWS CLI

SMSアプリケーションのステータスと設定に関する情報を取得するには

次の get-sms-channel の例は、アプリケーションの SMS チャンネルのステータスと設定を取得します。

```
aws pinpoint get-sms-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

出力:

```
{
  "SMSChannelResponse": {
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
    "CreationDate": "2019-10-08T18:39:18.511Z",
    "Enabled": true,
```

```
    "Id": "sms",
    "IsArchived": false,
    "LastModifiedDate": "2019-10-08T18:39:18.511Z",
    "Platform": "SMS",
    "PromotionalMessagesPerSecond": 20,
    "TransactionalMessagesPerSecond": 20,
    "Version": 1
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSmsChannel](#)」の「」を参照してください。

get-sms-template

次の例は、get-sms-template を使用する方法を説明しています。

AWS CLI

SMSチャネルを介して送信されるメッセージのメッセージテンプレートの内容と設定を取得します。

次のget-sms-template例では、SMSメッセージテンプレートの内容と設定を取得します。

```
aws pinpoint get-sms-template \
  --template-name TestTemplate \
  --region us-east-1
```

出力:

```
{
  "SMSTemplateResponse": {
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
    "Body": "hello\n how are you?\n food is good",
    "CreationDate": "2023-06-20T21:37:30.124Z",
    "LastModifiedDate": "2023-06-20T21:37:30.124Z",
    "tags": {},
    "TemplateDescription": "Test SMS Template",
    "TemplateName": "TestTemplate",
    "TemplateType": "SMS",
    "Version": "1"
  }
}
```

```
}  
}
```

詳細については、[Amazon Pinpoint ユーザーガイド](#)の「[Amazon Pinpoint メッセージテンプレート](#)」を参照してください。Amazon Pinpoint

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSmsTemplate](#)」の「」を参照してください。

get-voice-channel

次の例は、get-voice-channel を使用する方法を説明しています。

AWS CLI

アプリケーションの音声チャンネルのステータスと設定に関する情報を取得するには

次のget-voice-channel例では、アプリケーションの音声チャンネルのステータスと設定を取得します。

```
aws pinpoint get-voice-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

出力:

```
{  
  "VoiceChannelResponse": {  
    "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",  
    "CreationDate": "2022-04-28T00:17:03.836Z",  
    "Enabled": true,  
    "Id": "voice",  
    "IsArchived": false,  
    "LastModifiedDate": "2022-04-28T00:17:03.836Z",  
    "Platform": "VOICE",  
    "Version": 1  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVoiceChannel](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのタグのリストを取得するには

次のlist-tags-for-resource例では、指定されたリソースに関連付けられているすべてのタグ (キー名と値) を取得します。

```
aws pinpoint list-tags-for-resource \  
  --resource-arn arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example
```

出力:

```
{  
  "TagsModel": {  
    "tags": {  
      "Year": "2019",  
      "Stack": "Production"  
    }  
  }  
}
```

詳細については、「Amazon Pinpoint デベロッパーガイド」のAmazon Pinpoint リソース <<https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>>の「タグ付け」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

phone-number-validate

次のコード例は、phone-number-validate を使用する方法を示しています。

AWS CLI

電話番号に関する情報を取得します。

以下は、電話番号に関する情報 `phone-number-validate` を取得します。

```
aws pinpoint phone-number-validate \  
  --number-validate-request PhoneNumber="+12065550142" \  
  --region us-east-1
```

出力:

```
{  
  "NumberValidateResponse": {  
    "Carrier": "ExampleCorp Mobile",  
    "City": "Seattle",  
    "CleansedPhoneNumberE164": "+12065550142",  
    "CleansedPhoneNumberNational": "2065550142",  
    "Country": "United States",  
    "CountryCodeIso2": "US",  
    "CountryCodeNumeric": "1",  
    "OriginalPhoneNumber": "+12065550142",  
    "PhoneType": "MOBILE",  
    "PhoneTypeCode": 0,  
    "Timezone": "America/Los_Angeles",  
    "ZipCode": "98101"  
  }  
}
```

詳細については、[Amazon Pinpoint ユーザーガイド](#)の「[Amazon Pinpoint SMSチャンネル](#)」を参照してください。Amazon Pinpoint

- API 詳細については、AWS CLI 「コマンドリファレンス[PhoneNumberValidate](#)」の「」を参照してください。

send-messages

次のコード例は、`send-messages` を使用する方法を示しています。

AWS CLI

アプリケーションのエンドポイントを使用してSMSメッセージを送信するには

次の `send-messages` の例は、エンドポイントを使用してアプリケーションにダイレクトメッセージを送信します。

```
aws pinpoint send-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --message-request file://myfile.json \  
  --region us-west-2
```

myfile.json の内容:

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Endpoints": {  
    "testendpoint": {}  
  }  
}
```

出力:

```
{  
  "MessageResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "EndpointResult": {  
      "testendpoint": {  
        "Address": "+12345678900",  
        "DeliveryStatus": "SUCCESSFUL",  
        "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",  
        "StatusCode": 200,  
        "StatusMessage": "MessageId:  
itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"  
      }  
    },  
    "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"  
  }  
}
```

詳細については、[Amazon Pinpoint ユーザーガイド](#)の「Amazon Pinpoint SMSチャンネル」を参照してください。 Amazon Pinpoint

- API 詳細については、AWS CLI 「コマンドリファレンス[SendMessage](#)」の「」を参照してください。

send-users-messages

次の例は、send-users-messages を使用する方法を説明しています。

AWS CLI

アプリケーションのユーザーにSMSメッセージを送信するには

次のsend-users-messages例では、アプリケーションのユーザーに直接メッセージを送信します。

```
aws pinpoint send-users-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --send-users-message-request file://myfile.json \  
  --region us-west-2
```

myfile.json の内容:

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Users": {  
    "testuser": {}  
  }  
}
```

出力:

```
{  
  "SendUsersMessageResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "RequestId": "e0b12cf5-2359-11e9-bb0b-d5fb91876b25",  
    "Result": {  
      "testuser": {  
        "testuserendpoint": {  
          "DeliveryStatus": "SUCCESSFUL",  
          "MessageId": "7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",  
          "StatusCode": 200,  

```

```
        "StatusMessage": "MessageId:
7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",
        "Address": "+12345678900"
    }
}
}
```

詳細については、[Amazon Pinpoint ユーザーガイド](#)の「Amazon Pinpoint SMSチャンネル」を参照してください。Amazon Pinpoint

- API 詳細については、AWS CLI 「コマンドリファレンス[SendUsersMessages](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを追加するには

次の例では、リソースに 2 つのタグ (キー名と値) を追加します。

```
aws pinpoint list-tags-for-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tags-model tags={Stack=Production,Year=2019}
```

このコマンドでは何も出力されません。

詳細については、Amazon Pinpoint デベロッパーガイド」のAmazon Pinpoint リソース <<https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>>」の「タグ付け」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

例 1: リソースからタグを削除するには

次の`untag-resource`例では、リソースから指定されたタグ (キー名と値) を削除します。

```
aws pinpoint untag-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tag-keys Year
```

このコマンドでは何も出力されません。

例 2: リソースから複数のタグを削除するには

次の`untag-resource`例では、リソースから指定されたタグ (キー名と値) を削除します。

```
aws pinpoint untag-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tag-keys Year Stack
```

このコマンドでは何も出力されません。

詳細については、「Amazon Pinpoint デベロッパーガイド」の Amazon Pinpoint リソース <<https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>>」の「タグ付け」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-sms-channel

次のコード例は、`update-sms-channel` を使用する方法を示しています。

AWS CLI

SMSチャネルを有効にするか、SMSアプリケーションのステータスと設定を更新します。

次の`update-sms-channel`例では、アプリケーションのSMSチャネルのSMSチャネルを有効にします。

```
aws pinpoint update-sms-channel \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --sms-channel-request Enabled=true \  
  --region us-west-2
```

出力:

```
{  
  "SMSChannelResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "CreationDate": "2019-01-28T23:25:25.224Z",  
    "Enabled": true,  
    "Id": "sms",  
    "IsArchived": false,  
    "LastModifiedDate": "2023-05-18T23:22:50.977Z",  
    "Platform": "SMS",  
    "PromotionalMessagesPerSecond": 20,  
    "TransactionalMessagesPerSecond": 20,  
    "Version": 3  
  }  
}
```

詳細については、[Amazon Pinpoint ユーザーガイド](#)の「[Amazon Pinpoint SMSチャンネル](#)」を参照してください。Amazon Pinpoint

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSmsChannel](#)」の「」を参照してください。

を使用した Amazon Polly の例 AWS CLI

次のコード例は、Amazon Polly AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

• [アクション](#)

アクション

delete-lexicon

次のコード例は、delete-lexicon を使用する方法を示しています。

AWS CLI

レキシコンを削除するには

次の delete-lexicon の例は、指定されたレキシコンを削除します。

```
aws polly delete-lexicon \  
  --name w3c
```

このコマンドでは何も出力されません。

詳細については、Amazon Pollyデベロッパーガイド」の「[DeleteLexicon オペレーションの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLexicon](#)」の「」を参照してください。

get-lexicon

次の例は、get-lexicon を使用する方法を説明しています。

AWS CLI

レキシコンの内容を取得するには

次の get-lexicon の例は、指定された発音レキシコンの内容を取得します。

```
aws polly get-lexicon \  
  --name w3c
```

出力:


```
{
  "Lexicon": {
    "Content": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<lexicon version=
\"1.0\" \n      xmlns=      \"http://www.w3.org/2005/01/pronunciation-lexicon
\n      xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" \n
xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon \n
http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\"\n
  alphabet=\"ipa\" \n      xml:lang=\"en-US\">\n  <lexeme>\n    <grapheme>W3C</
grapheme>\n    <alias>World Wide Web Consortium</alias>\n  </lexeme>\n</lexicon>
\n",
    "Name": "w3c"
  },
  "LexiconAttributes": {
    "Alphabet": "ipa",
    "LanguageCode": "en-US",
    "LastModified": 1603908910.99,
    "LexiconArn": "arn:aws:polly:us-west-2:880185128111:lexicon/w3c",
    "LexemesCount": 1,
    "Size": 492
  }
}
```

詳細については、Amazon Pollyデベロッパーガイド」の「[GetLexicon オペレーションの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLexicon](#)」の「」を参照してください。

get-speech-synthesis-task

次の例は、get-speech-synthesis-task を使用する方法を説明しています。

AWS CLI

音声合成タスクに関する情報を取得するには

次の get-speech-synthesis-task の例は、指定された音声合成タスクに関する情報を取得します。

```
aws polly get-speech-synthesis-task \
  --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

出力:

```
{
  "SynthesisTask": {
    "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
    "TaskStatus": "completed",
    "OutputUri": "https://s3.us-west-2.amazonaws.com/my-s3-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
    "CreationTime": 1603911042.689,
    "RequestCharacters": 1311,
    "OutputFormat": "mp3",
    "TextType": "text",
    "VoiceId": "Joanna"
  }
}
```

詳細については、「Amazon Polly デイベロッパーガイド」の「[長いオーディオファイルの作成 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSpeechSynthesisTask](#)」の「」を参照してください。

list-lexicons

次の例は、list-lexicons を使用する方法を説明しています。

AWS CLI

レキシコンを一覧表示するには

次の list-lexicons の例は、発音レキシコンを一覧表示します。

```
aws polly list-lexicons
```

出力:

```
{
  "Lexicons": [
    {
      "Name": "w3c",
      "Attributes": {
```

```
    "Alphabet": "ipa",
    "LanguageCode": "en-US",
    "LastModified": 1603908910.99,
    "LexiconArn": "arn:aws:polly:us-east-2:123456789012:lexicon/w3c",
    "LexemesCount": 1,
    "Size": 492
  }
}
]
```

詳細については、Amazon Pollyデベロッパーガイド」の「[ListLexicons オペレーションの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLexicons](#)」の「」を参照してください。

list-speech-synthesis-tasks

次のコード例は、list-speech-synthesis-tasks を使用する方法を示しています。

AWS CLI

音声合成タスクを一覧表示するには

次のlist-speech-synthesis-tasks例では、音声合成タスクを一覧表示します。

```
aws polly list-speech-synthesis-tasks
```

出力:

```
{
  "SynthesisTasks": [
    {
      "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
      "TaskStatus": "completed",
      "OutputUri": "https://s3.us-west-2.amazonaws.com/my-s3-bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
      "CreationTime": 1603911042.689,
      "RequestCharacters": 1311,
      "OutputFormat": "mp3",
      "TextType": "text",
    }
  ]
}
```

```

        "VoiceId": "Joanna"
      }
    ]
  }

```

詳細については、「Amazon Polly デイベロッパーガイド」の「[長いオーディオファイルの作成 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSpeechSynthesisTasks](#)」の「」を参照してください。

put-lexicon

次のコード例は、put-lexicon を使用する方法を示しています。

AWS CLI

レキシコンを保存するには

次の put-lexicon の例は、指定された発音レキシコンを保存します。example.pls ファイルは、W3C PLS 準拠のレキシコンを指定します。

```

aws polly put-lexicon \
  --name w3c \
  --content file://example.pls

```

example.pls の内容

```

{
  <?xml version="1.0" encoding="UTF-8"?>
  <lexicon version="1.0"
    xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
      http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
    alphabet="ipa"
    xml:lang="en-US">
    <lexeme>
      <grapheme>W3C</grapheme>
      <alias>World Wide Web Consortium</alias>
    </lexeme>
  </lexicon>

```

```
}
```

このコマンドでは何も出力されません。

詳細については、「Amazon Pollyデベロッパーガイド」の「[PutLexicon オペレーションの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLexicon](#)」の「」を参照してください。

start-speech-synthesis-task

次のコード例は、start-speech-synthesis-task を使用する方法を示しています。

AWS CLI

テキストを合成するには

次のstart-speech-synthesis-task例では、テキストを `text_file.txt` に合成し、結果のMP3ファイルを指定されたバケットに保存します。

```
aws polly start-speech-synthesis-task \  
  --output-format mp3 \  
  --output-s3-bucket-name my-s3-bucket \  
  --text file://text_file.txt \  
  --voice-id Joanna
```

出力:

```
{  
  "SynthesisTask": {  
    "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",  
    "TaskStatus": "scheduled",  
    "OutputUri": "https://s3.us-east-2.amazonaws.com/my-s3-  
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",  
    "CreationTime": 1603911042.689,  
    "RequestCharacters": 1311,  
    "OutputFormat": "mp3",  
    "TextType": "text",  
    "VoiceId": "Joanna"  
  }  
}
```

```
}
```

詳細については、「Amazon Polly デイベロッパーガイド」の「[長いオーディオファイルの作成 \(CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartSpeechSynthesisTask](#)」の「」を参照してください。

AWS の料金表 を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS の料金表。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-services

次のコード例は、describe-services を使用する方法を示しています。

AWS CLI

サービスメタデータを取得するには

この例では、Amazon EC2サービスコードのメタデータを取得します。

コマンド:

```
aws pricing describe-services --service-code AmazonEC2 --format-version aws_v1 --  
max-items 1
```

出力:

```
{
  "Services": [
    {
      "ServiceCode": "AmazonEC2",
      "AttributeNames": [
        "volumeType",
        "maxIopsvolume",
        "instance",
        "instanceCapacity10xlarge",
        "locationType",
        "instanceFamily",
        "operatingSystem",
        "clockSpeed",
        "LeaseContractLength",
        "ecu",
        "networkPerformance",
        "instanceCapacity8xlarge",
        "group",
        "maxThroughputvolume",
        "gpuMemory",
        "ebsOptimized",
        "elasticGpuType",
        "maxVolumeSize",
        "gpu",
        "processorFeatures",
        "intelAvxAvailable",
        "instanceCapacity4xlarge",
        "servicecode",
        "groupDescription",
        "processorArchitecture",
        "physicalCores",
        "productFamily",
        "enhancedNetworkingSupported",
        "intelTurboAvailable",
        "memory",
        "dedicatedEbsThroughput",
        "vcpu",
        "OfferingClass",
        "instanceCapacityLarge",
        "capacitystatus",
        "termType",
        "storage",
```

```
        "intelAvx2Available",
        "storageMedia",
        "physicalProcessor",
        "provisioned",
        "servicename",
        "PurchaseOption",
        "instanceCapacity18xlarge",
        "instanceType",
        "tenancy",
        "usagetype",
        "normalizationSizeFactor",
        "instanceCapacity2xlarge",
        "instanceCapacity16xlarge",
        "maxIopsBurstPerformance",
        "instanceCapacity12xlarge",
        "instanceCapacity32xlarge",
        "instanceCapacityXlarge",
        "licenseModel",
        "currentGeneration",
        "preInstalledSw",
        "location",
        "instanceCapacity24xlarge",
        "instanceCapacity9xlarge",
        "instanceCapacityMedium",
        "operation"
    ]
}
],
"FormatVersion": "aws_v1"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeServices](#)」の「」を参照してください。

get-attribute-values

次のコード例は、get-attribute-values を使用する方法を示しています。

AWS CLI

属性値のリストを取得するには

次のget-attribute-values例では、特定の属性で使用できる値のリストを取得します。


```
aws pricing get-attribute-values \  
  --service-code AmazonEC2 \  
  --attribute-name volumeType \  
  --max-items 2
```

出力:

```
{  
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",  
  "AttributeValues": [  
    {  
      "Value": "Cold HDD"  
    },  
    {  
      "Value": "General Purpose"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAttributeValues](#)」の「」を参照してください。

get-products

次のコード例は、get-products を使用する方法を示しています。

AWS CLI

製品のリストを取得するには

この例では、指定された条件に一致する製品のリストを取得します。

コマンド:

```
aws pricing get-products --filters file:///filters.json --format-version aws_v1 --  
max-results 1 --service-code AmazonEC2
```

filters.json:

```
[
```

```
{
  "Type": "TERM_MATCH",
  "Field": "ServiceCode",
  "Value": "AmazonEC2"
},
{
  "Type": "TERM_MATCH",
  "Field": "volumeType",
  "Value": "Provisioned IOPS"
}
]
```

出力:

```
{
  "FormatVersion": "aws_v1",
  "NextToken": "WGDY7ko8fQXdlaUZVdasFQ==:RVSagyIFn770XQ0zdUIc09BY6ucBG9itXAZGZF/
zioUz0sUKh6PCcPwa0yPZRiMePb986TeoKYB9155fw/
CyoMq5ymnGmT1Vj39TljjbbAlhcqnVfTmPIilx8Uy5bdDaBYy/e/20fw9Edzsykbs8LTBUbniDQ
+BBds5yeI9AQkUepuKk3aEahFPxJ55kx/zk",
  "PriceList": [
    {"productFamily": "Storage", "attributes": {"storageMedia": "SSD-backed", "maxThroughputVolume": "320 MB/sec", "volumeType": "Provisioned IOPS", "maxIopsVolume": "20000", "serviceCode": "AmazonEC2", "usageType": "APS1-EBS:VolumeUsage.piops", "locationType": "AWS Region", "location": "Asia Pacific (Singapore)", "serviceName": "Amazon Elastic Compute Cloud", "maxVolumeSize": "16 TiB", "operation": ""}, "sku": "3MKHN58N7RDDVGKJ"}, {"serviceCode": "AmazonEC2", "terms": {"OnDemand": {"3MKHN58N7RDDVGKJ.JRTCKXETXF": {"priceDimensions": {"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7": {"unit": "GB-Mo", "endRange": "Inf", "description": "$0.138 per GB-month of Provisioned IOPS SSD (io1) provisioned storage - Asia Pacific (Singapore)", "appliesTo": [], "rateCode": "3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7", "beginRange": "0", "pricePerUnit": {"USD": "0.1380000000"}}}}, "sku": "3MKHN58N7RDDVGKJ", "effectiveDate": "2018-08-01T00:00:00Z", "offerTermCode": "JRTCKXETXF", "termAttributes": {}}}, {"version": "20180808005701", "publicationDate": "2018-08-08T00:57:01Z"}]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetProducts](#)」の「」を参照してください。

AWS Private CA を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Private CA。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-certificate-authority-audit-report

次のコード例は、create-certificate-authority-audit-report を使用する方法を示しています。

AWS CLI

認証機関監査レポートを作成するには

次のcreate-certificate-authority-audit-reportコマンドは、によって識別されるプライベート CA の監査レポートを作成しますARN。

```
aws acm-pca create-certificate-authority-audit-report --certificate-authority-arn arn:aws:acm-pca:us-east-1:accountid:certificate-authority/12345678-1234-1234-1234-123456789012 --s3-bucket-name your-bucket-name --audit-report-response-format JSON
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCertificateAuthorityAuditReport](#)」の「」を参照してください。

create-certificate-authority

次の例は、create-certificate-authority を使用する方法を説明しています。

AWS CLI

プライベート認証局を作成するには

次のcreate-certificate-authorityコマンドは、AWS アカウントにプライベート認証局を作成します。

```
aws acm-pca create-certificate-authority --certificate-authority-configuration
file://C:\ca_config.txt --revocation-configuration file://C:\revoke_config.txt --
certificate-authority-type "SUBORDINATE" --idempotency-token 98256344
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCertificateAuthority](#)」の「」を参照してください。

delete-certificate-authority

次の例は、delete-certificate-authority を使用する方法を説明しています。

AWS CLI

プライベート認証局を削除するには

次のdelete-certificate-authorityコマンドは、によって識別される認証局を削除します ARN。

```
aws acm-pca delete-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCertificateAuthority](#)」の「」を参照してください。

describe-certificate-authority-audit-report

次のコード例は、describe-certificate-authority-audit-report を使用する方法を示しています。

AWS CLI

認証機関の監査レポートを記述するには

次のdescribe-certificate-authority-audit-reportコマンドは、によって識別される CA の指定された監査レポートに関する情報を一覧表示しますARN。

```
aws acm-pca describe-certificate-authority-audit-report --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/99999999-8888-7777-6666-555555555555 --audit-report-  
id 11111111-2222-3333-4444-555555555555
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCertificateAuthorityAuditReport](#)」の「」を参照してください。

describe-certificate-authority

次の例は、describe-certificate-authority を使用する方法を説明しています。

AWS CLI

プライベート認証局を記述するには

次のdescribe-certificate-authorityコマンドは、によって識別されるプライベート CA に関する情報を一覧表示しますARN。

```
aws acm-pca describe-certificate-authority --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCertificateAuthority](#)」の「」を参照してください。

get-certificate-authority-certificate

次のコード例は、get-certificate-authority-certificate を使用する方法を示しています。

AWS CLI

認証局 (CA) 証明書を取得するには

次のget-certificate-authority-certificateコマンドは、によって指定されたプライベート CA の証明書と証明書チェーンを取得しますARN。

```
aws acm-pca get-certificate-authority-certificate --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012 --output text
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCertificateAuthorityCertificate](#)」の「」を参照してください。

get-certificate-authority-csr

次のコード例は、get-certificate-authority-csr を使用する方法を示しています。

AWS CLI

認証機関の証明書署名リクエストを取得するには

次のget-certificate-authority-csr コマンドは、で指定されたプライベート CA CSRの取得しますARN。

```
aws acm-pca get-certificate-authority-csr --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012 --output text
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCertificateAuthorityCsr](#)」の「」を参照してください。

get-certificate

次のコード例は、get-certificate を使用する方法を示しています。

AWS CLI

発行された証明書を取得するには

次のget-certificate例では、指定されたプライベート CA から証明書を取得します。

```
aws acm-pca get-certificate \  
  --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012 \  
  --certificate-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012/  
certificate/6707447683a9b7f4055627ffd55cebcc \  
  --output text
```

```
--output text
```

出力:

```
-----BEGIN CERTIFICATE-----
MIIEDzCCAvEgAwIBAgIRAJuJ8f6ZVYL7gG/rS3qvrZMwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMCVVMxEzARBgNVBAGMCl dhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
tKCSglgZZrd4FdLw1EkGm+UVXnodwMtJEQyy3oTfZjURPIyyaqsKtu/KSS7YDjK0
KQNY73D6Ltmd0EbAyq10XiDxqY411vKHJ1eZrPaBmYNABxU=
-----END CERTIFICATE----- -----BEGIN CERTIFICATE-----
MIIDrzCCApEgAwIBAgIRA0skdzLvcj1eShkoyEE693AwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMCVVMxEzARBgNVBAGMCl dhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
kdRGB6P2hpxstDOUIwAoCbhoaWwFA4ybJznf+j0QhAziNlRdKQRR8n0DwPkt7H9w
dJ5nxsTk/fniJz86Ddtp6n8s82wYdkN3cVffeK72A9aTCOU=
-----END CERTIFICATE-----
```

出力の最初の部分は証明書自体です。2 番目のパートは、ルート CA 証明書に連鎖する証明書チェーンです。--output text オプションを使用すると、2 つの証明書部分の間に TAB 文字が挿入されます (これがインデントされたテキストの原因です)。この出力を取得し、証明書を他のツールで解析する場合は、TAB 文字を削除して正しく処理する必要がある場合があります。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetCertificate](#)」の「」を参照してください。

import-certificate-authority-certificate

次のコード例は、import-certificate-authority-certificate を使用する方法を示しています。

AWS CLI

認証局証明書を にインポートするには ACM PCA

次のimport-certificate-authority-certificateコマンドは、によって指定された CA の署名付きプライベート CA 証明書を ACM ARN にインポートしますPCA。

```
aws acm-pca import-certificate-authority-certificate --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
```

```
authority/12345678-1234-1234-1234-123456789012 --certificate file://C:\ca_cert.pem  
--certificate-chain file://C:\ca_cert_chain.pem
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportCertificateAuthorityCertificate](#)」の「」を参照してください。

issue-certificate

次の例は、issue-certificate を使用する方法を説明しています。

AWS CLI

プライベート証明書を発行するには

次のissue-certificateコマンドは、で指定されたプライベート CA ARN を使用してプライベート証明書を発行します。

```
aws acm-pca issue-certificate --certificate-authority-arn arn:aws:acm-pca:us-  
west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012  
--csr file://C:\cert_1.csr --signing-algorithm "SHA256WITHRSA" --validity  
Value=365,Type="DAYS" --idempotency-token 1234
```

- API 詳細については、AWS CLI 「コマンドリファレンス[IssueCertificate](#)」の「」を参照してください。

list-certificate-authorities

次の例は、list-certificate-authorities を使用する方法を説明しています。

AWS CLI

プライベート認証機関を一覧表示するには

次のlist-certificate-authoritiesコマンドは、アカウントCAs内のすべてのプライベートに関する情報を表示します。

```
aws acm-pca list-certificate-authorities --max-results 10
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCertificateAuthorities](#)」の「」を参照してください。

list-tags

次の例は、list-tags を使用する方法を説明しています。

AWS CLI

認証機関のタグを一覧表示するには

次のlist-tagsコマンドは、 で指定されたプライベート CA に関連付けられたタグを一覧表示しますARN。

```
aws acm-pca list-tags --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012 --max-results 10
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTags](#)」の「」を参照してください。

revoke-certificate

次の例は、revoke-certificate を使用する方法を説明しています。

AWS CLI

プライベート証明書を取り消すには

次のrevoke-certificateコマンドは、 によって識別された CA からプライベート証明書を取り消しますARN。

```
aws acm-pca revoke-certificate --certificate-authority-arn arn:aws:acm-pca:us-west-2:1234567890:certificate-authority/12345678-1234-1234-1234-123456789012 --certificate-serial 67:07:44:76:83:a9:b7:f4:05:56:27:ff:d5:5c:eb:cc --revocation-reason "KEY_COMPROMISE"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RevokeCertificate](#)」の「」を参照してください。

tag-certificate-authority

次の例は、tag-certificate-authority を使用する方法を説明しています。

AWS CLI

プライベート認証局にタグをアタッチするには

次のtag-certificate-authorityコマンドは、プライベート CA に 1 つ以上のタグをアタッチします。

```
aws acm-pca tag-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012 --tags Key=Admin,Value=Alice
```

- API 詳細については、AWS CLI 「コマンドリファレンス[TagCertificateAuthority](#)」の「」を参照してください。

untag-certificate-authority

次の例は、untag-certificate-authority を使用する方法を説明しています。

AWS CLI

プライベート認証局から 1 つ以上のタグを削除するには

次のuntag-certificate-authorityコマンドは、によって識別されるプライベート CA からタグを削除しますARN。

```
aws acm-pca untag-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012 --tags Key=Purpose,Value=Website
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagCertificateAuthority](#)」の「」を参照してください。

update-certificate-authority

次の例は、update-certificate-authority を使用する方法を説明しています。

AWS CLI

プライベート認証局の設定を更新するには

次のupdate-certificate-authorityコマンドは、 によって識別されるプライベート CA のステータスと設定を更新しますARN。

```
aws acm-pca update-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-1232456789012 --revocation-configuration file://C:\revoke_config.txt --status "DISABLED"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateCertificateAuthority](#)」の「」を参照してください。

AWS Proton を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Proton。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

cancel-service-instance-deployment

次の例は、cancel-service-instance-deployment を使用する方法を説明しています。

AWS CLI

サービスインスタンスのデプロイをキャンセルするには

次のcancel-service-instance-deployment例では、サービスインスタンスのデプロイをキャンセルします。

```
aws proton cancel-service-instance-deployment \
```

```
--service-instance-name "instance-one" \  
--service-name "simple-svc"
```

出力:

```
{  
  "serviceInstance": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-  
instance/instance-one",  
    "createdAt": "2021-04-02T21:29:59.962000+00:00",  
    "deploymentStatus": "CANCELLING",  
    "environmentName": "simple-env",  
    "lastDeploymentAttemptedAt": "2021-04-02T21:45:15.406000+00:00",  
    "lastDeploymentSucceededAt": "2021-04-02T21:38:00.823000+00:00",  
    "name": "instance-one",  
    "serviceName": "simple-svc",  
    "spec": "proton: ServiceSpec\npipeline:\n  
my_sample_pipeline_optional_input: abc\n my_sample_pipeline_required_input:  
'123'\ninstances:\n- name: my-instance\n environment: MySimpleEnv  
\n spec:\n  my_sample_service_instance_optional_input: def\n my_sample_service_instance_required_input: '456'\n- name: my-other-instance\n environment: MySimpleEnv\n spec:\n  my_sample_service_instance_required_input:  
'789'\n",  
    "templateMajorVersion": "1",  
    "templateMinorVersion": "1",  
    "templateName": "svc-simple"  
  }  
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスインスタンスの更新」](#)またはAWS「Proton ユーザーガイド」の[「サービスインスタンスの更新」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CancelServiceInstanceDeployment](#)」の「」を参照してください。

cancel-service-pipeline-deployment

次の例は、cancel-service-pipeline-deployment を使用する方法を説明しています。

AWS CLI

サービスパイプラインのデプロイをキャンセルするには

次のcancel-service-pipeline-deployment例では、サービスパイプラインのデプロイをキャンセルします。

```
aws proton cancel-service-pipeline-deployment \  
  --service-name "simple-svc"
```

出力:

```
{  
  "pipeline": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline",  
    "createdAt": "2021-04-02T21:29:59.962000+00:00",  
    "deploymentStatus": "CANCELLING",  
    "lastDeploymentAttemptedAt": "2021-04-02T22:02:45.095000+00:00",  
    "lastDeploymentSucceededAt": "2021-04-02T21:39:28.991000+00:00",  
    "templateMajorVersion": "1",  
    "templateMinorVersion": "1",  
    "templateName": "svc-simple"  
  }  
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスパイプラインの更新」](#)またはAWS「Proton ユーザーガイド」の[「サービスパイプラインの更新」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CancelServicePipelineDeployment](#)」の「」を参照してください。

create-service

次のコード例は、create-service を使用する方法を示しています。

AWS CLI

サービスを作成するには

次のcreate-service例では、サービスパイプラインを使用してサービスを作成します。

```
aws proton create-service \  
  --name "MySimpleService" \  
  --template-name "fargate-service" \  
  --template-major-version "1" \  
  --branch-name "mainline" \  
  --
```

```
--repository-connection-arn "arn:aws:codestar-connections:region-id:account-
id:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \
--repository-id "myorg/myapp" \
--spec file://spec.yaml
```

spec.yaml の内容:

```
proton: ServiceSpec

pipeline:
  my_sample_pipeline_required_input: "hello"
  my_sample_pipeline_optional_input: "bye"

instances:
  - name: "acme-network-dev"
    environment: "ENV_NAME"
    spec:
      my_sample_service_instance_required_input: "hi"
      my_sample_service_instance_optional_input: "ho"
```

出力:

```
{
  "service": {
    "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",
    "createdAt": "2020-11-18T19:50:27.460000+00:00",
    "lastModifiedAt": "2020-11-18T19:50:27.460000+00:00",
    "name": "MySimpleService",
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "repositoryId": "myorg/myapp",
    "status": "CREATE_IN_PROGRESS",
    "templateName": "fargate-service"
  }
}
```

詳細については、「Proton <https://docs.aws.amazon.com/proton/latest/adminguide/ag-create-svc.html> 管理者ガイド」の「サービスの作成」および「Proton ユーザーガイド」の「サービスの作成」を参照してください。 AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateService](#)」の「」を参照してください。

delete-service

次のコード例は、delete-service を使用する方法を示しています。

AWS CLI

サービスを削除するには

次のdelete-service例では、サービスを削除します。

```
aws proton delete-service \  
  --name "simple-svc"
```

出力:

```
{  
  "service": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",  
    "branchName": "mainline",  
    "createdAt": "2020-11-28T22:40:50.512000+00:00",  
    "description": "Edit by updating description",  
    "lastModifiedAt": "2020-11-29T00:30:39.248000+00:00",  
    "name": "simple-svc",  
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-  
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "repositoryId": "myorg/myapp",  
    "status": "DELETE_IN_PROGRESS",  
    "templateName": "fargate-service"  
  }  
}
```

詳細については、AWS 「Proton 管理者ガイド」の「[サービスの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteService](#)」の「」を参照してください。

get-service-instance

次の例は、get-service-instance を使用する方法を説明しています。

AWS CLI

サービスインスタンスの詳細を取得するには

次のget-service-instance例では、サービスインスタンスの詳細データを取得します。

```
aws proton get-service-instance \  
  --name "instance-one" \  
  --service-name "simple-svc"
```

出力:

```
{  
  "serviceInstance": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-  
instance/instance-one",  
    "createdAt": "2020-11-28T22:40:50.512000+00:00",  
    "deploymentStatus": "SUCCEEDED",  
    "environmentName": "simple-env",  
    "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",  
    "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",  
    "name": "instance-one",  
    "serviceName": "simple-svc",  
    "spec": "proton: ServiceSpec\npipeline:\nmy_sample_pipeline_optional_input: hello world\nmy_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\nenvironment: my-simple-env\nspec:\n  my_sample_service_instance_optional_input:  
0la\n  my_sample_service_instance_required_input: Ciao\n",  
    "templateMajorVersion": "1",  
    "templateMinorVersion": "0",  
    "templateName": "svc-simple"  
  }  
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスデータの表示」](#)または「Proton ユーザーガイド」の[「サービスデータの表示」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[GetServiceInstance](#)」の「」を参照してください。

get-service

次のコード例は、get-service を使用する方法を示しています。

AWS CLI

サービスの詳細を取得するには

次のget-service例では、サービスの詳細データを取得します。

```
aws proton get-service \  
  --name "simple-svc"
```

出力:

```
{  
  "service": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc",  
    "branchName": "mainline",  
    "createdAt": "2020-11-28T22:40:50.512000+00:00",  
    "lastModifiedAt": "2020-11-28T22:44:51.207000+00:00",  
    "name": "simple-svc",  
    "pipeline": {  
      "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/  
pipeline/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "createdAt": "2020-11-28T22:40:50.512000+00:00",  
      "deploymentStatus": "SUCCEEDED",  
      "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",  
      "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",  
      "spec": "proton: ServiceSpec\npipeline:\n  
my_sample_pipeline_required_input: hello\n my_sample_pipeline_optional_input:  
bye\ninstances:\n- name: instance-svc-simple\n environment: my-simple-  
env\n spec:\n   my_sample_service_instance_required_input: hi\n   my_sample_service_instance_optional_input: ho\n",  
      "templateMajorVersion": "1",  
      "templateMinorVersion": "1",  
      "templateName": "svc-simple"  
    },  
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-  
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "repositoryId": "myorg/myapp",  
    "spec": "proton: ServiceSpec\npipeline:\n  
my_sample_pipeline_required_input: hello\n my_sample_pipeline_optional_input:  
bye\ninstances:\n- name: instance-svc-simple\n environment: my-simple-  
env\n spec:\n   my_sample_service_instance_required_input: hi\n   my_sample_service_instance_optional_input: ho\n",  
    "status": "ACTIVE",  
  },  
}
```

```
    "templateName": "svc-simple"
  }
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスデータの表示」](#)または「Proton ユーザーガイド」の[「サービスデータの表示」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[GetService](#)」の「」を参照してください。

list-service-instances

次のコード例は、list-service-instances を使用する方法を示しています。

AWS CLI

例 1: すべてのサービスインスタンスを一覧表示するには

次のlist-service-instances例では、サービスインスタンスを一覧表示します。

```
aws proton list-service-instances
```

出力:

```
{
  "serviceInstances": [
    {
      "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/
service-instance/instance-one",
      "createdAt": "2020-11-28T22:40:50.512000+00:00",
      "deploymentStatus": "SUCCEEDED",
      "environmentArn": "arn:aws:proton:region-id:123456789012:environment/
simple-env",
      "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",
      "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",
      "name": "instance-one",
      "serviceName": "simple-svc",
      "templateMajorVersion": "1",
      "templateMinorVersion": "0",
      "templateName": "fargate-service"
    }
  ]
}
```

```
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスインスタンスデータの表示」](#)または[「Proton ユーザーガイド」の「サービスインスタンスデータの表示」](#)を参照してください。

AWS

例 2: 指定されたサービスインスタンスを一覧表示するには

次のget-service-instance例では、サービスインスタンスを取得します。

```
aws proton get-service-instance \  
  --name "instance-one" \  
  --service-name "simple-svc"
```

出力:

```
{  
  "serviceInstance": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-  
instance/instance-one",  
    "createdAt": "2020-11-28T22:40:50.512000+00:00",  
    "deploymentStatus": "SUCCEEDED",  
    "environmentName": "simple-env",  
    "lastDeploymentAttemptedAt": "2020-11-28T22:40:50.512000+00:00",  
    "lastDeploymentSucceededAt": "2020-11-28T22:40:50.512000+00:00",  
    "name": "instance-one",  
    "serviceName": "simple-svc",  
    "spec": "proton: ServiceSpec\npipeline:\nmy_sample_pipeline_optional_input: hello world\nmy_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\nenvironment: my-simple-env\nspec:\n  my_sample_service_instance_optional_input:  
0la\n  my_sample_service_instance_required_input: Ciao",  
    "templateMajorVersion": "1",  
    "templateMinorVersion": "0",  
    "templateName": "svc-simple"  
  }  
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスインスタンスデータの表示」](#)または[「Proton ユーザーガイド」の「サービスインスタンスデータの表示」](#)を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServiceInstances](#)」の「」を参照してください。

update-service-instance

次の例は、update-service-instance を使用する方法を説明しています。

AWS CLI

サービスインスタンスを新しいマイナーバージョンに更新するには

次のupdate-service-instance例では、サービスインスタンスをサービステンプレートの新しいマイナーバージョンに更新し、my-other-instance「」という名前の新しいインスタンスを新しい必須入力に追加します。

```
aws proton update-service-instance \  
  --service-name "simple-svc" \  
  --spec "file://service-spec.yaml" \  
  --template-major-version "1" \  
  --template-minor-version "1" \  
  --deployment-type "MINOR_VERSION" \  
  --name "instance-one"
```

service-spec.yaml の内容:

```
proton: ServiceSpec  
pipeline:  
  my_sample_pipeline_optional_input: "abc"  
  my_sample_pipeline_required_input: "123"  
instances:  
  - name: "instance-one"  
    environment: "simple-env"  
    spec:  
      my_sample_service_instance_optional_input: "def"  
      my_sample_service_instance_required_input: "456"  
  - name: "my-other-instance"  
    environment: "simple-env"  
    spec:  
      my_sample_service_instance_required_input: "789"
```

出力:

```
{
  "serviceInstance": {
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/service-instance/instance-one",
    "createdAt": "2021-04-02T21:29:59.962000+00:00",
    "deploymentStatus": "IN_PROGRESS",
    "environmentName": "arn:aws:proton:region-id:123456789012:environment/simple-env",
    "lastDeploymentAttemptedAt": "2021-04-02T21:38:00.823000+00:00",
    "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",
    "name": "instance-one",
    "serviceName": "simple-svc",
    "templateMajorVersion": "1",
    "templateMinorVersion": "0",
    "templateName": "svc-simple"
  }
}
```

詳細については、AWS 「Proton 管理者ガイド」の[「サービスインスタンスの更新」](#)またはAWS 「Proton ユーザーガイド」の[「サービスインスタンスの更新」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateServiceInstance](#)」の「」を参照してください。

update-service-pipeline

次のコード例は、update-service-pipeline を使用する方法を示しています。

AWS CLI

サービスパイプラインを更新するには

次のupdate-service-pipeline例では、サービスパイプラインをサービステンプレートの新しいマイナーバージョンに更新します。

```
aws proton update-service-pipeline \
  --service-name "simple-svc" \
  --spec "file://service-spec.yaml" \
  --template-major-version "1" \
  --template-minor-version "1" \
  --deployment-type "MINOR_VERSION"
```

出力:

```
{
  "pipeline": {
    "arn": "arn:aws:proton:region-id:123456789012:service/simple-svc/pipeline/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "createdAt": "2021-04-02T21:29:59.962000+00:00",
    "deploymentStatus": "IN_PROGRESS",
    "lastDeploymentAttemptedAt": "2021-04-02T21:39:28.991000+00:00",
    "lastDeploymentSucceededAt": "2021-04-02T21:29:59.962000+00:00",
    "spec": "proton: ServiceSpec\n\npipeline:\n
my_sample_pipeline_optional_input: \"abc\"\n my_sample_pipeline_required_input:
\"123\"\n\ninstances:\n - name: \"my-instance\"\n environment: \"MySimpleEnv
\"\n spec:\n my_sample_service_instance_optional_input: \"def
\"\n my_sample_service_instance_required_input: \"456\"\n - name:
\"my-other-instance\"\n environment: \"MySimpleEnv\"\n spec:\n
my_sample_service_instance_required_input: \"789\"\n",
    "templateMajorVersion": "1",
    "templateMinorVersion": "0",
    "templateName": "svc-simple"
  }
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスパイプラインの更新」](#)またはAWS「Proton ユーザーガイド」の[「サービスパイプラインの更新」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateServicePipeline](#)」の「」を参照してください。

update-service

次の例は、update-service を使用する方法を説明しています。

AWS CLI

サービスを更新するには

次のupdate-service例では、サービスの説明を編集します。

```
aws proton update-service \
  --name "MySimpleService" \
  --description "Edit by updating description"
```

出力:

```
{
  "service": {
    "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",
    "branchName": "mainline",
    "createdAt": "2021-03-12T22:39:42.318000+00:00",
    "description": "Edit by updating description",
    "lastModifiedAt": "2021-03-12T22:44:21.975000+00:00",
    "name": "MySimpleService",
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-
id:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "repositoryId": "myorg/myapp",
    "status": "ACTIVE",
    "templateName": "fargate-service"
  }
}
```

詳細については、AWS「Proton 管理者ガイド」の[「サービスの編集」](#)または AWS「Proton ユーザーガイド」の[「サービスの編集」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateService](#)」の「」を参照してください。

QLDB を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますQLDB。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

cancel-journal-kinesis-stream

次のコード例は、cancel-journal-kinesis-stream を使用する方法を示しています。

AWS CLI

ジャーナルストリームをキャンセルするには

次のcancel-journal-kinesis-stream例では、台帳から指定されたジャーナルストリームをキャンセルします。

```
aws qldb cancel-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --stream-id 7ISckqwe4y25YyHLzYUFAf
```

出力:

```
{  
  "StreamId": "7ISckqwe4y25YyHLzYUFAf"  
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon からのジャーナルデータのストリーミングQLDB」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelJournalKinesisStream](#)」の「」を参照してください。

create-ledger

次のコード例は、create-ledger を使用する方法を示しています。

AWS CLI

例 1: デフォルトプロパティを使用して台帳を作成するには

次の create-ledger の例は、名前 myExampleLedger とアクセス許可モード STANDARD で台帳を作成します。削除保護と AWS KMSキーのオプションパラメータは指定されないため、それぞれ trueと AWS 所有KMSキーにデフォルト設定されます。


```
aws qlldb create-ledger \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

出力:

```
{  
  "State": "CREATING",  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "DeletionProtection": true,  
  "CreationDateTime": 1568839243.951,  
  "Name": "myExampleLedger",  
  "PermissionsMode": "STANDARD"  
}
```

例 2: 削除保護が無効になっている台帳、カスタマーマネージドKMSキー、および指定されたタグを作成するには

次の create-ledger の例は、名前 myExampleLedger2 とアクセス許可モード STANDARD で台帳を作成します。削除保護機能は無効になっており、指定されたカスタマーマネージドKMSキーが保管時の暗号化に使用され、指定されたタグがリソースにアタッチされます。

```
aws qlldb create-ledger \  
  --name myExampleLedger2 \  
  --permissions-mode STANDARD \  
  --no-deletion-protection \  
  --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tags IsTest=true,Domain=Test
```

出力:

```
{  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger2",  
  "DeletionProtection": false,  
  "CreationDateTime": 1568839543.557,  
  "State": "CREATING",  
  "Name": "myExampleLedger2",  
  "PermissionsMode": "STANDARD",  
  "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

```
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDB Ledgers の基本オペレーション」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLedger](#)」の「」を参照してください。

delete-ledger

次の例は、delete-ledger を使用する方法を説明しています。

AWS CLI

台帳を削除するには

次のdelete-ledger例では、指定された台帳を削除します。

```
aws qldb delete-ledger \  
  --name myExampleLedger
```

このコマンドでは何も出力されません。

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDB Ledgers の基本オペレーション」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLedger](#)」の「」を参照してください。

describe-journal-kinesis-stream

次の例は、describe-journal-kinesis-stream を使用する方法を説明しています。

AWS CLI

ジャーナルストリームを記述するには

次のdescribe-journal-kinesis-stream例では、台帳から指定されたジャーナルストリームの詳細を表示します。

```
aws qldb describe-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --journal-stream-name myExampleStream
```

```
--stream-id 7ISCKqwe4y25YyHLzYUFAf
```

出力:

```
{
  "Stream": {
    "LedgerName": "myExampleLedger",
    "CreationTime": 1591221984.677,
    "InclusiveStartTime": 1590710400.0,
    "ExclusiveEndTime": 1590796799.0,
    "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",
    "StreamId": "7ISCKqwe4y25YyHLzYUFAf",
    "Arn": "arn:aws:qldb:us-east-1:123456789012:stream/
myExampleLedger/7ISCKqwe4y25YyHLzYUFAf",
    "Status": "ACTIVE",
    "KinesisConfiguration": {
      "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-
qldb",
      "AggregationEnabled": true
    },
    "StreamName": "myExampleLedger-stream"
  }
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon からのジャーナルデータのストリーミングQLDB」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeJournalKinesisStream](#)」の「」を参照してください。

describe-journal-s3-export

次のコード例は、describe-journal-s3-export を使用する方法を示しています。

AWS CLI

ジャーナルエクスポートジョブを記述するには

次のdescribe-journal-s3-export例では、台帳から指定されたエクスポートジョブの詳細を表示します。

```
aws qldb describe-journal-s3-export \
```

```
--name myExampleLedger \  
--export-id ADR20NPKN5LINYGb4dp7yZ
```

出力:

```
{  
  "ExportDescription": {  
    "S3ExportConfiguration": {  
      "Bucket": "awsExampleBucket",  
      "Prefix": "ledgerexport1/",  
      "EncryptionConfiguration": {  
        "ObjectEncryptionType": "SSE_S3"  
      }  
    },  
    "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",  
    "Status": "COMPLETED",  
    "ExportCreationTime": 1568847801.418,  
    "InclusiveStartTime": 1568764800.0,  
    "ExclusiveEndTime": 1568847599.0,  
    "LedgerName": "myExampleLedger",  
    "ExportId": "ADR20NPKN5LINYGb4dp7yZ"  
  }  
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのジャーナルのエクスポート QLDB」](#)を参照してください。 QLDB

- API 詳細については、コマンドリファレンスの[DescribeJournalS3Export](#)を参照してください。 AWS CLI

describe-ledger

次の例は、describe-ledger を使用する方法を説明しています。

AWS CLI

台帳を記述するには

次のdescribe-ledger例では、指定された台帳の詳細を表示します。

```
aws qlldb describe-ledger \  
  --name myExampleLedger
```

出力:

```
{
  "CreationDateTime": 1568839243.951,
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "State": "ACTIVE",
  "Name": "myExampleLedger",
  "DeletionProtection": true,
  "PermissionsMode": "STANDARD",
  "EncryptionDescription": {
    "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
    "EncryptionStatus": "ENABLED"
  }
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDB Ledgers の基本オペレーション」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLedger](#)」の「」を参照してください。

export-journal-to-s3

次のコード例は、`export-journal-to-s3` を使用する方法を示しています。

AWS CLI

ジャーナルブロックを S3 にエクスポートするには

次の`export-journal-to-s3`例では、`myExampleLedger` という名前の台帳から、指定された日時範囲内のジャーナルブロックのエクスポートジョブを作成します。エクスポートジョブは、ブロックを指定された Amazon S3 バケットに書き込みます。

```
aws qldb export-journal-to-s3 \
  --name myExampleLedger \
  --inclusive-start-time 2019-09-18T00:00:00Z \
  --exclusive-end-time 2019-09-18T22:59:59Z \
  --role-arn arn:aws:iam::123456789012:role/my-s3-export-role \
  --s3-export-configuration file://my-s3-export-config.json
```

`my-s3-export-config.json` の内容:

```
{
  "Bucket": "awsExampleBucket",
  "Prefix": "ledgerexport1/",
  "EncryptionConfiguration": {
    "ObjectEncryptionType": "SSE_S3"
  }
}
```

出力:

```
{
  "ExportId": "ADR20NPKN5LINYGb4dp7yZ"
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのジャーナルのエクスポート QLDB」](#)を参照してください。 QLDB

- API 詳細については、「コマンドリファレンス」の[ExportJournalToS3](#)を参照してください。 AWS CLI

get-block

次のコード例は、get-block を使用する方法を示しています。

AWS CLI

例 1: 入力ファイルを使用してジャーナルブロックと検証の証明を取得するには

次のget-block例では、ブロックデータオブジェクトと、指定された台帳からの証明をリクエストします。リクエストは、指定されたダイジェストチップアドレスとブロックアドレス用です。

```
aws qldb get-block \
  --name vehicle-registration \
  --block-address file://myblockaddress.json \
  --digest-tip-address file://mydigesttipaddress.json
```

myblockaddress.json の内容:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iwl\\",sequenceNo:100}"
}
```

mydigesttipaddress.json の内容:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"
}
```

出力:

```
{
  "Block": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},transactionId:\\"FnQeJBAicTX0Ah32ZnVtSX\\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:{{NoChM92yKRuJAB/jeLd1VnYn4DHiWI071ACfic9uHc=}},entriesHash:{{105L0siKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:{{7kewBXhpdBc1cZKxhVmpoMHPUG0JtWQD0iY2LPfZkYA=}},entriesHashList:[{{eRSwnAM7WWANWDd5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},{{y5cCB7p0AIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX+KR+W/EJl4qD1mmV77KZQg=}}],transactionInfo:{statements:[{statement:\\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\\nINSERT INTO r.Owners.SecondaryOwners\\n  VALUE { 'PersonId' : 'CMVdR77XP8zAgLmmFDGTvt' }\\n}],startTime:2019-09-16T19:37:05.302Z,statementDigest:{{jcgPX2vs0J0waum4qmDYtn1pCAT9xKNIZa+2k4R+mxA=}}}],documents:[JUJgkIcNbhS2goq8RqLuZ4:{tableName:\\"VehicleRegistration\\",tableId:\\"BFJKdXgzT9oF4wjMbuXy4G\\",statements:[0]}],revisions:[{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},data:{VIN:\\"1N4AL11D75C109151\\",LicensePlateNumber:\\"LEWISR261LL\\",State:\\"WA\\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:{PrimaryOwner:{PersonId:\\"BFJKdXhnLRT27sXBnojNGW\\"},SecondaryOwners:[{PersonId:\\"CMVdR77XP8zAgLmmFDGTvt\\"}]},City:\\"Everett\\",metadata:{id:\\"JUJgkIcNbhS2goq8RqLuZ4\\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:\\"FnQeJBAicTX0Ah32ZnVtSX\\"}}}],
    },
    "Proof": {
      "IonText": "[{{13+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJkF3Do=}},{{9UnwnKSQT0i3ge1JMva+tMIqCEDaOPTkwxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFctLufgPM6qXHyTNECb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkN0xoI=}},{{+3CwPYG/ytf/vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]"
    }
  }
}
```

}

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのデータ検証QLDBQLDB」](#)を参照してください。

例 2: 省略構文を使用してジャーナルブロックと検証の証明を取得するには

次のget-block例では、短縮構文を使用して、ブロックデータオブジェクトと指定された台帳からの証明をリクエストします。リクエストは、指定されたダイジェストチップアドレスとブロックアドレス用です。

```
aws qldb get-block \
  --name vehicle-registration \
  --block-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100}"'
  \
  --digest-tip-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"'
```

出力:

```
{
  "Block": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},transactionId:\\"FnQeJBAicTX0Ah32ZnVtSX\\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:{{NoChM92yKRuJAB/jeLd1VnYn4DHiWIIf071ACfic9uHc=}},entriesHash:{{105L0siKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:{{7kewBXhpdBc1cZKxhVmpoMHPUGOJtWQD0iY2LPfZkYA=}},entriesHashList:{{eRSwnmAM7WWANWDd5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},{{y5cCB7p0AIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX+KR+W/EJl4qD1mmV77KZQg=}}}],transactionInfo:{statements:[{statement:\\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\\nINSERT INTO r.Owners.SecondaryOwners\\n  VALUE { 'PersonId' : 'CMVdR77XP8zAg1mmFDGTvt' }\\n},startTime:2019-09-16T19:37:05.302Z,statementDigest:{{jcgPX2vs0J0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:[JUJgkIcNbhS2goq8RqLuZ4:{tableName:\\"VehicleRegistration\\",tableId:\\"BFJKdXgzT9oF4wjMbuXy4G\\",statements:[0]}]},revisions:[{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},data:{VIN:\\"1N4AL11D75C109151\\",LicensePlateNumber:\\"LEWISR261LL\\",State:\\"WA\\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:{PrimaryOwner:{PersonId:\\"BFJKdXhnLRT27sXBnojNGW\\"},SecondaryOwners:
```



```
[{"PersonId":"CMVdR77XP8zAg1mmFDGTvt\"}],City\":\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}]}"
  },
  "Proof": {
    "IonText": "[{{13+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMva
+tMIqCEDa0PTkwxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFctLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhbBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkn0xoI=}},{{+3CwpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT
+qE=}}]"
  }
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのデータ検証QLDBQLDB」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBlock](#)」の「」を参照してください。

get-digest

次のコード例は、get-digest を使用する方法を示しています。

AWS CLI

台帳のダイジェストを取得するには

次のget-digest例では、ジャーナル内の最新のコミットされたブロックで、指定された台帳からダイジェストをリクエストします。

```
aws qlldb get-digest \
  --name vehicle-registration
```

出力:

```
{
  "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",
  "DigestTipAddress": {
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iw1\",sequenceNo:123}"
  }
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのデータ検証QLDBQLDB」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDigest](#)」の「」を参照してください。

get-revision

次の例は、get-revision を使用する方法を説明しています。

AWS CLI

例 1: 入力ファイルを使用してドキュメントのリビジョンと検証の証明を取得するには

次のget-revision例では、指定された台帳からリビジョンデータオブジェクトと証明をリクエストします。リクエストは、リビジョンの指定されたダイジェストチップアドレス、ドキュメント ID、およびブロックアドレスに対するものです。

```
aws qlldb get-revision \  
  --name vehicle-registration \  
  --block-address file://myblockaddress.json \  
  --document-id JUJgkIcNbhS2goq8RqLuZ4 \  
  --digest-tip-address file://mydigesttipaddress.json
```

myblockaddress.json の内容:

```
{  
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100}"  
}
```

mydigesttipaddress.json の内容:

```
{  
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"  
}
```

出力:

```
{  
  "Revision": {  
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},data:
```

```
{VIN:"1N4AL11D75C109151",LicensePlateNumber:"LEWISR261LL",State:"WA",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:{PrimaryOwner:{PersonId:"BFJKdXhnLRT27sXBnojNGW"},SecondaryOwners:[{PersonId:"CMVdR77XP8zAglmmFDGTvt"}]},City:"Everett"},metadata:{id:"JUJgkIcNbhS2goq8RqLuZ4",version:3,txTime:2019-09-16T19:37:05.344Z,txId:"FnQeJBAicTX0Ah32ZnVtSX"}},
  "Proof": {
    "IonText": "[{{eRSwnmAM7WWANWDD5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf+yJZVGlmsM6gr2T52QvB08Lg+KgpjcnWAU=}},{{7kewBXhpdBc1cZKxhVmpoMHPUG0JtwQD0iY2LPfZkYA=}},{{13+EXs69K1+reh1qyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso00RpWAA08010RJkf3Do=}},{{9UnwnKSQT0i3ge1JMVA+tMIqCEDaOPTkwxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkN0xoI=}},{{+3CwPYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnq0y3jfy=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]"
  }
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon での Data VerificationQLDBQLDB」](#)を参照してください。

例 2: ショートハンド構文を使用してドキュメントのリビジョンと検証の証明を取得するには

次のget-revision例では、短縮構文を使用して、指定された台帳からリビジョンデータオブジェクトと証明をリクエストします。リクエストは、リビジョンの指定されたダイジェストチップアドレス、ドキュメント ID、およびブロックアドレスに対するものです。

```
aws qldb get-revision \
  --name vehicle-registration \
  --block-address 'IonText="{strandId:"KmA3ZZca7vAIiJAK9S5Iw1",sequenceNo:100}"' \
  --document-id JUJgkIcNbhS2goq8RqLuZ4 \
  --digest-tip-address 'IonText="{strandId:"KmA3ZZca7vAIiJAK9S5Iw1",sequenceNo:123}"'
```

出力:

```
{
  "Revision": {
    "IonText": "{blockAddress:{strandId:"KmA3ZZca7vAIiJAK9S5Iw1",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKjK2FBa9faquUVNtg=}},data:{VIN:"1N4AL11D75C109151",LicensePlateNumber:"LEWISR261LL",State:"WA
```

```

\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\"BFJKdXhnLRT27sXBnojNGW\"},SecondaryOwners:
[[{PersonId:\"CMVdR77XP8zAg1mmFDGTvt\"}]],City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}
  },
  \"Proof\": {
    \"IonText\": \"[[{eRSwnmAM7WWANWDD5iG0yK+T4tDXyzUq6HZ/0fgLHos=}],{{VV1rdaNuf
+yJZVG1msM6gr2T52QvB08Lg+KgpjcnWAU=}},
{{7kewBXhpdBc1cZKxhVmpoMhpUGOJtWQD0iY2LPfZkYA=}},{{13+EXs69K1+rehlqyWLkt
+oHD1w4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNECb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhz1nGT/
ZQPadSmgfdtMYcwkN0xoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnq0y3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]\"
  }
}

```

詳細については、[「Amazon デベロッパーガイド」の「Amazon での Data VerificationQLDBQLDB」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRevision](#)」の「」を参照してください。

list-journal-kinesis-streams-for-ledger

次の例は、list-journal-kinesis-streams-for-ledger を使用する方法を説明しています。

AWS CLI

台帳のジャーナルストリームを一覧表示するには

次のlist-journal-kinesis-streams-for-ledger例では、指定された台帳のジャーナルストリームを一覧表示します。

```

aws qlldb list-journal-kinesis-streams-for-ledger \
  --ledger-name myExampleLedger

```

出力:

```

{
  \"Streams\": [
    {

```

```

    "LedgerName": "myExampleLedger",
    "CreationTime": 1591221984.677,
    "InclusiveStartTime": 1590710400.0,
    "ExclusiveEndTime": 1590796799.0,
    "RoleArn": "arn:aws:iam::123456789012:role/my-kinesis-stream-role",
    "StreamId": "7ISCKqwe4y25YyHLzYUFaf",
    "Arn": "arn:aws:qldb:us-east-1:123456789012:stream/
myExampleLedger/7ISCKqwe4y25YyHLzYUFaf",
    "Status": "ACTIVE",
    "KinesisConfiguration": {
      "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-
for-qldb",
      "AggregationEnabled": true
    },
    "StreamName": "myExampleLedger-stream"
  }
]
}

```

詳細については、[「Amazon デベロッパーガイド」の「Amazon からのジャーナルデータのストリーミングQLDB」](#)を参照してください。QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス [ListJournalKinesisStreamsForLedger](#)」の「」を参照してください。

list-journal-s3-exports-for-ledger

次の例は、list-journal-s3-exports-for-ledger を使用する方法を説明しています。

AWS CLI

台帳のジャーナルエクスポートジョブを一覧表示するには

次のlist-journal-s3-exports-for-ledger例では、指定された台帳のジャーナルエクスポートジョブを一覧表示します。

```
aws qldb list-journal-s3-exports-for-ledger \
  --name myExampleLedger
```

出力:

```
{
  "JournalS3Exports": [
```

```
{
  "LedgerName": "myExampleLedger",
  "ExclusiveEndTime": 1568847599.0,
  "ExportCreationTime": 1568847801.418,
  "S3ExportConfiguration": {
    "Bucket": "awsExampleBucket",
    "Prefix": "ledgerexport1/",
    "EncryptionConfiguration": {
      "ObjectEncryptionType": "SSE_S3"
    }
  },
  "ExportId": "ADR20NPKN5LINYGb4dp7yZ",
  "RoleArn": "arn:aws:iam::123456789012:role/qlldb-s3-export",
  "InclusiveStartTime": 1568764800.0,
  "Status": "IN_PROGRESS"
}
]
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのジャーナルのエクスポート QLDB」](#)を参照してください。 QLDB

- API 詳細については、「コマンドリファレンス」の[ListJournalS3ExportsForLedger](#)を参照してください。 AWS CLI

list-journal-s3-exports

次の例は、list-journal-s3-exports を使用する方法を説明しています。

AWS CLI

ジャーナルエクスポートジョブを一覧表示するには

次のlist-journal-s3-exports例では、現在の AWS アカウントとリージョンに関連付けられているすべての台帳のジャーナルエクスポートジョブを一覧表示します。

```
aws qlldb list-journal-s3-exports
```

出力:

```
{
  "JournalS3Exports": [
    {
```

```
"Status": "IN_PROGRESS",
"LedgerName": "myExampleLedger",
"S3ExportConfiguration": {
  "EncryptionConfiguration": {
    "ObjectEncryptionType": "SSE_S3"
  },
  "Bucket": "awsExampleBucket",
  "Prefix": "ledgerexport1/"
},
"RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
"ExportCreationTime": 1568847801.418,
"ExportId": "ADR20NPKN5LINYGb4dp7yZ",
"InclusiveStartTime": 1568764800.0,
"ExclusiveEndTime": 1568847599.0
},
{
  "Status": "COMPLETED",
  "LedgerName": "myExampleLedger2",
  "S3ExportConfiguration": {
    "EncryptionConfiguration": {
      "ObjectEncryptionType": "SSE_S3"
    },
    "Bucket": "awsExampleBucket",
    "Prefix": "ledgerexport1/"
  },
  "RoleArn": "arn:aws:iam::123456789012:role/my-s3-export-role",
  "ExportCreationTime": 1568846847.638,
  "ExportId": "2pdvW8UQrjBAiYTMehEJDI",
  "InclusiveStartTime": 1568592000.0,
  "ExclusiveEndTime": 1568764800.0
}
]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon でのジャーナルのエクスポート QLDB」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI コマンドリファレンスの[ListJournalS3Exports](#)を参照してください。

list-ledgers

次のコード例は、list-ledgers を使用する方法を示しています。

AWS CLI

使用可能な台帳を一覧表示するには

次のlist-ledgers例では、現在の AWS アカウントとリージョンに関連付けられているすべての台帳を一覧表示します。

```
aws qlldb list-ledgers
```

出力:

```
{
  "Ledgers": [
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839243.951,
      "Name": "myExampleLedger"
    },
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839543.557,
      "Name": "myExampleLedger2"
    }
  ]
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDB Ledgers の基本オペレーション」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLedgers](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

台帳にアタッチされたタグを一覧表示するには

次のlist-tags-for-resource例では、指定された台帳にアタッチされたすべてのタグを一覧表示します。


```
aws qlldb list-tags-for-resource \  
  --resource-arn arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger
```

出力:

```
{  
  "Tags": {  
    "IsTest": "true",  
    "Domain": "Test"  
  }  
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDBリソースのタグ付け」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

stream-journal-to-kinesis

次のコード例は、stream-journal-to-kinesis を使用する方法を示しています。

AWS CLI

例 1: 入力ファイルを使用してジャーナルデータを Kinesis Data Streams にストリーミングするには

次のstream-journal-to-kinesis例では、という名前の台帳から、指定された日付と時刻の範囲内でジャーナルデータのストリームを作成しますmyExampleLedger。ストリームは、指定された Amazon Kinesis データストリームにデータを送信します。

```
aws qlldb stream-journal-to-kinesis \  
  --ledger-name myExampleLedger \  
  --inclusive-start-time 2020-05-29T00:00:00Z \  
  --exclusive-end-time 2020-05-29T23:59:59Z \  
  --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \  
  --kinesis-configuration file://my-kinesis-config.json \  
  --stream-name myExampleLedger-stream
```

my-kinesis-config.json の内容:

```
{
  "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb",
  "AggregationEnabled": true
}
```

出力:

```
{
  "StreamId": "7ISCKqwe4y25YyHLzYUFAf"
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon からのジャーナルデータのストリーミングQLDB」](#)を参照してください。 QLDB

例 2: 簡略構文を使用してジャーナルデータを Kinesis Data Streams にストリーミングするには

次のstream-journal-to-kinesis例では、という名前の台帳から、指定された日付と時刻の範囲内でジャーナルデータのストリームを作成しますmyExampleLedger。ストリームは、指定された Amazon Kinesis データストリームにデータを送信します。

```
aws qldb stream-journal-to-kinesis \
  --ledger-name myExampleLedger \
  --inclusive-start-time 2020-05-29T00:00:00Z \
  --exclusive-end-time 2020-05-29T23:59:59Z \
  --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
  --stream-name myExampleLedger-stream \
  --kinesis-configuration StreamArn=arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb,AggregationEnabled=true
```

出力:

```
{
  "StreamId": "7ISCKqwe4y25YyHLzYUFAf"
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon からのジャーナルデータのストリーミングQLDB」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[StreamJournalToKinesis](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

台帳にタグを付けるには

次のtag-resource例では、指定された台帳に一連のタグを追加します。

```
aws qldb tag-resource \  
  --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tags IsTest=true,Domain=Test
```

このコマンドでは何も出力されません。

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDBリソースのタグ付け」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースからタグを削除する

次のuntag-resource例では、指定されたタグキーを持つタグを指定された台帳から削除します。

```
aws qldb untag-resource \  
  --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tag-keys IsTest Domain
```

このコマンドでは何も出力されません。

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDBリソースのタグ付け」](#)を参照してください。 QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-ledger-permissions-mode

次の例は、update-ledger-permissions-mode を使用する方法を説明しています。

AWS CLI

例 1: 台帳のアクセス許可モードを に更新するには STANDARD

次のupdate-ledger-permissions-mode例では、指定された台帳にSTANDARDアクセス許可モードを割り当てます。

```
aws qldb update-ledger-permissions-mode \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

出力:

```
{  
  "Name": "myExampleLedger",  
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "PermissionsMode": "STANDARD"  
}
```

例 2: 台帳のアクセス許可モードを ALLOW_ に更新するには ALL

次のupdate-ledger-permissions-mode例では、指定された台帳にALLOW_ALLアクセス許可モードを割り当てます。

```
aws qldb update-ledger-permissions-mode \  
  --name myExampleLedger \  
  --permissions-mode ALLOW_ALL
```

出力:

```
{  
  "Name": "myExampleLedger",
```

```
"Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
"PermissionsMode": "ALLOW_ALL"
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDB Ledgers の基本オペレーション」](#)を参照してください。QLDB

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateLedgerPermissionsMode](#)」の「」を参照してください。

update-ledger

次のコード例は、update-ledger を使用する方法を示しています。

AWS CLI

例 1: 台帳の削除保護プロパティを更新するには

次のupdate-ledger例では、指定された台帳を更新して、削除保護機能を無効にします。

```
aws qldb update-ledger \
  --name myExampleLedger \
  --no-deletion-protection
```

出力:

```
{
  "CreationDateTime": 1568839243.951,
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "DeletionProtection": false,
  "Name": "myExampleLedger",
  "State": "ACTIVE"
}
```

例 2: 台帳のキーを AWS KMSカスタマーマネージドキーに更新するには

次のupdate-ledger例では、指定された台帳を更新して、保管中の暗号化にカスタマーマネージドKMSキーを使用します。

```
aws qldb update-ledger \
```

```
--name myExampleLedger \  
--kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111
```

出力:

```
{  
  "CreationDateTime": 1568839243.951,  
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "DeletionProtection": false,  
  "Name": "myExampleLedger",  
  "State": "ACTIVE",  
  "EncryptionDescription": {  
    "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111",  
    "EncryptionStatus": "UPDATING"  
  }  
}
```

例 3: 台帳のキーを AWS KMS AWS オウンドキーに更新するには

次のupdate-ledger例では、指定された台帳を更新して、保管中の暗号化に AWS 所有KMS キーを使用します。

```
aws qldb update-ledger \  
  --name myExampleLedger \  
  --kms-key AWS_OWNED_KMS_KEY
```

出力:

```
{  
  "CreationDateTime": 1568839243.951,  
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "DeletionProtection": false,  
  "Name": "myExampleLedger",  
  "State": "ACTIVE",  
  "EncryptionDescription": {  
    "KmsKeyArn": "AWS_OWNED_KMS_KEY",  
    "EncryptionStatus": "UPDATING"  
  }  
}
```

詳細については、[「Amazon デベロッパーガイド」の「Amazon QLDB Ledgers の基本オペレーション」](#)を参照してください。QLDB

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateLedger](#)」の「」を参照してください。

を使用した Amazon RDS の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますRDS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-option-to-option-group

次のコード例は、add-option-to-option-group を使用する方法を示しています。

AWS CLI

オプションをオプショングループに追加するには

次のadd-option-to-option-group例では、指定されたオプショングループにオプションを追加します。

```
aws rds add-option-to-option-group \  
  --option-group-name myoptiongroup \  
  --options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \  
  --apply-immediately
```

出力:

```

{
  "OptionGroup": {
    "OptionGroupName": "myoptiongroup",
    "OptionGroupDescription": "Test Option Group",
    "EngineName": "oracle-ee",
    "MajorEngineVersion": "12.1",
    "Options": [
      {
        "OptionName": "Timezone",
        "OptionDescription": "Change time zone",
        "Persistent": true,
        "Permanent": false,
        "OptionSettings": [
          {
            "Name": "TIME_ZONE",
            "Value": "Australia/Sydney",
            "DefaultValue": "UTC",
            "Description": "Specifies the timezone the user wants to
change the system time to",
            "ApplyType": "DYNAMIC",
            "DataType": "STRING",
            "AllowedValues": "Africa/Cairo,Africa/Casablanca,Africa/
Harare,Africa/Lagos,Africa/Luanda,Africa/Monrovia,Africa/Nairobi,Africa/
Tripoli,Africa/Windhoek,America/Araguaina,America/Argentina/Buenos_Aires,America/
Asuncion,America/Bogota,America/Caracas,America/Chicago,America/Chihuahua,America/
Cuiaba,America/Denver,America/Detroit,America/Fortaleza,America/Godthab,America/
Guatemala,America/Halifax,America/Lima,America/Los_Angeles,America/Manaus,America/
Matamoros,America/Mexico_City,America/Monterrey,America/Montevideo,America/
New_York,America/Phoenix,America/Santiago,America/Sao_Paulo,America/Tijuana,America/
Toronto,Asia/Amman,Asia/Ashgabat,Asia/Baghdad,Asia/Baku,Asia/Bangkok,Asia/
Beirut,Asia/Calcutta,Asia/Damascus,Asia/Dhaka,Asia/Hong_Kong,Asia/Irkutsk,Asia/
Jakarta,Asia/Jerusalem,Asia/Kabul,Asia/Karachi,Asia/Kathmandu,Asia/Kolkata,Asia/
Krasnoyarsk,Asia/Magadan,Asia/Manila,Asia/Muscat,Asia/Novosibirsk,Asia/Rangoon,Asia/
Riyadh,Asia/Seoul,Asia/Shanghai,Asia/Singapore,Asia/Taipei,Asia/Tehran,Asia/
Tokyo,Asia/Ulaanbaatar,Asia/Vladivostok,Asia/Yakutsk,Asia/Yerevan,Atlantic/
Azores,Atlantic/Cape_Verde,Australia/Adelaide,Australia/Brisbane,Australia/
Darwin,Australia/Eucla,Australia/Hobart,Australia/Lord_Howe,Australia/
Perth,Australia/Sydney,Brazil/DeNoronha,Brazil/East,Canada/Newfoundland,Canada/
Saskatchewan,Etc/GMT-3,Europe/Amsterdam,Europe/Athens,Europe/Berlin,Europe/
Dublin,Europe/Helsinki,Europe/Kaliningrad,Europe/London,Europe/Madrid,Europe/
Moscow,Europe/Paris,Europe/Prague,Europe/Rome,Europe/Sarajevo,Pacific/Apia,Pacific/
Auckland,Pacific/Chatham,Pacific/Fiji,Pacific/Guam,Pacific/Honolulu,Pacific/

```



```

Kiritimati,Pacific/Marquesas,Pacific/Samoa,Pacific/Tongatapu,Pacific/Wake,US/
Alaska,US/Central,US/East-Indiana,US/Eastern,US/Pacific,UTC",
        "IsModifiable": true,
        "IsCollection": false
    }
],
"DBSecurityGroupMemberships": [],
"VpcSecurityGroupMemberships": []
},
{
    "OptionName": "OEM",
    "OptionDescription": "Oracle 12c EM Express",
    "Persistent": false,
    "Permanent": false,
    "Port": 5500,
    "OptionSettings": [],
    "DBSecurityGroupMemberships": [
        {
            "DBSecurityGroupName": "default",
            "Status": "authorized"
        }
    ],
    "VpcSecurityGroupMemberships": []
}
],
"AllowsVpcAndNonVpcInstanceMemberships": false,
"OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
}
}

```

詳細については、「Amazon RDSユーザーガイド」の「[オプショングループにオプションを追加する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddOptionToOptionGroup](#)」の「」を参照してください。

add-role-to-db-cluster

次のコード例は、add-role-to-db-cluster を使用する方法を示しています。

AWS CLI

AWS Identity and Access Management (IAM) ロールを DB クラスターに関連付けるには

次のadd-role-to-db-cluster例では、ロールを DB クラスターに関連付けます。

```
aws rds add-role-to-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターとIAMロールの関連付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddRoleToDbCluster](#)」の「」を参照してください。

add-role-to-db-instance

次のコード例は、add-role-to-db-instance を使用する方法を示しています。

AWS CLI

AWS Identity and Access Management (IAM) ロールを DB インスタンスに関連付けるには

次のadd-role-to-db-instance例では、という名前の Oracle DB インスタンスに ロールを追加しますtest-instance。

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier test-instance \  
  --feature-name S3_INTEGRATION \  
  --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

このコマンドでは何も出力されません。

詳細については、[「Amazon ユーザーガイド」の「Amazon RDS Oracle と Amazon S3 の統合の前提条件」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[AddRoleToDbInstance](#)」の「」を参照してください。

add-source-identifier-to-subscription

次の例は、add-source-identifier-to-subscription を使用する方法を説明しています。

AWS CLI

サブスクリプションにソース識別子を追加するには

次のadd-source-identifier例では、既存のサブスクリプションに別のソース識別子を追加します。

```
aws rds add-source-identifier-to-subscription \  
  --subscription-name my-instance-events \  
  --source-identifier test-instance-repl
```

出力:

```
{  
  "EventSubscription": {  
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",  
    "CustSubscriptionId": "my-instance-events",  
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-  
events",  
    "Enabled": false,  
    "Status": "modifying",  
    "EventCategoriesList": [  
      "backup",  
      "recovery"  
    ],  
    "CustomerAwsId": "123456789012",  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",  
    "SourceType": "db-instance",  
    "SourceIdsList": [  
      "test-instance",  
      "test-instance-repl"  
    ]  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AddSourceIdentifierToSubscription](#)」の「」を参照してください。

add-tags-to-resource

次のコード例は、add-tags-to-resource を使用する方法を示しています。

AWS CLI

リソースにタグを追加するには

次のadd-tags-to-resource例では、RDSデータベースにタグを追加します。

```
aws rds add-tags-to-resource \  
  --resource-name arn:aws:rds:us-east-1:123456789012:db:database-mysql \  
  --tags "[{\"Key\": \"Name\", \"Value\": \"MyDatabase\"}, {\"Key\": \"Environment\", \"Value\": \"test\"}]"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon ユーザーガイド」の「Amazon RDSリソースのタグ付け」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTagsToResource](#)」の「」を参照してください。

apply-pending-maintenance-action

次の例は、apply-pending-maintenance-action を使用する方法を説明しています。

AWS CLI

保留中のメンテナンスアクションを適用するには

次のapply-pending-maintenance-action例では、DB クラスターの保留中のメンテナンスアクションを適用します。

```
aws rds apply-pending-maintenance-action \  
  --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:my-db-cluster \  
  --apply-action system-update \  
  --opt-in-type immediate
```

出力:

```
{  
  "ResourcePendingMaintenanceActions": {  
    "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:cluster:my-db-cluster",  
    "PendingMaintenanceActionDetails": [  
      {
```

```

        "Action": "system-update",
        "OptInStatus": "immediate",
        "CurrentApplyDate": "2021-01-23T01:07:36.100Z",
        "Description": "Upgrade to Aurora PostgreSQL 3.3.2"
    }
]
}
}

```

詳細については、[「Amazon ユーザーガイド」の「DB インスタンスの維持」](#)および[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターの維持」](#)を参照してください。

RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[ApplyPendingMaintenanceAction](#)」の「」を参照してください。

authorize-db-security-group-ingress

次の例は、`authorize-db-security-group-ingress` を使用する方法を説明しています。

AWS CLI

AWS Identity and Access Management (IAM) ロールを DB インスタンスに関連付けるには

次の`authorize-db-security-group-ingress`例では、CIDRIP 範囲 `192.0.2.0/24` のイングレスルールを使用してデフォルトのセキュリティグループを設定します。

```

aws rds authorize-db-security-group-ingress \
  --db-security-group-name default \
  --cidrip 192.0.2.0/24

```

出力:

```

{
  "DBSecurityGroup": {
    "OwnerId": "123456789012",
    "DBSecurityGroupName": "default",
    "DBSecurityGroupDescription": "default",
    "EC2SecurityGroups": [],
    "IPRanges": [
      {
        "Status": "authorizing",

```

```
        "CIDRIP": "192.0.2.0/24"
      }
    ],
    "DBSecurityGroupArn": "arn:aws:rds:us-east-1:111122223333:secgrp:default"
  }
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[IP 範囲からの DB セキュリティグループへのネットワークアクセスの承認](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AuthorizeDbSecurityGroupIngress](#)」の「」を参照してください。

backtrack-db-cluster

次の例は、backtrack-db-cluster を使用する方法を説明しています。

AWS CLI

Aurora DB クラスターをバックトラックするには

次のbacktrack-db-cluster例では、指定された DB クラスターの sample-cluster を 2018 年 3 月 19 日午前 10 時にバックトラックします。

```
aws rds backtrack-db-cluster --db-cluster-identifier sample-cluster --backtrack-  
to 2018-03-19T10:00:00+00:00
```

このコマンドは、RDSリソースへの変更を承認するJSONブロックを出力します。

- API 詳細については、AWS CLI 「コマンドリファレンス[BacktrackDbCluster](#)」の「」を参照してください。

cancel-export-task

次の例は、cancel-export-task を使用する方法を説明しています。

AWS CLI

Amazon S3 へのスナップショットのエクスポートをキャンセルするには

次のcancel-export-task例では、スナップショットを Amazon S3 にエクスポートする進行中のエクスポートタスクをキャンセルします。

```
aws rds cancel-export-task \  
--export-task-identifier my-s3-export-1
```

出力:

```
{  
  "ExportTaskIdentifier": "my-s3-export-1",  
  "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:publisher-final-snapshot",  
  "SnapshotTime": "2019-03-24T20:01:09.815Z",  
  "S3Bucket": "mybucket",  
  "S3Prefix": "",  
  "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/export-snap-S3-role",  
  "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/abcd0000-7bfd-4594-af38-aabbccddeeff",  
  "Status": "CANCELING",  
  "PercentProgress": 0,  
  "TotalExtractedDataInGB": 0  
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[スナップショットエクスポートタスクのキャンセル](#)」または「[Amazon Aurora ユーザーガイド](#)」の「[スナップショットエクスポートタスクのキャンセル](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelExportTask](#)」の「」を参照してください。

copy-db-cluster-parameter-group

次のコード例は、copy-db-cluster-parameter-group を使用する方法を示しています。

AWS CLI

DB クラスターパラメータグループをコピーするには

次のcopy-db-cluster-parameter-group例では、DB クラスターパラメータグループのコピーを作成します。

```
aws rds copy-db-cluster-parameter-group \  
--source-db-cluster-parameter-group-identifier mydbclusterpg \  
--target-db-cluster-parameter-group-identifier mydbclusterpgcopy \  
--target-db-cluster-parameter-group-name mydbclusterpgcopy
```

```
--target-db-cluster-parameter-group-description "Copy of mydbclusterpg parameter group"
```

出力:

```
{
  "DBClusterParameterGroup": {
    "DBClusterParameterGroupName": "mydbclusterpgcopy",
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-pg:mydbclusterpgcopy",
    "DBParameterGroupFamily": "aurora-mysql5.7",
    "Description": "Copy of mydbclusterpg parameter group"
  }
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB クラスターパラメータグループのコピー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyDbClusterParameterGroup](#)」の「」を参照してください。

copy-db-cluster-snapshot

次の例は、copy-db-cluster-snapshot を使用する方法を説明しています。

AWS CLI

DB クラスタースナップショットをコピーするには

次のcopy-db-cluster-snapshot例では、タグを含む DB クラスタースナップショットのコピーを作成します。

```
aws rds copy-db-cluster-snapshot \  
  --source-db-cluster-snapshot-identifier arn:aws:rds:us-east-1:123456789012:cluster-snapshot:rds:myaurora-2019-06-04-09-16 \  
  --target-db-cluster-snapshot-identifier myclustersnapshotcopy \  
  --copy-tags
```

出力:

```
{
```



```
"DBClusterSnapshot": {
  "AvailabilityZones": [
    "us-east-1a",
    "us-east-1b",
    "us-east-1e"
  ],
  "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
  "DBClusterIdentifier": "myaurora",
  "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
  "Engine": "aurora-mysql",
  "AllocatedStorage": 0,
  "Status": "available",
  "Port": 0,
  "VpcId": "vpc-6594f31c",
  "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
  "MasterUsername": "myadmin",
  "EngineVersion": "5.7.mysql_aurora.2.04.2",
  "LicenseModel": "aurora-mysql",
  "SnapshotType": "manual",
  "PercentProgress": 100,
  "StorageEncrypted": true,
  "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
  "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:myclustersnapshotcopy",
  "IAMDatabaseAuthenticationEnabled": false
}
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[スナップショットのコピー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyDbClusterSnapshot](#)」の「」を参照してください。

copy-db-parameter-group

次のコード例は、copy-db-parameter-group を使用する方法を示しています。

AWS CLI

DB クラスターパラメータグループをコピーするには

次のcopy-db-parameter-group例では、DB パラメータグループのコピーを作成します。

```
aws rds copy-db-parameter-group \  
  --source-db-parameter-group-identifier mydbpg \  
  --target-db-parameter-group-identifier mydbpgcopy \  
  --target-db-parameter-group-description "Copy of mydbpg parameter group"
```

出力:

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbpgcopy",  
    "DBParameterGroupArn": "arn:aws:rds:us-east-1:814387698303:pg:mydbpgcopy",  
    "DBParameterGroupFamily": "mysql5.7",  
    "Description": "Copy of mydbpg parameter group"  
  }  
}
```

詳細については、「Amazon RDSユーザーガイド」の「[DB パラメータグループのコピー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyDbParameterGroup](#)」の「」を参照してください。

copy-db-snapshot

次のコード例は、copy-db-snapshot を使用する方法を示しています。

AWS CLI

DB スナップショットをコピーするには

次のcopy-db-snapshot例では、DB スナップショットのコピーを作成します。

```
aws rds copy-db-snapshot \  
  --source-db-snapshot-identifier rds:database-mysql-2019-06-06-08-38  
  --target-db-snapshot-identifier mydbsnapshotcopy
```

出力:

```
{  
  "DBSnapshot": {  
    "VpcId": "vpc-6594f31c",  
    "Status": "creating",
```

```

    "Encrypted": true,
    "SourceDBSnapshotIdentifier": "arn:aws:rds:us-
east-1:123456789012:snapshot:rds:database-mysql-2019-06-06-08-38",
    "MasterUsername": "admin",
    "Iops": 1000,
    "Port": 3306,
    "LicenseModel": "general-public-license",
    "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshotcopy",
    "EngineVersion": "5.6.40",
    "OptionGroupName": "default:mysql-5-6",
    "ProcessorFeatures": [],
    "Engine": "mysql",
    "StorageType": "io1",
    "DbiResourceId": "db-ZI7UJ5BLKMBYFGX7FDENCKADC4",
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
    "SnapshotType": "manual",
    "IAMDatabaseAuthenticationEnabled": false,
    "SourceRegion": "us-east-1",
    "DBInstanceIdentifier": "database-mysql",
    "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
    "AvailabilityZone": "us-east-1f",
    "PercentProgress": 0,
    "AllocatedStorage": 100,
    "DBSnapshotIdentifier": "mydbsnapshotcopy"
  }
}

```

詳細については、「Amazon RDSユーザーガイド」の「[スナップショットのコピー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyDbSnapshot](#)」の「」を参照してください。

copy-option-group

次の例は、copy-option-group を使用する方法を説明しています。

AWS CLI

オプショングループをコピーするには

次のcopy-option-group例では、オプショングループのコピーを作成します。

```
aws rds copy-option-group \  
  --source-option-group-identifier myoptiongroup \  
  --target-option-group-identifier new-option-group \  
  --target-option-group-description "My option group copy"
```

出力:

```
{  
  "OptionGroup": {  
    "Options": [],  
    "OptionGroupName": "new-option-group",  
    "MajorEngineVersion": "11.2",  
    "OptionGroupDescription": "My option group copy",  
    "AllowsVpcAndNonVpcInstanceMemberships": true,  
    "EngineName": "oracle-ee",  
    "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:new-option-group"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「オプショングループのコピーの作成」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyOptionGroup](#)」の「」を参照してください。

create-blue-green-deployment

次の例は、create-blue-green-deployment を使用する方法を説明しています。

AWS CLI

例 1: RDS for MySQL DB インスタンスのブルー/グリーンデプロイを作成するには

次のcreate-blue-green-deployment例では、MySQL DB インスタンスのブルー/グリーンデプロイを作成します。

```
aws rds create-blue-green-deployment \  
  --blue-green-deployment-name bgd-cli-test-instance \  
  --source arn:aws:rds:us-east-1:123456789012:db:my-db-instance \  
  --target-engine-version 8.0 \  
  --target-db-parameter-group-name mysql-80-group
```

出力:

```
{
  "BlueGreenDeployment": {
    "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
    "BlueGreenDeploymentName": "bgd-cli-test-instance",
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3"
      }
    ],
    "Tasks": [
      {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "PENDING"
      },
      {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "PENDING"
      },
      {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "PENDING"
      },
      {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "PENDING"
      }
    ],
    "Status": "PROVISIONING",
  }
}
```

```
    "CreateTime": "2022-02-25T21:18:51.183000+00:00"
  }
}
```

詳細については、「Amazon RDSユーザーガイド」の[「ブルー/グリーンデプロイの作成」](#)を参照してください。

例 2: Aurora MySQL DB クラスターのブルー/グリーンデプロイを作成するには

次のcreate-blue-green-deployment例では、Aurora MySQL DB クラスターのブルー/グリーンデプロイを作成します。

```
aws rds create-blue-green-deployment \
  --blue-green-deployment-name my-blue-green-deployment \
  --source arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster \
  --target-engine-version 8.0 \
  --target-db-cluster-parameter-group-name ams-80-binlog-enabled \
  --target-db-parameter-group-name mysql-80-cluster-group
```

出力:

```
{
  "BlueGreenDeployment": {
    "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
    "BlueGreenDeploymentName": "my-blue-green-deployment",
    "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
        "Status": "PROVISIONING"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1",
        "Status": "PROVISIONING"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2",
        "Status": "PROVISIONING"
      }
    ]
  }
}
```

```
{
  "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-3",
  "Status": "PROVISIONING"
},
{
  "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
excluded-member-endpoint",
  "Status": "PROVISIONING"
},
{
  "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
reader-endpoint",
  "Status": "PROVISIONING"
}
],
"Tasks": [
  {
    "Name": "CREATING_READ_REPLICA_OF_SOURCE",
    "Status": "PENDING"
  },
  {
    "Name": "DB_ENGINE_VERSION_UPGRADE",
    "Status": "PENDING"
  },
  {
    "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
    "Status": "PENDING"
  },
  {
    "Name": "CREATE_CUSTOM_ENDPOINTS",
    "Status": "PENDING"
  }
],
"Status": "PROVISIONING",
"CreateTime": "2022-02-25T21:12:00.288000+00:00"
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[ブルー/グリーンデプロイの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateBlueGreenDeployment](#)」の「」を参照してください。

create-db-cluster-endpoint

次の例は、create-db-cluster-endpoint を使用する方法を説明しています。

AWS CLI

カスタム DB クラスターエンドポイントを作成するには

次のcreate-db-cluster-endpoint例では、カスタム DB クラスターエンドポイントを作成し、指定された Aurora DB クラスターに関連付けます。

```
aws rds create-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint \  
  --endpoint-type reader \  
  --db-cluster-identifier mydbcluster \  
  --static-members dbinstance1 dbinstance2
```

出力:

```
{  
  "DBClusterEndpointIdentifier": "mycustomendpoint",  
  "DBClusterIdentifier": "mydbcluster",  
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",  
  "Endpoint": "mycustomendpoint.cluster-custom-cnexample.us-  
east-1.rds.amazonaws.com",  
  "Status": "creating",  
  "EndpointType": "CUSTOM",  
  "CustomEndpointType": "READER",  
  "StaticMembers": [  
    "dbinstance1",  
    "dbinstance2"  
  ],  
  "ExcludedMembers": [],  
  "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
endpoint:mycustomendpoint"  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora Connection Management」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbClusterEndpoint](#)」の「」を参照してください。

create-db-cluster-parameter-group

次の例は、create-db-cluster-parameter-group を使用する方法を説明しています。

AWS CLI

DB クラスターパラメータグループを作成するには

次のcreate-db-cluster-parameter-group例では、DB クラスターパラメータグループを作成します。

```
aws rds create-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterparametergroup \  
  --db-parameter-group-family aurora5.6 \  
  --description "My new cluster parameter group"
```

出力:

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "mydbclusterparametergroup",  
    "DBParameterGroupFamily": "aurora5.6",  
    "Description": "My new cluster parameter group",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
pg:mydbclusterparametergroup"  
  }  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「DB クラスターパラメータグループの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbClusterParameterGroup](#)」の「」を参照してください。

create-db-cluster-snapshot

次の例は、create-db-cluster-snapshot を使用する方法を説明しています。

AWS CLI

DB クラスタースナップショットを作成するには

次のcreate-db-cluster-snapshot例では、DB クラスタースナップショットを作成します。

```
aws rds create-db-cluster-snapshot \  
  --db-cluster-identifier mydbcluster \  
  --db-cluster-snapshot-identifier mydbclustersnapshot
```

出力:

```
{  
  "DBClusterSnapshot": {  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1e"  
    ],  
    "DBClusterSnapshotIdentifier": "mydbclustersnapshot",  
    "DBClusterIdentifier": "mydbcluster",  
    "SnapshotCreateTime": "2019-06-18T21:21:00.469Z",  
    "Engine": "aurora-mysql",  
    "AllocatedStorage": 1,  
    "Status": "creating",  
    "Port": 0,  
    "VpcId": "vpc-6594f31c",  
    "ClusterCreateTime": "2019-04-15T14:18:42.785Z",  
    "MasterUsername": "myadmin",  
    "EngineVersion": "5.7.mysql_aurora.2.04.2",  
    "LicenseModel": "aurora-mysql",  
    "SnapshotType": "manual",  
    "PercentProgress": 0,  
    "StorageEncrypted": true,  
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",  
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
snapshot:mydbclustersnapshot",  
    "IAMDatabaseAuthenticationEnabled": false  
  }  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB クラスタースナップショットの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbClusterSnapshot](#)」の「」を参照してください。

create-db-cluster

次の例は、create-db-cluster を使用する方法を説明しています。

AWS CLI

例 1: MySQL 5.7 互換 DB クラスターを作成するには

次のcreate-db-cluster例では、デフォルトのエンジンバージョンを使用して MySQL 5.7 互換 DB クラスターを作成します。サンプルパスワードを安全なパスワードsecret99に置き換えます。コンソールを使用して DB クラスターを作成すると、Amazon は DB クラスターのライター DB インスタンスRDSを自動的に作成します。ただし、を使用して AWS CLI DB クラスターを作成する場合は、create-db-instance AWS CLI コマンドを使用して DB クラスターのライター DB インスタンスを明示的に作成する必要があります。

```
aws rds create-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --engine aurora-mysql \  
  --engine-version 5.7 \  
  --master-username admin \  
  --master-user-password secret99 \  
  --db-subnet-group-name default \  
  --vpc-security-group-ids sg-0b9130572daf3dc16
```

出力:

```
{  
  "DBCluster": {  
    "DBSubnetGroup": "default",  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-0b9130572daf3dc16",  
        "Status": "active"  
      }  
    ],  
    "AllocatedStorage": 1,  
    "AssociatedRoles": [],  
    "PreferredBackupWindow": "09:12-09:42",  
    "ClusterCreateTime": "2023-02-27T23:21:33.048Z",  
    "DeletionProtection": false,  
    "IAMDatabaseAuthenticationEnabled": false,  
    "ReadReplicaIdentifiers": [],
```

```

    "EngineMode": "provisioned",
    "Engine": "aurora-mysql",
    "StorageEncrypted": false,
    "MultiAZ": false,
    "PreferredMaintenanceWindow": "mon:04:31-mon:05:01",
    "HttpEndpointEnabled": false,
    "BackupRetentionPeriod": 1,
    "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",
    "DBClusterIdentifier": "sample-cluster",
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1e"
    ],
    "MasterUsername": "master",
    "EngineVersion": "5.7.mysql_aurora.2.11.1",
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
    "DBClusterMembers": [],
    "Port": 3306,
    "Status": "creating",
    "Endpoint": "sample-cluster.cluster-cnpxexample.us-east-1.rds.amazonaws.com",
    "DBClusterParameterGroup": "default.aurora-mysql5.7",
    "HostedZoneId": "Z2R2ITUGPM61AM",
    "ReaderEndpoint": "sample-cluster.cluster-ro-cnpxexample.us-
east-1.rds.amazonaws.com",
    "CopyTagsToSnapshot": false
  }
}

```

例 2: PostgreSQL 互換 DB クラスターを作成するには

次の `create-db-cluster` 例では、デフォルトのエンジンバージョンを使用して PostgreSQL 互換 DB クラスターを作成します。サンプルパスワードを安全なパスワード `secret99` に置き換えます。コンソールを使用して DB クラスターを作成すると、Amazon は DB クラスターのライター DB インスタンス RDS を自動的に作成します。ただし、を使用して AWS CLI DB クラスターを作成する場合は、`create-db-instance` AWS CLI コマンドを使用して DB クラスターのライター DB インスタンスを明示的に作成する必要があります。

```

aws rds create-db-cluster \
  --db-cluster-identifier sample-pg-cluster \
  --engine aurora-postgresql \
  --master-username master \
  --master-user-password secret99 \

```

```
--db-subnet-group-name default \  
--vpc-security-group-ids sg-0b9130572daf3dc16
```

出力:

```
{  
  "DBCluster": {  
    "Endpoint": "sample-pg-cluster.cluster-cnpxample.us-  
east-1.rds.amazonaws.com",  
    "HttpEndpointEnabled": false,  
    "DBClusterMembers": [],  
    "EngineMode": "provisioned",  
    "CopyTagsToSnapshot": false,  
    "HostedZoneId": "Z2R2ITUGPM61AM",  
    "IAMDatabaseAuthenticationEnabled": false,  
    "AllocatedStorage": 1,  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-0b9130572daf3dc16",  
        "Status": "active"  
      }  
    ],  
    "DeletionProtection": false,  
    "StorageEncrypted": false,  
    "BackupRetentionPeriod": 1,  
    "PreferredBackupWindow": "09:56-10:26",  
    "ClusterCreateTime": "2023-02-27T23:26:08.371Z",  
    "DBClusterParameterGroup": "default.aurora-postgresql13",  
    "EngineVersion": "13.7",  
    "Engine": "aurora-postgresql",  
    "Status": "creating",  
    "DBClusterIdentifier": "sample-pg-cluster",  
    "MultiAZ": false,  
    "Port": 5432,  
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-pg-  
cluster",  
    "AssociatedRoles": [],  
    "DbClusterResourceId": "cluster-ANPAJ4AE5446DAEXAMPLE",  
    "PreferredMaintenanceWindow": "wed:03:33-wed:04:03",  
    "ReaderEndpoint": "sample-pg-cluster.cluster-ro-cnpxample.us-  
east-1.rds.amazonaws.com",  
    "MasterUsername": "master",  
    "AvailabilityZones": [  

```

```
        "us-east-1a",
        "us-east-1b",
        "us-east-1c"
    ],
    "ReadReplicaIdentifiers": [],
    "DBSubnetGroup": "default"
}
}
```

詳細については、[「Amazon Aurora ユーザーガイド」](#)の「[Amazon Aurora DB クラスターの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateDbCluster](#)」の「」を参照してください。

create-db-instance-read-replica

次のコード例は、`create-db-instance-read-replica` を使用する方法を示しています。

AWS CLI

DB インスタンスリードレプリカを作成するには

この例では、`test-instance` という名前の既存の DB インスタンスのリードレプリカを作成します。リードレプリカの名前は `test-instance-repl` です。

```
aws rds create-db-instance-read-replica \
  --db-instance-identifier test-instance-repl \
  --source-db-instance-identifier test-instance
```

出力:

```
{
  "DBInstance": {
    "IAMDatabaseAuthenticationEnabled": false,
    "MonitoringInterval": 0,
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
    "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
    "DBInstanceIdentifier": "test-instance-repl",
    ...some output truncated...
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDbInstanceReadReplica](#)」の「」を参照してください。

create-db-instance

次の例は、create-db-instance を使用する方法を説明しています。

AWS CLI

DB インスタンスを作成するには

次の create-db-instance の例は、必須のオプションを使用して新しい DB インスタンスを起動します。

```
aws rds create-db-instance \  
  --db-instance-identifier test-mysql-instance \  
  --db-instance-class db.t3.micro \  
  --engine mysql \  
  --master-username admin \  
  --master-user-password secret99 \  
  --allocated-storage 20
```

出力:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "test-mysql-instance",  
    "DBInstanceClass": "db.t3.micro",  
    "Engine": "mysql",  
    "DBInstanceStatus": "creating",  
    "MasterUsername": "admin",  
    "AllocatedStorage": 20,  
    "PreferredBackupWindow": "12:55-13:25",  
    "BackupRetentionPeriod": 1,  
    "DBSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-12345abc",  
        "Status": "active"  
      }  
    ],  
    "DBParameterGroups": [  
      {  
        "DBParameterGroupName": "default:mysql5.7",  
        "Status": "in-sync"  
      }  
    ]  
  }  
}
```

```
{
  "DBParameterGroupName": "default.mysql5.7",
  "ParameterApplyStatus": "in-sync"
},
"DBSubnetGroup": {
  "DBSubnetGroupName": "default",
  "DBSubnetGroupDescription": "default",
  "VpcId": "vpc-2ff2ff2f",
  "SubnetGroupStatus": "Complete",
  "Subnets": [
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2c"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2b"
      },
      "SubnetStatus": "Active"
    }
  ]
},
"PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
"PendingModifiedValues": {
  "MasterUserPassword": "*****"
}
```



```
    },
    "MultiAZ": false,
    "EngineVersion": "5.7.22",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",
    "OptionGroupMemberships": [
      {
        "OptionGroupName": "default:mysql-5-7",
        "Status": "in-sync"
      }
    ],
    "PubliclyAccessible": true,
    "StorageType": "gp2",
    "DbInstancePort": 0,
    "StorageEncrypted": false,
    "DbiResourceId": "db-5555EXAMPLE444444444EXAMPLE",
    "CACertificateIdentifier": "rds-ca-2019",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
    "MonitoringInterval": 0,
    "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
    "IAMDatabaseAuthenticationEnabled": false,
    "PerformanceInsightsEnabled": false,
    "DeletionProtection": false,
    "AssociatedRoles": []
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon RDS DB インスタンスの作成」](#)を参照してください。 RDS

- API 詳細については、「コマンドリファレンス」の[「CreateDBInstance」](#)を参照してください。 AWS CLI

create-db-parameter-group

次のコード例は、create-db-parameter-group を使用する方法を示しています。

AWS CLI

DB パラメータグループを作成するには

次の `create-db-parameter-group` の例は、DB パラメータグループを作成します。

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --db-parameter-group-family MySQL5.6 \  
  --description "My new parameter group"
```

出力:

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbparametergroup",  
    "DBParameterGroupFamily": "mysql5.6",  
    "Description": "My new parameter group",  
    "DBParameterGroupArn": "arn:aws:rds:us-  
east-1:123456789012:pg:mydbparametergroup"  
  }  
}
```

詳細については、「Amazon ユーザーガイド」の [「DB パラメータグループの作成」](#) を参照してください。 RDS

- API 詳細については、「コマンドリファレンス」の [「C reateDBParameterグループ」](#) を参照してください。 AWS CLI

create-db-proxy-endpoint

次のコード例は、`create-db-proxy-endpoint` を使用する方法を示しています。

AWS CLI

RDS データベースの DB プロキシエンドポイントを作成するには

次の `create-db-proxy-endpoint` 例では、DB プロキシエンドポイントを作成します。

```
aws rds create-db-proxy-endpoint \  
  --db-proxy-name proxyExample \  
  --db-proxy-endpoint-name "proxyep1" \  
  --vpc-subnet-ids subnetgroup1 subnetgroup2
```

出力:

```
{
  "DBProxyEndpoint": {
    "DBProxyEndpointName": "proxyep1",
    "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
    "DBProxyName": "proxyExample",
    "Status": "creating",
    "VpcId": "vpc-1234567",
    "VpcSecurityGroupIds": [
      "sg-1234",
      "sg-5678"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Endpoint": "proxyep1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "TargetRole": "READ_WRITE",
    "IsDefault": false
  }
}
```

詳細については、「Amazon RDSユーザーガイド」の「[プロキシエンドポイントの作成](#)」および「[Amazon Aurora ユーザーガイド](#)」の「[プロキシエンドポイントの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbProxyEndpoint](#)」の「」を参照してください。

create-db-proxy

次のコード例は、create-db-proxy を使用する方法を示しています。

AWS CLI

RDS データベースの DB プロキシを作成するには

次のcreate-db-proxy例では、DB プロキシを作成します。

```
aws rds create-db-proxy \
```

```

--db-proxy-name proxyExample \
--engine-family MYSQL \
--auth
Description="proxydescription1",AuthScheme="SECRETS",SecretArn="arn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",IAMAuth="DISABLED",ClientPasswordAuthType="MYSO
\
--role-arn arn:aws:iam::123456789123:role/ProxyRole \
--vpc-subnet-ids subnetgroup1 subnetgroup2

```

出力:

```

{
  "DBProxy": {
    "DBProxyName": "proxyExample",
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
    "EngineFamily": "MYSQL",
    "VpcId": "vpc-1234567",
    "VpcSecurityGroupIds": [
      "sg-1234",
      "sg-5678",
      "sg-9101"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Auth": "[
      {
        "Description": "proxydescription1",
        "AuthScheme": "SECRETS",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
        "IAMAuth": "DISABLED"
      }
    ]",
    "RoleArn": "arn:aws:iam::12345678912:role/ProxyRole",
    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
    "RequireTLS": false,
    "IdleClientTimeout": 1800,
    "DebuggingLogging": false,
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "UpdateDate": "2023-04-13T01:49:38.568000+00:00"
  }
}

```

```
}  
}
```

詳細については、「Amazon RDSユーザーガイド」の[RDS「プロキシの作成」](#)と「Amazon Aurora ユーザーガイド」の[RDS「プロキシの作成」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateDbProxy](#)」の「」を参照してください。

create-db-security-group

次のコード例は、create-db-security-group を使用する方法を示しています。

AWS CLI

Amazon RDS DB セキュリティグループを作成するには

次のcreate-db-security-groupコマンドは、新しい Amazon RDS DB セキュリティグループを作成します。

```
aws rds create-db-security-group --db-security-group-name mysecgroup --db-security-group-description "My Test Security Group"
```

この例では、新しい DB セキュリティグループの名前mysecgroupと説明があります。

出力:

```
{  
  "DBSecurityGroup": {  
    "OwnerId": "123456789012",  
    "DBSecurityGroupName": "mysecgroup",  
    "DBSecurityGroupDescription": "My Test Security Group",  
    "VpcId": "vpc-a1b2c3d4",  
    "EC2SecurityGroups": [],  
    "IPRanges": [],  
    "DBSecurityGroupArn": "arn:aws:rds:us-west-2:123456789012:secgrp:mysecgroup"  
  }  
}
```

- API 詳細については、AWS CLI「コマンドリファレンス[CreateDbSecurityGroup](#)」の「」を参照してください。

create-db-shard-group

次のコード例は、create-db-shard-group を使用する方法を示しています。

AWS CLI

例 1: Aurora PostgreSQL プライマリ DB クラスターを作成するには

次のcreate-db-cluster例では、Aurora Serverless v2 および Aurora Limitless Database と互換性のある Aurora PostgreSQL SQLプライマリ DB クラスターを作成します。

```
aws rds create-db-cluster \  
  --db-cluster-identifier my-sv2-cluster \  
  --engine aurora-postgresql \  
  --engine-version 15.2-limitless \  
  --storage-type aurora-iopt1 \  
  --serverless-v2-scaling-configuration MinCapacity=2,MaxCapacity=16 \  
  --enable-limitless-database \  
  --master-username myuser \  
  --master-user-password mypassword \  
  --enable-cloudwatch-logs-exports postgresql
```

出力:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-2b",  
      "us-east-2c",  
      "us-east-2a"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DBClusterIdentifier": "my-sv2-cluster",  
    "DBClusterParameterGroup": "default.aurora-postgresql15",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "my-sv2-cluster.cluster-cekyexample.us-east-2.rds.amazonaws.com",  
    "ReaderEndpoint": "my-sv2-cluster.cluster-ro-cekyexample.us-east-2.rds.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "aurora-postgresql",
```

```
"EngineVersion": "15.2-limitless",
"Port": 5432,
"MasterUsername": "myuser",
"PreferredBackupWindow": "06:05-06:35",
"PreferredMaintenanceWindow": "mon:08:25-mon:08:55",
"ReadReplicaIdentifiers": [],
"DBClusterMembers": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-#####",
    "Status": "active"
  }
],
"HostedZoneId": "Z2XHWR1EXAMPLE",
"StorageEncrypted": false,
"DbClusterResourceId": "cluster-XYEDT6ML6FHIXH4Q2J1EXAMPLE",
"DBClusterArn": "arn:aws:rds:us-east-2:123456789012:cluster:my-sv2-cluster",
"AssociatedRoles": [],
"IAMDatabaseAuthenticationEnabled": false,
"ClusterCreateTime": "2024-02-19T16:24:07.771000+00:00",
"EnabledCloudwatchLogsExports": [
  "postgresql"
],
"EngineMode": "provisioned",
"DeletionProtection": false,
"HttpEndpointEnabled": false,
"CopyTagsToSnapshot": false,
"CrossAccountClone": false,
"DomainMemberships": [],
"TagList": [],
"StorageType": "aurora-iopt1",
"AutoMinorVersionUpgrade": true,
"ServerlessV2ScalingConfiguration": {
  "MinCapacity": 2.0,
  "MaxCapacity": 16.0
},
"NetworkType": "IPV4",
"I00OptimizedNextAllowedModificationTime":
"2024-03-21T16:24:07.781000+00:00",
"LimitlessDatabase": {
  "Status": "not-in-use",
  "MinRequiredACU": 96.0
}
}
```

```
}
```

例 2: プライマリ (ライター) DB インスタンスを作成するには

次のcreate-db-instance例では、Aurora Serverless v2 プライマリ (ライター) DB インスタンスを作成します。コンソールを使用して DB クラスターを作成すると、Amazon は DB クラスターのライター DB インスタンスRDSを自動的に作成します。ただし、を使用して AWS CLI DB クラスターを作成する場合は、create-db-instance AWS CLI コマンドを使用して DB クラスターのライター DB インスタンスを明示的に作成する必要があります。

```
aws rds create-db-instance \  
  --db-instance-identifier my-sv2-instance \  
  --db-cluster-identifier my-sv2-cluster \  
  --engine aurora-postgresql \  
  --db-instance-class db.serverless
```

出力:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "my-sv2-instance",  
    "DBInstanceClass": "db.serverless",  
    "Engine": "aurora-postgresql",  
    "DBInstanceStatus": "creating",  
    "MasterUsername": "myuser",  
    "AllocatedStorage": 1,  
    "PreferredBackupWindow": "06:05-06:35",  
    "BackupRetentionPeriod": 1,  
    "DBSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-#####",  
        "Status": "active"  
      }  
    ],  
    "DBParameterGroups": [  
      {  
        "DBParameterGroupName": "default.aurora-postgresql15",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "DBSubnetGroup": {
```



```
"DBSubnetGroupName": "default",
"DBSubnetGroupDescription": "default",
"VpcId": "vpc-#####",
"SubnetGroupStatus": "Complete",
"Subnets": [
  {
    "SubnetIdentifier": "subnet-#####",
    "SubnetAvailabilityZone": {
      "Name": "us-east-2c"
    },
    "SubnetOutpost": {},
    "SubnetStatus": "Active"
  },
  {
    "SubnetIdentifier": "subnet-#####",
    "SubnetAvailabilityZone": {
      "Name": "us-east-2a"
    },
    "SubnetOutpost": {},
    "SubnetStatus": "Active"
  },
  {
    "SubnetIdentifier": "subnet-#####",
    "SubnetAvailabilityZone": {
      "Name": "us-east-2b"
    },
    "SubnetOutpost": {},
    "SubnetStatus": "Active"
  }
],
"PreferredMaintenanceWindow": "fri:09:01-fri:09:31",
"PendingModifiedValues": {
  "PendingCloudwatchLogsExports": {
    "LogTypesToEnable": [
      "postgresql"
    ]
  }
},
"MultiAZ": false,
"EngineVersion": "15.2-limitless",
"AutoMinorVersionUpgrade": true,
"ReadReplicaDBInstanceIdentifiers": [],
"LicenseModel": "postgresql-license",
```

```

    "OptionGroupMemberships": [
      {
        "OptionGroupName": "default:aurora-postgresql-15",
        "Status": "in-sync"
      }
    ],
    "PubliclyAccessible": false,
    "StorageType": "aurora-iopt1",
    "DbInstancePort": 0,
    "DBClusterIdentifier": "my-sv2-cluster",
    "StorageEncrypted": false,
    "DbiResourceId": "db-BIQTE3B3K3RM7M74SK5EXAMPLE",
    "CACertificateIdentifier": "rds-ca-rsa2048-g1",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
    "MonitoringInterval": 0,
    "PromotionTier": 1,
    "DBInstanceArn": "arn:aws:rds:us-east-2:123456789012:db:my-sv2-instance",
    "IAMDatabaseAuthenticationEnabled": false,
    "PerformanceInsightsEnabled": false,
    "DeletionProtection": false,
    "AssociatedRoles": [],
    "TagList": [],
    "CustomerOwnedIpEnabled": false,
    "BackupTarget": "region",
    "NetworkType": "IPV4",
    "StorageThroughput": 0,
    "CertificateDetails": {
      "CAIdentifier": "rds-ca-rsa2048-g1"
    },
    "DedicatedLogVolume": false
  }
}

```

例 3: DB シャードグループを作成するには

次の `create-db-shard-group` 例では、Aurora PostgreSQL プライマリ DB クラスターに DB シャードグループを作成します。

```

aws rds create-db-shard-group \
  --db-shard-group-identifier my-db-shard-group \
  --db-cluster-identifier my-sv2-cluster \
  --max-acu 768

```

出力:

```
{
  "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
  "DBShardGroupIdentifier": "my-db-shard-group",
  "DBClusterIdentifier": "my-sv2-cluster",
  "MaxACU": 768.0,
  "ComputeRedundancy": 0,
  "Status": "creating",
  "PubliclyAccessible": false,
  "Endpoint": "my-sv2-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"
}
```

詳細については、「Amazon [Aurora ユーザーガイド](#)」の「[Aurora Serverless v2 の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDbShardGroup](#)」の「」を参照してください。

create-db-snapshot

次の例は、create-db-snapshot を使用する方法を説明しています。

AWS CLI

DB スナップショットを作成するには

次の create-db-snapshot の例は、DB スナップショットを作成します。

```
aws rds create-db-snapshot \  
  --db-instance-identifier database-mysql \  
  --db-snapshot-identifier mydbsnapshot
```

出力:

```
{
  "DBSnapshot": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBInstanceIdentifier": "database-mysql",
    "Engine": "mysql",
    "AllocatedStorage": 100,
  }
}
```

```
"Status": "creating",
"Port": 3306,
"AvailabilityZone": "us-east-1b",
"VpcId": "vpc-6594f31c",
"InstanceCreateTime": "2019-04-30T15:45:53.663Z",
"MasterUsername": "admin",
"EngineVersion": "5.6.40",
"LicenseModel": "general-public-license",
"SnapshotType": "manual",
"Iops": 1000,
"OptionGroupName": "default:mysql-5-6",
"PercentProgress": 0,
"StorageType": "io1",
"Encrypted": true,
"KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",
"IAMDatabaseAuthenticationEnabled": false,
"ProcessorFeatures": [],
"DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
}
}
```

詳細については、「Amazon RDSユーザーガイド」の「[DB スナップショットの作成](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[CreateDBSnapshot](#)」を参照してください。AWS CLI

create-db-subnet-group

次のコード例は、create-db-subnet-group を使用する方法を示しています。

AWS CLI

DB サブネットグループを作成するには

次のcreate-db-subnet-group例では、既存のサブネットmysubnetgroupを使用して という名前の DB サブネットグループを作成します。

```
aws rds create-db-subnet-group \
  --db-subnet-group-name mysubnetgroup \
  --db-subnet-group-description "test DB subnet group" \
```

```
--subnet-ids
```

```
'["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef"]'
```

出力:

```
{
  "DBSubnetGroup": {
    "DBSubnetGroupName": "mysubnetgroup",
    "DBSubnetGroupDescription": "test DB subnet group",
    "VpcId": "vpc-0f08e7610a1b2c3d4",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        },
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2b"
        },
        "SubnetStatus": "Active"
      }
    ],
    "DBSubnetGroupArn": "arn:aws:rds:us-west-2:0123456789012:subgrp:mysubnetgroup"
  }
}
```

詳細については、[「Amazon ユーザーガイド」の「での DB インスタンスの作成VPCRDS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDbSubnetGroup](#)」の「」を参照してください。

create-event-subscription

次の例は、create-event-subscription を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを作成するには

次のcreate-event-subscription例では、現在の AWS アカウントで DB インスタンスのバックアップイベントとリカバリイベントのサブスクリプションを作成します。通知は、で指定された Amazon Simple Notification Service トピックに送信されます--sns-topic-arn。

```
aws rds create-event-subscription \  
  --subscription-name my-instance-events \  
  --source-type db-instance \  
  --event-categories '["backup","recovery"]' \  
  --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

出力:

```
{  
  "EventSubscription": {  
    "Status": "creating",  
    "CustSubscriptionId": "my-instance-events",  
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",  
    "EventCategoriesList": [  
      "backup",  
      "recovery"  
    ],  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",  
    "CustomerAwsId": "123456789012",  
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-events",  
    "SourceType": "db-instance",  
    "Enabled": true  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateEventSubscription](#)」の「」を参照してください。

create-global-cluster

次のコード例は、create-global-cluster を使用する方法を示しています。

AWS CLI

グローバル DB クラスターを作成するには

次のcreate-global-cluster例では、新しい Aurora My SQL互換のグローバル DB クラスターを作成します。

```
aws rds create-global-cluster \  
  --global-cluster-identifier myglobalcluster \  
  --engine aurora-mysql
```

出力:

```
{  
  "GlobalCluster": {  
    "GlobalClusterIdentifier": "myglobalcluster",  
    "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",  
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-  
cluster:myglobalcluster",  
    "Status": "available",  
    "Engine": "aurora-mysql",  
    "EngineVersion": "5.7.mysql_aurora.2.07.2",  
    "StorageEncrypted": false,  
    "DeletionProtection": false,  
    "GlobalClusterMembers": []  
  }  
}
```

詳細については、「Amazon [Aurora ユーザーガイド](#)」の「[Aurora グローバルデータベースの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateGlobalCluster](#)」の「」を参照してください。

create-option-group

次のコード例は、create-option-group を使用する方法を示しています。

AWS CLI

Amazon RDS オプショングループを作成するには

次の create-option-group コマンドは、Oracle Enterprise Edition バージョンの新しい Amazon RDS オプショングループを作成し 11.2、is named MyOptionGroup、説明を含めます。

```
aws rds create-option-group \  
  --option-group-name MyOptionGroup \  
  --engine-name oracle-ee \  
  --major-engine-version 11.2 \  
  --option-group-description "Oracle Database Manager Database Control"
```

出力:

```
{  
  "OptionGroup": {  
    "OptionGroupName": "myoptiongroup",  
    "OptionGroupDescription": "Oracle Database Manager Database Control",  
    "EngineName": "oracle-ee",  
    "MajorEngineVersion": "11.2",  
    "Options": [],  
    "AllowsVpcAndNonVpcInstanceMemberships": true,  
    "OptionGroupArn": "arn:aws:rds:us-west-2:123456789012:og:myoptiongroup"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateOptionGroup](#)」の「」を参照してください。

delete-blue-green-deployment

次のコード例は、delete-blue-green-deployment を使用する方法を示しています。

AWS CLI

例 1: RDS for MySQL DB インスタンスのグリーン環境でリソースを削除するには

次のdelete-blue-green-deployment例では、RDS for MySQL DB インスタンスのグリーン環境にあるリソースを削除します。

```
aws rds delete-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-v53303651eexfake \  
  --delete-target
```

出力:

```
{  
  "BlueGreenDeployment": {  
    "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",  
    "BlueGreenDeploymentName": "bgd-cli-test-instance",  
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",  
    "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-rkfbpe",  
    "SwitchoverDetails": [  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-rkfbpe",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-1",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-1-green-j382ha",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-2",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-2-green-ejv4ao",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-3",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-3-green-vlpz3t",
```

```

        "Status": "AVAILABLE"
      }
    ],
    "Tasks": [
      {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
      },
      {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
      },
      {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "COMPLETED"
      },
      {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "COMPLETED"
      }
    ],
    "Status": "DELETING",
    "CreateTime": "2022-02-25T21:18:51.183000+00:00",
    "DeleteTime": "2022-02-25T22:25:31.331000+00:00"
  }
}

```

詳細については、「Amazon RDSユーザーガイド」の [「ブルー/グリーンデプロイの削除」](#) を参照してください。

例 2: Aurora MySQL DB クラスターのグリーン環境でリソースを削除するには

次のdelete-blue-green-deployment例では、Aurora MySQL DB クラスターのグリーン環境にあるリソースを削除します。

```

aws rds delete-blue-green-deployment \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
  --delete-target

```

出力:

```

{
  "BlueGreenDeployment": {

```

```
    "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
    "BlueGreenDeploymentName": "my-blue-green-deployment",
    "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
    "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1-green-gpmaxf",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2-green-j2oajq",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-mkxies",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
        "Status": "AVAILABLE"
      },
      {
```

```
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwzlg",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
],
"Status": "DELETING",
"CreateTime": "2022-02-25T21:12:00.288000+00:00",
>DeleteTime": "2022-02-25T22:29:11.336000+00:00"
}
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[ブルー/グリーンデプロイの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBlueGreenDeployment](#)」の「」を参照してください。

delete-db-cluster-endpoint

次の例は、delete-db-cluster-endpoint を使用する方法を説明しています。

AWS CLI

カスタム DB クラスターエンドポイントを削除するには

次のdelete-db-cluster-endpoint例では、指定されたカスタム DB クラスターエンドポイントを削除します。

```
aws rds delete-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint
```

出力:

```
{  
  "DBClusterEndpointIdentifier": "mycustomendpoint",  
  "DBClusterIdentifier": "mydbcluster",  
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",  
  "Endpoint": "mycustomendpoint.cluster-custom-cnpeexample.us-  
east-1.rds.amazonaws.com",  
  "Status": "deleting",  
  "EndpointType": "CUSTOM",  
  "CustomEndpointType": "READER",  
  "StaticMembers": [  
    "dbinstance1",  
    "dbinstance2",  
    "dbinstance3"  
  ],  
  "ExcludedMembers": [],  
  "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
endpoint:mycustomendpoint"  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora Connection Management」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbClusterEndpoint](#)」の「」を参照してください。

delete-db-cluster-parameter-group

次の例は、delete-db-cluster-parameter-group を使用する方法を説明しています。

AWS CLI

DB クラスターパラメータグループを削除するには

次のdelete-db-cluster-parameter-group例では、指定された DB クラスターパラメータグループを削除します。

```
aws rds delete-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterparametergroup
```

このコマンドでは何も出力されません。

詳細については、「Amazon Aurora ユーザーガイド」の「[DB パラメータグループと DB クラスターパラメータグループの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbClusterParameterGroup](#)」の「」を参照してください。

delete-db-cluster-snapshot

次の例は、delete-db-cluster-snapshot を使用する方法を説明しています。

AWS CLI

DB クラスタースナップショットを削除するには

次のdelete-db-cluster-snapshot例では、指定された DB クラスタースナップショットを削除します。

```
aws rds delete-db-cluster-snapshot \  
  --db-cluster-snapshot-identifier mydbclustersnapshot
```

出力:

```
{  
  "DBClusterSnapshot": {  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1e"  
    ],  
    "DBClusterSnapshotIdentifier": "mydbclustersnapshot",
```

```
"DBClusterIdentifier": "mydbcluster",
"SnapshotCreateTime": "2019-06-18T21:21:00.469Z",
"Engine": "aurora-mysql",
"AllocatedStorage": 0,
"Status": "available",
"Port": 0,
"VpcId": "vpc-6594f31c",
"ClusterCreateTime": "2019-04-15T14:18:42.785Z",
"MasterUsername": "myadmin",
"EngineVersion": "5.7.mysql_aurora.2.04.2",
"LicenseModel": "aurora-mysql",
"SnapshotType": "manual",
"PercentProgress": 100,
"StorageEncrypted": true,
"KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",
"DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:mydbclustersnapshot",
"IAMDatabaseAuthenticationEnabled": false
}
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[スナップショットの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbClusterSnapshot](#)」の「」を参照してください。

delete-db-cluster

次の例は、delete-db-cluster を使用する方法を説明しています。

AWS CLI

例 1: DB クラスター内の DB インスタンスを削除するには

次のdelete-db-instance例では、DB クラスター内の最後の DB インスタンスを削除します。削除状態にない DB インスタンスが含まれている DB クラスターは削除できません。DB クラスターで DB インスタンスを削除するときに、最終的なスナップショットを作成することはできません。

```
aws rds delete-db-instance \  
  --db-instance-identifier database-3
```

出力:

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "database-3",
    "DBInstanceClass": "db.r4.large",
    "Engine": "aurora-postgresql",
    "DBInstanceStatus": "deleting",

    ...output omitted...

  }
}
```

詳細については、「Amazon Aurora [ユーザーガイド](#)」の「[Aurora DB クラスターの DB インスタンスの削除](#)」を参照してください。

例 2: DB クラスターを削除するには

次のdelete-db-cluster例では、という名前の DB クラスターを削除myclusterし、という名前の最終スナップショットを作成しますmycluster-final-snapshot。DB クラスターのステータスは、スナップショットの作成中に使用できません。削除の進行状況を確認するには、describe-db-clusters CLI コマンドを使用します。

```
aws rds delete-db-cluster \
  --db-cluster-identifier mycluster \
  --no-skip-final-snapshot \
  --final-db-snapshot-identifier mycluster-final-snapshot
```

出力:

```
{
  "DBCluster": {
    "AllocatedStorage": 20,
    "AvailabilityZones": [
      "eu-central-1b",
      "eu-central-1c",
      "eu-central-1a"
    ],
    "BackupRetentionPeriod": 7,
    "DBClusterIdentifier": "mycluster",
    "DBClusterParameterGroup": "default.aurora-postgresql110",
```



```

        "DBSubnetGroup": "default-vpc-aa11bb22",
        "Status": "available",

...output omitted...

    }
}

```

詳細については、「Amazon [Aurora ユーザーガイド](#)」の「[単一の DB インスタンスを持つ Aurora クラスター](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDbCluster](#)」の「」を参照してください。

delete-db-instance-automated-backup

次のコード例は、delete-db-instance-automated-backup を使用する方法を示しています。

AWS CLI

リージョンからレプリケートされた自動バックアップを削除するには

次のdelete-db-instance-automated-backup例では、指定された Amazon リソースネーム () で自動バックアップを削除しますARN。

```

aws rds delete-db-instance-automated-backup \
  --db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadusbrktni2bn4example"

```

出力:

```

{
  "DBInstanceAutomatedBackup": {
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
    "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
    "Region": "us-east-1",
    "DBInstanceIdentifier": "new-orcl-db",
    "RestoreWindow": {},
    "AllocatedStorage": 20,
    "Status": "deleting",
    "Port": 1521,
    "AvailabilityZone": "us-east-1b",
    "VpcId": "vpc-#####",
  }
}

```

```
"InstanceCreateTime": "2020-12-04T15:28:31Z",
"MasterUsername": "admin",
"Engine": "oracle-se2",
"EngineVersion": "12.1.0.2.v21",
"LicenseModel": "bring-your-own-license",
"OptionGroupName": "default:oracle-se2-12-1",
"Encrypted": false,
"StorageType": "gp2",
"IAMDatabaseAuthenticationEnabled": false,
"BackupRetentionPeriod": 7,
"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadabrktni2bn4example"
}
}
```

詳細については、「Amazon RDSユーザーガイド」の「[レプリケートされたバックアップの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDbInstanceAutomatedBackup](#)」の「」を参照してください。

delete-db-instance

次のコード例は、delete-db-instance を使用する方法を示しています。

AWS CLI

DB インスタンスを削除するには

次の delete-db-instance の例は、test-instance-final-snap という名前の最終 DB スナップショットを作成した後に、指定された DB インスタンスを削除します。

```
aws rds delete-db-instance \
  --db-instance-identifier test-instance \
  --final-db-snapshot-identifier test-instance-final-snap
```

出力:

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "test-instance",
    "DBInstanceStatus": "deleting",
    ...some output truncated...
  }
}
```

```
}  
}
```

- API 詳細については、「コマンドリファレンス」の「[DeleteDBInstance](#)」を参照してください。AWS CLI

delete-db-parameter-group

次のコード例は、delete-db-parameter-group を使用する方法を示しています。

AWS CLI

DB パラメータグループを削除するには

次の command の例は、DB パラメータグループを削除します。

```
aws rds delete-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup
```

このコマンドでは何も出力されません。

詳細については、「Amazon RDSユーザーガイド」の「[DB パラメータグループの使用](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[DeleteDBParameterグループ](#)」を参照してください。AWS CLI

delete-db-proxy-endpoint

次の例は、delete-db-proxy-endpoint を使用する方法を説明しています。

AWS CLI

RDS データベースの DB プロキシエンドポイントを削除するには

次のdelete-db-proxy-endpoint例では、ターゲットデータベースの DB プロキシエンドポイントを削除します。

```
aws rds delete-db-proxy-endpoint \  
  --db-proxy-endpoint-name proxyEP1
```

出力:

```
{
  "DBProxyEndpoint":
    {
      "DBProxyEndpointName": "proxyEP1",
      "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
      "DBProxyName": "proxyExample",
      "Status": "deleting",
      "VpcId": "vpc-1234567",
      "VpcSecurityGroupIds": [
        "sg-1234",
        "sg-5678"
      ],
      "VpcSubnetIds": [
        "subnetgroup1",
        "subnetgroup2"
      ],
      "Endpoint": "proxyEP1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
      "CreateDate": "2023-04-13T01:49:38.568000+00:00",
      "TargetRole": "READ_ONLY",
      "IsDefault": false
    }
}
```

詳細については、「Amazon RDSユーザーガイド」の「[プロキシエンドポイントの削除](#)」および「Amazon Aurora ユーザーガイド」の「[プロキシエンドポイントの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbProxyEndpoint](#)」の「」を参照してください。

delete-db-proxy

次の例は、delete-db-proxy を使用する方法を説明しています。

AWS CLI

RDS データベースの DB プロキシを削除するには

次のdelete-db-proxy例では、DB プロキシを削除します。

```
aws rds delete-db-proxy \  
--db-proxy-name proxyExample
```

出力:

```
{  
  "DBProxy":  
  {  
    "DBProxyName": "proxyExample",  
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-  
proxy:prx-0123a01b12345c0ab",  
    "Status": "deleting",  
    "EngineFamily": "PostgreSQL",  
    "VpcId": "vpc-1234567",  
    "VpcSecurityGroupIds": [  
      "sg-1234",  
      "sg-5678"  
    ],  
    "VpcSubnetIds": [  
      "subnetgroup1",  
      "subnetgroup2"  
    ],  
    "Auth": "[  
      {  
        "Description": "proxydescription`",  
        "AuthScheme": "SECRETS",  
        "SecretArn": "arn:aws:secretsmanager:us-  
west-2:123456789123:secret:proxysecret1-Abcd1e",  
        "IAMAuth": "DISABLED"  
      } ],  
    "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",  
    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-  
east-1.rds.amazonaws.com",  
    "RequireTLS": false,  
    "IdleClientTimeout": 1800,  
    "DebuggingLogging": false,  
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",  
    "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"  
  }  
}
```

詳細については、「Amazon RDSユーザーガイド」の[RDS「プロキシの削除」](#)と「Amazon Aurora ユーザーガイド」の[RDS「プロキシの削除」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteDbProxy](#)」の「」を参照してください。

delete-db-security-group

次のコード例は、delete-db-security-group を使用する方法を示しています。

AWS CLI

DB セキュリティグループを削除するには

次のdelete-db-security-group例では、という名前の DB セキュリティグループを削除しますmysecuritygroup。

```
aws rds delete-db-security-group \  
  --db-security-group-name mysecuritygroup
```

このコマンドでは何も出力されません。

詳細については、「Amazon RDSユーザーガイド」の[「DB セキュリティグループ \(EC2-Classic プラットフォーム\) の使用」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteDbSecurityGroup](#)」の「」を参照してください。

delete-db-shard-group

次の例は、delete-db-shard-group を使用する方法を説明しています。

AWS CLI

例 1: DB シャードグループの削除に失敗するには

次のdelete-db-shard-group例は、すべてのデータベースとスキーマを削除する前に DB シャードグループを削除しようとするると発生するエラーを示しています。

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifier limitless-test-shard-grp
```

出力:

```
An error occurred (InvalidDBShardGroupState) when calling the DeleteDBShardGroup operation: Unable to delete the DB shard group limitless-test-db-shard-group. Delete all of your Limitless Database databases and schemas, then try again.
```

例 2: DB シャードグループを正常に削除するには

次のdelete-db-shard-group例では、publicスキーマを含むすべてのデータベースとスキーマを削除した後、DB シャードグループを削除します。

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifier Limitless-test-shard-grp
```

出力:

```
{  
  "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",  
  "DBShardGroupIdentifier": "limitless-test-shard-grp",  
  "DBClusterIdentifier": "limitless-test-cluster",  
  "MaxACU": 768.0,  
  "ComputeRedundancy": 0,  
  "Status": "deleting",  
  "PubliclyAccessible": true,  
  "Endpoint": "limitless-test-cluster.limitless-cekyceexample.us-east-2.rds.amazonaws.com"  
}
```

詳細については、「Amazon [Aurora ユーザーガイド](#)」の「[Aurora DB クラスタと DB インスタンスの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbShardGroup](#)」の「」を参照してください。

delete-db-snapshot

次の例は、delete-db-snapshot を使用する方法を説明しています。

AWS CLI

DB スナップショットを削除するには

次のdelete-db-snapshot例では、指定された DB スナップショットを削除します。

```
aws rds delete-db-snapshot \  
  --db-snapshot-identifier mydbsnapshot
```

出力:

```
{  
  "DBSnapshot": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBInstanceIdentifier": "database-mysql",  
    "SnapshotCreateTime": "2019-06-18T22:08:40.702Z",  
    "Engine": "mysql",  
    "AllocatedStorage": 100,  
    "Status": "deleted",  
    "Port": 3306,  
    "AvailabilityZone": "us-east-1b",  
    "VpcId": "vpc-6594f31c",  
    "InstanceCreateTime": "2019-04-30T15:45:53.663Z",  
    "MasterUsername": "admin",  
    "EngineVersion": "5.6.40",  
    "LicenseModel": "general-public-license",  
    "SnapshotType": "manual",  
    "Iops": 1000,  
    "OptionGroupName": "default:mysql-5-6",  
    "PercentProgress": 100,  
    "StorageType": "io1",  
    "Encrypted": true,  
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE",  
    "DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mydbsnapshot",  
    "IAMDatabaseAuthenticationEnabled": false,  
    "ProcessorFeatures": [],  
    "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"  
  }  
}
```

詳細については、「Amazon RDS [ユーザーガイド](#)」の「[スナップショットの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteDbSnapshot](#)」の「」を参照してください。

delete-db-subnet-group

次の例は、delete-db-subnet-group を使用する方法を説明しています。

AWS CLI

DB サブネットグループを削除するには

次のdelete-db-subnet-group例では、 という名前の DB サブネットグループを削除しますmysubnetgroup。

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

このコマンドでは何も出力されません。

詳細については、「Amazon RDSユーザーガイド」の「[での DB インスタンスの使用VPC](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDbSubnetGroup](#)」の「」を参照してください。

delete-event-subscription

次の例は、delete-event-subscription を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを削除するには

次のdelete-event-subscription例では、指定されたイベントサブスクリプションを削除します。

```
aws rds delete-event-subscription --subscription-name my-instance-events
```

出力:

```
{
  "EventSubscription": {
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-events",
    "CustomerAwsId": "123456789012",
    "Enabled": false,
```

```
    "SourceIdsList": [
      "test-instance"
    ],
    "SourceType": "db-instance",
    "EventCategoriesList": [
      "backup",
      "recovery"
    ],
    "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
    "CustSubscriptionId": "my-instance-events",
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",
    "Status": "deleting"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEventSubscription](#)」の「」を参照してください。

delete-global-cluster

次の例は、delete-global-cluster を使用する方法を説明しています。

AWS CLI

グローバル DB クラスターを削除するには

次のdelete-global-cluster例では、Aurora My SQL互換のグローバル DB クラスターを削除します。出力には、削除しているクラスターが表示されますが、後続のdescribe-global-clustersコマンドにはその DB クラスターはリストされません。

```
aws rds delete-global-cluster \
  --global-cluster-identifier myglobalcluster
```

出力:

```
{
  "GlobalCluster": {
    "GlobalClusterIdentifier": "myglobalcluster",
    "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-cluster:myglobalcluster",
    "Status": "available",
```

```
    "Engine": "aurora-mysql",
    "EngineVersion": "5.7.mysql_aurora.2.07.2",
    "StorageEncrypted": false,
    "DeletionProtection": false,
    "GlobalClusterMembers": []
  }
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Aurora グローバルデータベースの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGlobalCluster](#)」の「」を参照してください。

delete-option-group

次のコード例は、delete-option-group を使用する方法を示しています。

AWS CLI

オプショングループを削除するには

次のdelete-option-group例では、指定されたオプショングループを削除します。

```
aws rds delete-option-group \  
  --option-group-name myoptiongroup
```

このコマンドでは何も出力されません。

詳細については、「Amazon RDSユーザーガイド」の[「オプショングループの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteOptionGroup](#)」の「」を参照してください。

deregister-db-proxy-targets

次のコード例は、deregister-db-proxy-targets を使用する方法を示しています。

AWS CLI

DB プロキシターゲットをデータベースターゲットグループから登録解除するには

次のderegister-db-proxy-targets例では、プロキシproxyExampleとそのターゲット間の関連付けを削除します。

```
aws rds deregister-db-proxy-targets \  
  --db-proxy-name proxyExample \  
  --db-instance-identifiers database-1
```

このコマンドでは何も出力されません。

詳細については、「Amazon RDSユーザーガイド」の[RDS「プロキシの削除」](#)と「Amazon Aurora ユーザーガイド」の[RDS「プロキシの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterDbProxyTargets](#)」の「」を参照してください。

describe-account-attributes

次の例は、describe-account-attributes を使用する方法を説明しています。

AWS CLI

アカウントの属性を記述するには

次のdescribe-account-attributes例では、現在の AWS アカウントの属性を取得します。

```
aws rds describe-account-attributes
```

出力:

```
{  
  "AccountQuotas": [  
    {  
      "Max": 40,  
      "Used": 4,  
      "AccountQuotaName": "DBInstances"  
    },  
    {  
      "Max": 40,  
      "Used": 0,  
      "AccountQuotaName": "ReservedDBInstances"  
    },  
    {
```

```
    "Max": 100000,
    "Used": 40,
    "AccountQuotaName": "AllocatedStorage"
  },
  {
    "Max": 25,
    "Used": 0,
    "AccountQuotaName": "DBSecurityGroups"
  },
  {
    "Max": 20,
    "Used": 0,
    "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
  },
  {
    "Max": 50,
    "Used": 1,
    "AccountQuotaName": "DBParameterGroups"
  },
  {
    "Max": 100,
    "Used": 3,
    "AccountQuotaName": "ManualSnapshots"
  },
  {
    "Max": 20,
    "Used": 0,
    "AccountQuotaName": "EventSubscriptions"
  },
  {
    "Max": 50,
    "Used": 1,
    "AccountQuotaName": "DBSubnetGroups"
  },
  {
    "Max": 20,
    "Used": 1,
    "AccountQuotaName": "OptionGroups"
  },
  {
    "Max": 20,
    "Used": 6,
    "AccountQuotaName": "SubnetsPerDBSubnetGroup"
  },
},
```

```
{
  "Max": 5,
  "Used": 0,
  "AccountQuotaName": "ReadReplicasPerMaster"
},
{
  "Max": 40,
  "Used": 1,
  "AccountQuotaName": "DBClusters"
},
{
  "Max": 50,
  "Used": 0,
  "AccountQuotaName": "DBClusterParameterGroups"
},
{
  "Max": 5,
  "Used": 0,
  "AccountQuotaName": "DBClusterRoles"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAccountAttributes](#)」の「」を参照してください。

describe-blue-green-deployments

次の例は、describe-blue-green-deployments を使用する方法を説明しています。

AWS CLI

例 1: 作成完了後の RDS DB インスタンスのブルー/グリーンデプロイを記述するには

次のdescribe-blue-green-deployment例では、作成が完了した後にブルー/グリーンデプロイの詳細を取得します。

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-v53303651eexfake
```

出力:

```
{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
      "BlueGreenDeploymentName": "bgd-cli-test-instance",
      "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
      "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-
rkfbpe",
      "SwitchoverDetails": [
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-green-rkfbpe",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-j382ha",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ejv4ao",
          "Status": "AVAILABLE"
        },
        {
          "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
          "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-vlpz3t",
          "Status": "AVAILABLE"
        }
      ],
      "Tasks": [
        {
          "Name": "CREATING_READ_REPLICA_OF_SOURCE",
          "Status": "COMPLETED"
        }
      ]
    }
  ]
}
```

```

        {
            "Name": "DB_ENGINE_VERSION_UPGRADE",
            "Status": "COMPLETED"
        },
        {
            "Name": "CONFIGURE_BACKUPS",
            "Status": "COMPLETED"
        },
        {
            "Name": "CREATING_TOPOLOGY_OF_SOURCE",
            "Status": "COMPLETED"
        }
    ],
    "Status": "AVAILABLE",
    "CreateTime": "2022-02-25T21:18:51.183000+00:00"
}
]
}

```

詳細については、「Amazon RDSユーザーガイド」の「[ブルー/グリーンデプロイの表示](#)」を参照してください。

例 2: Aurora MySQL DB クラスターのブルー/グリーンデプロイを記述するには

次のdescribe-blue-green-deployment例では、ブルー/グリーンデプロイの詳細を取得します。

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

出力:

```

{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
      "BlueGreenDeploymentName": "my-blue-green-deployment",
      "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
      "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-green-3rnukl",
      "SwitchoverDetails": [
        {

```



```
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-green-3rnukl",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-1-green-gpmaxf",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-2-green-j2oajq",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-cluster-3-green-mkxies",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-excluded-member-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-excluded-member-endpoint-green-4sqjrq",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-reader-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-reader-endpoint-green-gwzlg",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
```

```

        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
],
"Status": "AVAILABLE",
"CreateTime": "2022-02-25T21:12:00.288000+00:00"
}
]
}

```

詳細については、「[Amazon Aurora ユーザーガイド](#)」の「[ブルー/グリーンデプロイの表示](#)」を参照してください。

例 3: スイッチオーバー後の Aurora MySQL クラスターのブルー/グリーンデプロイを記述するには

次のdescribe-blue-green-deployment例では、グリーン環境を本番環境に昇格させた後に、ブルー/グリーンデプロイの詳細を取得します。

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

出力:

```

{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
      "BlueGreenDeploymentName": "my-blue-green-deployment",
      "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-cluster-old1",
    }
  ]
}

```

```
    "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
        "Status": "SWITCHOVER_COMPLETED"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
        "Status": "SWITCHOVER_COMPLETED"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
        "Status": "SWITCHOVER_COMPLETED"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
        "Status": "SWITCHOVER_COMPLETED"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
        "Status": "SWITCHOVER_COMPLETED"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
        "Status": "SWITCHOVER_COMPLETED"
      }
    ]
  }
}
```

```
    }
  ],
  "Tasks": [
    {
      "Name": "CREATING_READ_REPLICA_OF_SOURCE",
      "Status": "COMPLETED"
    },
    {
      "Name": "DB_ENGINE_VERSION_UPGRADE",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_CUSTOM_ENDPOINTS",
      "Status": "COMPLETED"
    }
  ],
  "Status": "SWITCHOVER_COMPLETED",
  "CreateTime": "2022-02-25T22:38:49.522000+00:00"
}
]
```

詳細については、「[Amazon Aurora ユーザーガイド](#)」の「[ブルー/グリーンデプロイの表示](#)」を参照してください。

例 4: ブルーとグリーンを組み合わせたデプロイを記述するには

次のdescribe-blue-green-deployment例では、ブルーとグリーンを組み合わせたデプロイの詳細を取得します。

```
aws rds describe-blue-green-deployments
```

出力:

```
{
  "BlueGreenDeployments": [
    {
      "BlueGreenDeploymentIdentifier": "bgd-wi89nwzgfakelccs",
      "BlueGreenDeploymentName": "my-blue-green-deployment",
```

```
    "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
    "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-1-green-gpmaxf",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-2-green-j2oajq",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-
aurora-mysql-cluster-3-green-mkxies",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
```

```

        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwzlg",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATE_CUSTOM_ENDPOINTS",
        "Status": "COMPLETED"
    }
],
"Status": "AVAILABLE",
"CreateTime": "2022-02-25T21:12:00.288000+00:00"
},
{
    "BlueGreenDeploymentIdentifier": "bgd-v5330365fake1eex",
    "BlueGreenDeploymentName": "bgd-cli-test-instance",
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-old1",
    "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "SwitchoverDetails": [
        {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-old1",
            "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance",
            "Status": "SWITCHOVER_COMPLETED"
        },
        {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-old1",
            "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1",

```

```
        "Status": "SWITCHOVER_COMPLETED"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
        "Status": "SWITCHOVER_COMPLETED"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-old1",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
        "Status": "SWITCHOVER_COMPLETED"
      }
    ],
    "Tasks": [
      {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
      },
      {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
      },
      {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "COMPLETED"
      },
      {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "COMPLETED"
      }
    ],
    "Status": "SWITCHOVER_COMPLETED",
    "CreateTime": "2022-02-25T22:33:22.225000+00:00"
  }
]
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[ブルー/グリーンデプロイの表示](#)」および「[Amazon Aurora ユーザーガイド](#)」の「[ブルー/グリーンデプロイの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeBlueGreenDeployments](#)」の「」を参照してください。

describe-certificates

次の例は、describe-certificates を使用する方法を説明しています。

AWS CLI

証明書を記述するには

次のdescribe-certificates例では、ユーザーのデフォルトリージョンに関連付けられた証明書の詳細を取得します。

```
aws rds describe-certificates
```

出力:

```
{
  "Certificates": [
    {
      "CertificateIdentifier": "rds-ca-ecc384-g1",
      "CertificateType": "CA",
      "Thumbprint": "2ee3dcc06e50192559b13929e73484354f23387d",
      "ValidFrom": "2021-05-24T22:06:59+00:00",
      "ValidTill": "2121-05-24T23:06:59+00:00",
      "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-ecc384-g1",
      "CustomerOverride": false
    },
    {
      "CertificateIdentifier": "rds-ca-rsa4096-g1",
      "CertificateType": "CA",
      "Thumbprint": "19da4f2af579a8ae1f6a0fa77aa5befd874b4cab",
      "ValidFrom": "2021-05-24T22:03:20+00:00",
      "ValidTill": "2121-05-24T23:03:20+00:00",
      "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa4096-g1",
      "CustomerOverride": false
    },
  ]
}
```



```
    "CertificateIdentifier": "rds-ca-rsa2048-g1",
    "CertificateType": "CA",
    "Thumbprint": "7c40cb42714b6fdb2b296f9bbd0e8bb364436a76",
    "ValidFrom": "2021-05-24T21:59:00+00:00",
    "ValidTill": "2061-05-24T22:59:00+00:00",
    "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa2048-g1",
    "CustomerOverride": true,
    "CustomerOverrideValidTill": "2061-05-24T22:59:00+00:00"
  },
  {
    "CertificateIdentifier": "rds-ca-2019",
    "CertificateType": "CA",
    "Thumbprint": "d40ddb29e3750dffa671c3140bbf5f478d1c8096",
    "ValidFrom": "2019-08-22T17:08:50+00:00",
    "ValidTill": "2024-08-22T17:08:50+00:00",
    "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-2019",
    "CustomerOverride": false
  }
],
"DefaultCertificateForNewLaunches": "rds-ca-rsa2048-g1"
}
```

詳細については、「Amazon RDSユーザーガイド」の[SSL/TLS「/」を使用して DB インスタンスへの接続を暗号化する](#) および「Amazon Aurora ユーザーガイド」の[SSL「/TLS」を使用して DB クラスターへの接続を暗号化する](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeCertificates](#)」の「」を参照してください。

describe-db-cluster-backtracks

次のコード例は、describe-db-cluster-backtracks を使用する方法を示しています。

AWS CLI

DB クラスターのバックトラックを記述するには

次のdescribe-db-cluster-backtracks例では、指定された DB クラスターの詳細を取得します。

```
aws rds describe-db-cluster-backtracks \
  --db-cluster-identifier mydbcluster
```

出力:

```
{
  "DBClusterBacktracks": [
    {
      "DBClusterIdentifier": "mydbcluster",
      "BacktrackIdentifier": "2f5f5294-0dd2-44c9-9f50-EXAMPLE",
      "BacktrackTo": "2021-02-12T04:59:22Z",
      "BacktrackedFrom": "2021-02-12T14:37:31.640Z",
      "BacktrackRequestCreationTime": "2021-02-12T14:36:18.819Z",
      "Status": "COMPLETED"
    },
    {
      "DBClusterIdentifier": "mydbcluster",
      "BacktrackIdentifier": "3c7a6421-af2a-4ea3-ae95-EXAMPLE",
      "BacktrackTo": "2021-02-11T22:53:46Z",
      "BacktrackedFrom": "2021-02-12T00:09:27.006Z",
      "BacktrackRequestCreationTime": "2021-02-12T00:07:53.487Z",
      "Status": "COMPLETED"
    }
  ]
}
```

詳細については、「Amazon [Aurora ユーザーガイド](#)」の「[Aurora DB クラスターのバックトラック](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeDbClusterBacktracks](#)」の「」を参照してください。

describe-db-cluster-endpoints

次のコード例は、describe-db-cluster-endpoints を使用方法を示しています。

AWS CLI

例 1: DB クラスターエンドポイントを記述するには

次のdescribe-db-cluster-endpoints例では、DB クラスターエンドポイントの詳細を取得します。最も一般的な種類の Aurora クラスターには 2 つのエンドポイントがあります。1 つのエンドポイントにはタイプ `WRITER` があります。このエンドポイントは、すべての SQL ステートメントに使用できます。もう 1 つのエンドポイントにはタイプ `READER` があります。このエンドポイントは、`SELECT` およびその他の読み取り専用 SQL ステートメントでのみ使用できます。

aws rds describe-db-cluster-endpoints

出力:

```
{
  "DBClusterEndpoints": [
    {
      "DBClusterIdentifier": "my-database-1",
      "Endpoint": "my-database-1.cluster-cnpxample.us-east-1.rds.amazonaws.com",
      "Status": "creating",
      "EndpointType": "WRITER"
    },
    {
      "DBClusterIdentifier": "my-database-1",
      "Endpoint": "my-database-1.cluster-ro-cnpxample.us-east-1.rds.amazonaws.com",
      "Status": "creating",
      "EndpointType": "READER"
    },
    {
      "DBClusterIdentifier": "mydbcluster",
      "Endpoint": "mydbcluster.cluster-cnpxample.us-east-1.rds.amazonaws.com",
      "Status": "available",
      "EndpointType": "WRITER"
    },
    {
      "DBClusterIdentifier": "mydbcluster",
      "Endpoint": "mydbcluster.cluster-ro-cnpxample.us-east-1.rds.amazonaws.com",
      "Status": "available",
      "EndpointType": "READER"
    }
  ]
}
```

例 2: 単一の DB クラスターの DB クラスターエンドポイントを記述するには

次のdescribe-db-cluster-endpoints例では、単一の指定された DB クラスターの DB クラスターエンドポイントの詳細を取得します。Aurora Serverless クラスターには、タイプのエンドポイントが 1 つだけありますWRITER。

```
aws rds describe-db-cluster-endpoints \  
--db-cluster-identifier serverless-cluster
```

出力:

```
{  
  "DBClusterEndpoints": [  
    {  
      "Status": "available",  
      "Endpoint": "serverless-cluster.cluster-cnpxample.us-  
east-1.rds.amazonaws.com",  
      "DBClusterIdentifier": "serverless-cluster",  
      "EndpointType": "WRITER"  
    }  
  ]  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora Connection Management」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbClusterEndpoints](#)」の「」を参照してください。

describe-db-cluster-parameter-groups

次のコード例は、describe-db-cluster-parameter-groups を使用する方法を示しています。

AWS CLI

DB クラスターパラメータグループを記述するには

次のdescribe-db-cluster-parameter-groups例では、DB クラスターパラメータグループの詳細を取得します。

```
aws rds describe-db-cluster-parameter-groups
```

出力:

```
{  
  "DBClusterParameterGroups": [  

```

```
{
  "DBClusterParameterGroupName": "default.aurora-mysql5.7",
  "DBParameterGroupFamily": "aurora-mysql5.7",
  "Description": "Default cluster parameter group for aurora-mysql5.7",
  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora-mysql5.7"
},
{
  "DBClusterParameterGroupName": "default.aurora-postgresql9.6",
  "DBParameterGroupFamily": "aurora-postgresql9.6",
  "Description": "Default cluster parameter group for aurora-
postgresql9.6",
  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora-postgresql9.6"
},
{
  "DBClusterParameterGroupName": "default.aurora5.6",
  "DBParameterGroupFamily": "aurora5.6",
  "Description": "Default cluster parameter group for aurora5.6",
  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:default.aurora5.6"
},
{
  "DBClusterParameterGroupName": "mydbclusterpg",
  "DBParameterGroupFamily": "aurora-mysql5.7",
  "Description": "My DB cluster parameter group",
  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpg"
},
{
  "DBClusterParameterGroupName": "mydbclusterpgcopy",
  "DBParameterGroupFamily": "aurora-mysql5.7",
  "Description": "Copy of mydbclusterpg parameter group",
  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:mydbclusterpgcopy"
}
]
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB パラメータグループと DB クラスターパラメータグループの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDbClusterParameterGroups](#)」の「」を参照してください。

describe-db-cluster-parameters

次の例は、describe-db-cluster-parameters を使用する方法を説明しています。

AWS CLI

例 1: DB クラスターパラメータグループのパラメータを記述するには

次のdescribe-db-cluster-parameters例では、DB クラスターパラメータグループのパラメータに関する詳細を取得します。

```
aws rds describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name mydbclusterpg
```

出力:

```
{  
  "Parameters": [  
    {  
      "ParameterName": "allow-suspicious-udfs",  
      "Description": "Controls whether user-defined functions that have only  
an xxx symbol for the main function can be loaded",  
      "Source": "engine-default",  
      "ApplyType": "static",  
      "DataType": "boolean",  
      "AllowedValues": "0,1",  
      "IsModifiable": false,  
      "ApplyMethod": "pending-reboot",  
      "SupportedEngineModes": [  
        "provisioned"  
      ]  
    },  
    {  
      "ParameterName": "aurora_lab_mode",  
      "ParameterValue": "0",  
      "Description": "Enables new features in the Aurora engine.",  
      "Source": "engine-default",  
      "ApplyType": "static",  
      "DataType": "boolean",  
      "AllowedValues": "0,1",  
      "IsModifiable": true,  
      "ApplyMethod": "pending-reboot",  
      "SupportedEngineModes": [  
        "provisioned"  
      ]  
    }  
  ]  
}
```

```

        "provisioned"
      ]
    },
    ...some output truncated...
  ]
}

```

例 2: DB クラスターパラメータグループのパラメータ名のみを一覧表示するには

次のdescribe-db-cluster-parameters例では、DB クラスターパラメータグループ内のパラメータの名前のみを取得します。

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].[ParameterName:ParameterName]'

```

出力:

```

[
  {
    "ParameterName": "allow-suspicious-udfs"
  },
  {
    "ParameterName": "aurora_binlog_read_buffer_size"
  },
  {
    "ParameterName": "aurora_binlog_replication_max_yield_seconds"
  },
  {
    "ParameterName": "aurora_binlog_use_large_read_buffer"
  },
  {
    "ParameterName": "aurora_lab_mode"
  },
  ...some output truncated...
]

```

例 3: DB クラスターパラメータグループの変更可能なパラメータのみを記述するには

次のdescribe-db-cluster-parameters例では、DB クラスターパラメータグループで変更できるパラメータの名前のみを取得します。

```
aws rds describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \  
  --query 'Parameters[].[ParameterName:ParameterName,IsModifiable:IsModifiable] |  
  [?IsModifiable == `true`]'
```

出力:

```
[  
  {  
    "ParameterName": "aurora_binlog_read_buffer_size",  
    "IsModifiable": true  
  },  
  {  
    "ParameterName": "aurora_binlog_replication_max_yield_seconds",  
    "IsModifiable": true  
  },  
  {  
    "ParameterName": "aurora_binlog_use_large_read_buffer",  
    "IsModifiable": true  
  },  
  {  
    "ParameterName": "aurora_lab_mode",  
    "IsModifiable": true  
  },  
  ...some output truncated...  
]
```

例 4: DB クラスターパラメータグループ内の変更可能なブールパラメータのみを記述するには

次のdescribe-db-cluster-parameters例では、DB クラスターパラメータグループで変更でき、ブール型データ型を持つパラメータの名前のみを取得します。

```
aws rds describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \  
  --query 'Parameters[.].  
  {ParameterName:ParameterName,DataType:DataType,IsModifiable:IsModifiable} | [?  
  DataType == `boolean`] | [?IsModifiable == `true`]'
```


出力:

```
[
  {
    "DataType": "boolean",
    "ParameterName": "aurora_binlog_use_large_read_buffer",
    "IsModifiable": true
  },
  {
    "DataType": "boolean",
    "ParameterName": "aurora_lab_mode",
    "IsModifiable": true
  },
  {
    "DataType": "boolean",
    "ParameterName": "autocommit",
    "IsModifiable": true
  },
  {
    "DataType": "boolean",
    "ParameterName": "automatic_sp_privileges",
    "IsModifiable": true
  },
  ...some output truncated...
]
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB パラメータグループと DB クラスターパラメータグループの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbClusterParameters](#)」の「」を参照してください。

describe-db-cluster-snapshot-attributes

次の例は、describe-db-cluster-snapshot-attributes を使用する方法を説明しています。

AWS CLI

DB クラスタースナップショットの属性名と値を記述するには

次のdescribe-db-cluster-snapshot-attributes例では、指定された DB クラスタースナップショットの属性名と値の詳細を取得します。

```
aws rds describe-db-cluster-snapshot-attributes \  
--db-cluster-snapshot-identifier myclustersnapshot
```

出力:

```
{  
  "DBClusterSnapshotAttributesResult": {  
    "DBClusterSnapshotIdentifier": "myclustersnapshot",  
    "DBClusterSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "123456789012"  
        ]  
      }  
    ]  
  }  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB クラスタースナップショットの共有](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDbClusterSnapshotAttributes](#)」の「」を参照してください。

describe-db-cluster-snapshots

次の例は、describe-db-cluster-snapshots を使用する方法を説明しています。

AWS CLI

DB クラスターの DB クラスタースナップショットを記述するには

次のdescribe-db-cluster-snapshots例では、指定された DB クラスターの DB クラスタースナップショットの詳細を取得します。

```
aws rds describe-db-cluster-snapshots \  
--db-cluster-identifier mydbcluster
```

出力:

```
{
```

```
"DBClusterSnapshots": [
  {
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1e"
    ],
    "DBClusterSnapshotIdentifier": "myclustersnapshotcopy",
    "DBClusterIdentifier": "mydbcluster",
    "SnapshotCreateTime": "2019-06-04T09:16:42.649Z",
    "Engine": "aurora-mysql",
    "AllocatedStorage": 0,
    "Status": "available",
    "Port": 0,
    "VpcId": "vpc-6594f31c",
    "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
    "MasterUsername": "myadmin",
    "EngineVersion": "5.7.mysql_aurora.2.04.2",
    "LicenseModel": "aurora-mysql",
    "SnapshotType": "manual",
    "PercentProgress": 100,
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:814387698303:cluster-
snapshot:myclustersnapshotcopy",
    "IAMDatabaseAuthenticationEnabled": false
  },
  {
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1e"
    ],
    "DBClusterSnapshotIdentifier": "rds:mydbcluster-2019-06-20-09-16",
    "DBClusterIdentifier": "mydbcluster",
    "SnapshotCreateTime": "2019-06-20T09:16:26.569Z",
    "Engine": "aurora-mysql",
    "AllocatedStorage": 0,
    "Status": "available",
    "Port": 0,
    "VpcId": "vpc-6594f31c",
    "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
    "MasterUsername": "myadmin",
```

```
    "EngineVersion": "5.7.mysql_aurora.2.04.2",
    "LicenseModel": "aurora-mysql",
    "SnapshotType": "automated",
    "PercentProgress": 100,
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
    "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:123456789012:cluster-
snapshot:rds:mydbcluster-2019-06-20-09-16",
    "IAMDatabaseAuthenticationEnabled": false
  }
]
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB クラスタースナップショットの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbClusterSnapshots](#)」の「」を参照してください。

describe-db-clusters

次のコード例は、describe-db-clusters を使用する方法を示しています。

AWS CLI

例 1: DB クラスターを記述するには

次のdescribe-db-clusters例では、指定された DB クラスターの詳細を取得します。

```
aws rds describe-db-clusters \
  --db-cluster-identifier mydbcluster
```

出力:

```
{
  "DBClusters": [
    {
      "AllocatedStorage": 1,
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1e"
      ]
    }
  ]
}
```

```
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "mydbcluster",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterParameterGroup": "default.aurora-mysql5.7",
    "DBSubnetGroup": "default",
    "Status": "available",
    "EarliestRestorableTime": "2019-06-19T09:16:28.210Z",
    "Endpoint": "mydbcluster.cluster-cnpxample.us-
east-1.rds.amazonaws.com",
    "ReaderEndpoint": "mydbcluster.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
    "MultiAZ": true,
    "Engine": "aurora-mysql",
    "EngineVersion": "5.7.mysql_aurora.2.04.2",
    "LatestRestorableTime": "2019-06-20T22:38:14.908Z",
    "Port": 3306,
    "MasterUsername": "myadmin",
    "PreferredBackupWindow": "09:09-09:39",
    "PreferredMaintenanceWindow": "sat:04:09-sat:04:39",
    "ReadReplicaIdentifiers": [],
    "DBClusterMembers": [
      {
        "DBInstanceIdentifier": "dbinstance3",
        "IsClusterWriter": false,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      },
      {
        "DBInstanceIdentifier": "dbinstance1",
        "IsClusterWriter": false,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      },
      {
        "DBInstanceIdentifier": "dbinstance2",
        "IsClusterWriter": false,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      },
      {
        "DBInstanceIdentifier": "mydbcluster",
        "IsClusterWriter": false,
        "DBClusterParameterGroupStatus": "in-sync",
```

```
        "PromotionTier": 1
      },
      {
        "DBInstanceIdentifier": "mydbcluster-us-east-1b",
        "IsClusterWriter": false,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      },
      {
        "DBInstanceIdentifier": "mydbcluster",
        "IsClusterWriter": true,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      }
    ],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
        "Status": "active"
      }
    ],
    "HostedZoneId": "Z2R2ITUGPM61AM",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
    "DbClusterResourceId": "cluster-AKIAIOSFODNN7EXAMPLE",
    "DBClusterArn": "arn:aws:rds:us-
east-1:123456789012:cluster:mydbcluster",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false
  }
]
}
```

例 2: すべての DB クラスターの特定の属性を一覧表示するには

次のdescribe-db-clusters例ではDBClusterIdentifier、現在の AWS リージョン内のすべての DB クラスターの、Endpoint、および ReaderEndpoint 属性のみを取得します。

```
aws rds describe-db-clusters \  
  --query 'DBClusters[.]  
{DBClusterIdentifier:DBClusterIdentifier,Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint}'
```

出力:

```
[  
  {  
    "Endpoint": "cluster-57-2020-05-01-2270.cluster-cnpxample.us-  
east-1.rds.amazonaws.com",  
    "ReaderEndpoint": "cluster-57-2020-05-01-2270.cluster-ro-cnpxample.us-  
east-1.rds.amazonaws.com",  
    "DBClusterIdentifier": "cluster-57-2020-05-01-2270"  
  },  
  {  
    "Endpoint": "cluster-57-2020-05-01-4615.cluster-cnpxample.us-  
east-1.rds.amazonaws.com",  
    "ReaderEndpoint": "cluster-57-2020-05-01-4615.cluster-ro-cnpxample.us-  
east-1.rds.amazonaws.com",  
    "DBClusterIdentifier": "cluster-57-2020-05-01-4615"  
  },  
  {  
    "Endpoint": "pg2-cluster.cluster-cnpxample.us-east-1.rds.amazonaws.com",  
    "ReaderEndpoint": "pg2-cluster.cluster-ro-cnpxample.us-  
east-1.rds.amazonaws.com",  
    "DBClusterIdentifier": "pg2-cluster"  
  },  
  ...output omitted...  
]
```

例 3: 特定の属性を持つ DB クラスターを一覧表示するには

次のdescribe-db-clusters例では、DB エンジンを使用する aurora-postgresql DB クラスターの DBClusterIdentifierおよび Engine 属性のみを取得します。

```
aws rds describe-db-clusters \  
  --query 'DBClusters[.]{DBClusterIdentifier:DBClusterIdentifier,Engine:Engine} |  
[?Engine == `aurora-postgresql`]'
```

出力:

```
[
  {
    "Engine": "aurora-postgresql",
    "DBClusterIdentifier": "pg2-cluster"
  }
]
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスター」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbClusters](#)」の「」を参照してください。

describe-db-engine-versions

次のコード例は、describe-db-engine-versions を使用する方法を示しています。

AWS CLI

MySQL DB エンジンの DB エンジンバージョンを記述するには

次の describe-db-engine-versions の例は、指定された DB エンジンの各 DB エンジンバージョンに関する詳細が表示されます。

```
aws rds describe-db-engine-versions \
  --engine mysql
```

出力:

```
{
  "DBEngineVersions": [
    {
      "Engine": "mysql",
      "EngineVersion": "5.5.46",
      "DBParameterGroupFamily": "mysql5.5",
      "DBEngineDescription": "MySQL Community Edition",
      "DBEngineVersionDescription": "MySQL 5.5.46",
      "ValidUpgradeTarget": [
        {
          "Engine": "mysql",
          "EngineVersion": "5.5.53",
          "Description": "MySQL 5.5.53",

```



```
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
    },
    {
        "Engine": "mysql",
        "EngineVersion": "5.5.54",
        "Description": "MySQL 5.5.54",
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
    },
    {
        "Engine": "mysql",
        "EngineVersion": "5.5.57",
        "Description": "MySQL 5.5.57",
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
    },
    ...some output truncated...
]
}
```

詳細については、[Amazon Relational Database Service \(Amazon RDS\) とは](#) を参照してください。 RDS

- API 詳細については、[「コマンドリファレンス」の「DescribeDBEngineバージョン」](#) を参照してください。 AWS CLI

describe-db-instance-automated-backups

次の例は、describe-db-instance-automated-backups を使用する方法を説明しています。

AWS CLI

DB インスタンスの自動バックアップを記述するには

次のdescribe-db-instance-automated-backups例では、指定された DB インスタンスの自動バックアップの詳細を表示します。詳細には、他の AWS リージョンでのレプリケートされた自動バックアップが含まれます。

```
aws rds describe-db-instance-automated-backups \
  --db-instance-identifier new-orcl-db
```

出力:

```
{
  "DBInstanceAutomatedBackups": [
    {
      "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
      "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
      "Region": "us-east-1",
      "DBInstanceIdentifier": "new-orcl-db",
      "RestoreWindow": {
        "EarliestTime": "2020-12-07T21:05:20.939Z",
        "LatestTime": "2020-12-07T21:05:20.939Z"
      },
      "AllocatedStorage": 20,
      "Status": "replicating",
      "Port": 1521,
      "InstanceCreateTime": "2020-12-04T15:28:31Z",
      "MasterUsername": "admin",
      "Engine": "oracle-se2",
      "EngineVersion": "12.1.0.2.v21",
      "LicenseModel": "bring-your-own-license",
      "OptionGroupName": "default:oracle-se2-12-1",
      "Encrypted": false,
      "StorageType": "gp2",
      "IAMDatabaseAuthenticationEnabled": false,
      "BackupRetentionPeriod": 14,
      "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfg5rv7replzadabrktni2bn4example"
    }
  ]
}
```

詳細については、「Amazon RDSユーザーガイド」の「[レプリケートされたバックアップの検出情報](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDbInstanceAutomatedBackups](#)」の「」を参照してください。

describe-db-instances

次の例は、describe-db-instances を使用する方法を説明しています。

AWS CLI

DB インスタンスを記述するには

次の `describe-db-instances` の例は、指定された DB インスタンスの詳細情報を取得します。

```
aws rds describe-db-instances \  
  --db-instance-identifier mydbinstancecf
```

出力:

```
{  
  "DBInstances": [  
    {  
      "DBInstanceIdentifier": "mydbinstancecf",  
      "DBInstanceClass": "db.t3.small",  
      "Engine": "mysql",  
      "DBInstanceStatus": "available",  
      "MasterUsername": "masterawsuser",  
      "Endpoint": {  
        "Address": "mydbinstancecf.abcxample.us-east-1.rds.amazonaws.com",  
        "Port": 3306,  
        "HostedZoneId": "Z2R2ITUGPM61AM"  
      },  
      "...some output truncated..."  
    },  
  ]  
}
```

- API 詳細については、「コマンドリファレンス」の「[DescribeDBInstances](#)」を参照してください。AWS CLI

describe-db-log-files

次の例は、`describe-db-log-files` を使用する方法を説明しています。

AWS CLI

DB インスタンスのログファイルを記述するには

次のdescribe-db-log-files例では、指定された DB インスタンスのログファイルの詳細を取得します。

```
aws rds describe-db-log-files -\
  -db-instance-identifier test-instance
```

出力:

```
{
  "DescribeDBLogFiles": [
    {
      "Size": 0,
      "LastWritten": 1533060000000,
      "LogFileName": "error/mysql-error-running.log"
    },
    {
      "Size": 2683,
      "LastWritten": 1532994300000,
      "LogFileName": "error/mysql-error-running.log.0"
    },
    {
      "Size": 107,
      "LastWritten": 1533057300000,
      "LogFileName": "error/mysql-error-running.log.18"
    },
    {
      "Size": 13105,
      "LastWritten": 1532991000000,
      "LogFileName": "error/mysql-error-running.log.23"
    },
    {
      "Size": 0,
      "LastWritten": 1533061200000,
      "LogFileName": "error/mysql-error.log"
    },
    {
      "Size": 3519,
      "LastWritten": 1532989252000,
      "LogFileName": "mysqlUpgrade"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbLogFiles](#)」の「」を参照してください。

describe-db-parameter-groups

次の例は、describe-db-parameter-groups を使用する方法を説明しています。

AWS CLI

DB パラメータグループを記述するには

次の describe-db-parameter-groups の例では、DB パラメータグループに関する詳細を取得します。

```
aws rds describe-db-parameter-groups
```

出力:

```
{
  "DBParameterGroups": [
    {
      "DBParameterGroupName": "default.aurora-mysql5.7",
      "DBParameterGroupFamily": "aurora-mysql5.7",
      "Description": "Default parameter group for aurora-mysql5.7",
      "DBParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:pg:default.aurora-mysql5.7"
    },
    {
      "DBParameterGroupName": "default.aurora-postgresql9.6",
      "DBParameterGroupFamily": "aurora-postgresql9.6",
      "Description": "Default parameter group for aurora-postgresql9.6",
      "DBParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:pg:default.aurora-postgresql9.6"
    },
    {
      "DBParameterGroupName": "default.aurora5.6",
      "DBParameterGroupFamily": "aurora5.6",
      "Description": "Default parameter group for aurora5.6",
      "DBParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:pg:default.aurora5.6"
    },
    {
```

```
    "DBParameterGroupName": "default.mariadb10.1",
    "DBParameterGroupFamily": "mariadb10.1",
    "Description": "Default parameter group for mariadb10.1",
    "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
  },
  ...some output truncated...
]
}
```

詳細については、「Amazon RDSユーザーガイド」の [「DB パラメータグループの操作」](#) を参照してください。

- API 詳細については、「コマンドリファレンス」の [「DescribeDBParameterグループ」](#) を参照してください。AWS CLI

describe-db-parameters

次の例は、describe-db-parameters を使用する方法を説明しています。

AWS CLI

DB パラメータグループのパラメータを記述するには

次の describe-db-parameters の例では、指定された DB パラメータグループに関する詳細を取得します。

```
aws rds describe-db-parameters \
  --db-parameter-group-name mydbpg
```

出力:

```
{
  "Parameters": [
    {
      "ParameterName": "allow-suspicious-udfs",
      "Description": "Controls whether user-defined functions that have only
an xxx symbol for the main function can be loaded",
      "Source": "engine-default",
      "ApplyType": "static",
      "DataType": "boolean",
      "AllowedValues": "0,1",
      "IsModifiable": false,
```

```
        "ApplyMethod": "pending-reboot"
    },
    {
        "ParameterName": "auto_generate_certs",
        "Description": "Controls whether the server autogenerates SSL key and
certificate files in the data directory, if they do not already exist.",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "boolean",
        "AllowedValues": "0,1",
        "IsModifiable": false,
        "ApplyMethod": "pending-reboot"
    },
    ...some output truncated...
]
}
```

詳細については、「Amazon RDSユーザーガイド」の [「DB パラメータグループの操作」](#) を参照してください。

- API 詳細については、「コマンドリファレンス」の [「DescribeDBParameters」](#) を参照してください。AWS CLI

describe-db-proxies

次の例は、describe-db-proxies を使用する方法を説明しています。

AWS CLI

RDS データベースの DB プロキシを記述するには

次のdescribe-db-proxies例では、DB プロキシに関する情報を返します。

```
aws rds describe-db-proxies
```

出力:

```
{
  "DBProxies": [
    {
      "DBProxyName": "proxyExample1",
      "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
```

```
    "Status": "available",
    "EngineFamily": "PostgreSQL",
    "VpcId": "vpc-1234567",
    "VpcSecurityGroupIds": [
      "sg-1234"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Auth": "[
      {
        "Description": "proxydescription1"
        "AuthScheme": "SECRETS",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
        "IAMAuth": "DISABLED"
      }
    ]",
    "RoleArn": "arn:aws:iam::12345678912?:role/ProxyPostgreSQLRole",
    "Endpoint": "proxyExample1.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
    "RequireTLS": false,
    "IdleClientTimeout": 1800,
    "DebuggingLogging": false,
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
  },
  {
    "DBProxyName": "proxyExample2",
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-1234a12b23456c1ab",
    "Status": "available",
    "EngineFamily": "PostgreSQL",
    "VpcId": "vpc-1234567",
    "VpcSecurityGroupIds": [
      "sg-1234"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Auth": "[
      {
```



```
        "Description": "proxydescription2"
        "AuthScheme": "SECRETS",
        "SecretArn": "aarn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
        "IAMAuth": "DISABLED"
    }
  ],
  "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
  "Endpoint": "proxyExample2.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
  "RequireTLS": false,
  "IdleClientTimeout": 1800,
  "DebuggingLogging": false,
  "CreateDate": "2022-01-05T16:19:33.452000+00:00",
  "UpdateDate": "2023-04-13T01:49:38.568000+00:00"
}
]
```

詳細については、「[Amazon RDSユーザーガイド](#)」の[RDS「プロキシの表示](#)」および「[Amazon Aurora ユーザーガイド](#)」の[RDS「プロキシの表示](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeDbProxies](#)」の「」を参照してください。

describe-db-proxy-endpoints

次のコード例は、describe-db-proxy-endpoints を使用する方法を示しています。

AWS CLI

DB プロキシエンドポイントを記述するには

次のdescribe-db-proxy-endpoints例では、DB プロキシエンドポイントに関する情報を返します。

```
aws rds describe-db-proxy-endpoints
```

出力:

```
{
  "DBProxyEndpoints": [
    {
```

```
    "DBProxyEndpointName": "proxyEndpoint1",
    "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
    "DBProxyName": "proxyExample",
    "Status": "available",
    "VpcId": "vpc-1234567",
    "VpcSecurityGroupIds": [
        "sg-1234"
    ],
    "VpcSubnetIds": [
        "subnetgroup1",
        "subnetgroup2"
    ],
    "Endpoint": "proxyEndpoint1.endpoint.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "TargetRole": "READ_WRITE",
    "IsDefault": false
  },
  {
    "DBProxyEndpointName": "proxyEndpoint2",
    "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-4567a01b12345c0ab",
    "DBProxyName": "proxyExample2",
    "Status": "available",
    "VpcId": "vpc1234567",
    "VpcSecurityGroupIds": [
        "sg-5678"
    ],
    "VpcSubnetIds": [
        "subnetgroup1",
        "subnetgroup2"
    ],
    "Endpoint": "proxyEndpoint2.endpoint.proxy-cd1ef2klmnop.us-
east-1.rds.amazonaws.com",
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "TargetRole": "READ_WRITE",
    "IsDefault": false
  }
]
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[プロキシエンドポイントの表示](#)」と「[Amazon Aurora ユーザーガイド](#)」の「[プロキシエンドポイントの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeDbProxyEndpoints](#)」の「」を参照してください。

describe-db-proxy-target-groups

次のコード例は、describe-db-proxy-target-groups を使用する方法を示しています。

AWS CLI

DB プロキシエンドポイントを記述するには

次のdescribe-db-proxy-target-groups例では、DB プロキシターゲットグループに関する情報を返します。

```
aws rds describe-db-proxy-target-groups \  
  --db-proxy-name proxyExample
```

出力:

```
{  
  "TargetGroups":  
    {  
      "DBProxyName": "proxyExample",  
      "TargetGroupName": "default",  
      "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-  
tg-0123a01b12345c0ab",  
      "IsDefault": true,  
      "Status": "available",  
      "ConnectionPoolConfig": {  
        "MaxConnectionsPercent": 100,  
        "MaxIdleConnectionsPercent": 50,  
        "ConnectionBorrowTimeout": 120,  
        "SessionPinningFilters": []  
      },  
      "CreateDate": "2023-05-02T18:41:19.495000+00:00",  
      "UpdateDate": "2023-05-02T18:41:21.762000+00:00"  
    }  
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」のRDS「[プロキシの表示](#)」および「[Amazon Aurora ユーザーガイド](#)」のRDS「[プロキシの表示](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeDbProxyTargetGroups](#)」の「」を参照してください。

describe-db-proxy-targets

次の例は、describe-db-proxy-targets を使用する方法を説明しています。

AWS CLI

DB プロキシターゲットを記述するには

次のdescribe-db-proxy-targets例では、DB プロキシターゲットに関する情報を返します。

```
aws rds describe-db-proxy-targets \  
  --db-proxy-name proxyExample
```

出力:

```
{  
  "Targets": [  
    {  
      "Endpoint": "database1.ab0cd1efghij.us-east-1.rds.amazonaws.com",  
      "TrackedClusterId": "database1",  
      "RdsResourceId": "database1-instance-1",  
      "Port": 3306,  
      "Type": "RDS_INSTANCE",  
      "Role": "READ_WRITE",  
      "TargetHealth": {  
        "State": "UNAVAILABLE",  
        "Reason": "PENDING_PROXY_CAPACITY",  
        "Description": "DBProxy Target is waiting for proxy to scale to  
desired capacity"  
      }  
    }  
  ]  
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」のRDS「[プロキシの表示](#)」および「[Amazon Aurora ユーザーガイド](#)」のRDS「[プロキシの表示](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeDbProxyTargets](#)」の「」を参照してください。

describe-db-recommendations

次の例は、describe-db-recommendations を使用する方法を説明しています。

AWS CLI

例 1: すべての DB レコメンデーションを一覧表示するには

次のdescribe-db-recommendations例では、AWS アカウント内のすべての DB レコメンデーションを一覧表示します。

```
aws rds describe-db-recommendations
```

出力:

```
{
  "DBRecommendations": [
    {
      "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
      "TypeId": "config_recommendation::old_minor_version",
      "Severity": "informational",
      "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
      "Status": "active",
      "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
      "UpdatedTime": "2024-02-21T23:14:19+00:00",
      "Detection": "***[resource-name]** is not running the latest minor DB engine version",
      "Recommendation": "Upgrade to latest engine version",
      "Description": "Your database resources aren't running the latest minor DB engine version. The latest minor version contains the latest security fixes and other improvements.",
      "RecommendedActions": [
        {
          "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
          "Operation": "modifyDbInstance",
          "Parameters": [
            {
```

```

        "Key": "EngineVersion",
        "Value": "5.7.44"
    },
    {
        "Key": "DBInstanceIdentifier",
        "Value": "database-1"
    }
],
"ApplyModes": [
    "immediately",
    "next-maintenance-window"
],
>Status": "ready",
"ContextAttributes": [
    {
        "Key": "Recommended value",
        "Value": "5.7.44"
    },
    {
        "Key": "Current engine version",
        "Value": "5.7.42"
    }
]
}
],
"Category": "security",
"Source": "RDS",
"TypeDetection": "***[resource-count] resources** are not running the
latest minor DB engine version",
"TypeRecommendation": "Upgrade to latest engine version",
"Impact": "Reduced database performance and data security at risk",
"AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
which are backward-compatible with earlier minor versions of the same major version
of the DB engine.",
"Links": [
    {
        "Text": "Upgrading an RDS DB instance engine version",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
    },
    {

```

```

        "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon Aurora",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
    },
    {
        "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon RDS",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
    }
]
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon レコRDSメンデーションの表示と対応」](#) および [「Amazon Aurora ユーザーガイド」の「Amazon RDSレコメンデーションの表示と対応」](#) を参照してください。 RDS

例 2: 重要度の高い DB レコメンデーションを一覧表示するには

次のdescribe-db-recommendations例では、AWS アカウントの重要度の高い DB レコメンデーションを一覧表示します。

```

aws rds describe-db-recommendations \
  --filters Name=severity,Values=high

```

出力:

```

{
  "DBRecommendations": [
    {
      "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
      "TypeId": "config_recommendation::rds_extended_support",
      "Severity": "high",
      "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
      "Status": "active",
      "CreatedTime": "2024-02-21T23:14:19.392000+00:00",
      "UpdatedTime": "2024-02-21T23:14:19+00:00",
      "Detection": "Your databases will be auto-enrolled to RDS Extended
Support on February 29",
    }
  ]
}

```

```
"Recommendation": "Upgrade your major version before February 29, 2024
to avoid additional charges",
  "Description": "Your PostgreSQL 11 and MySQL 5.7 databases will be
automatically enrolled into RDS Extended Support on February 29, 2024. To avoid
the increase in charges due to RDS Extended Support, we recommend upgrading your
databases to a newer major engine version before February 29, 2024.\n\nTo learn more
about the RDS Extended Support pricing, refer to the pricing page.",
  "RecommendedActions": [
    {
      "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
      "Parameters": [],
      "ApplyModes": [
        "manual"
      ],
      "Status": "ready",
      "ContextAttributes": []
    }
  ],
  "Category": "cost optimization",
  "Source": "RDS",
  "TypeDetection": "Your database will be auto-enrolled to RDS Extended
Support on February 29",
  "TypeRecommendation": "Upgrade your major version before February 29,
2024 to avoid additional charges",
  "Impact": "Increase in charges due to RDS Extended Support",
  "AdditionalInfo": "With Amazon RDS Extended Support, you can continue
running your database on a major engine version past the RDS end of standard
support date for an additional cost. This paid feature gives you more time to
upgrade to a supported major engine version.\n\nDuring Extended Support, Amazon RDS
will supply critical CVE patches and bug fixes.",
  "Links": [
    {
      "Text": "Amazon RDS Extended Support pricing for RDS for MySQL",
      "Url": "https://aws.amazon.com/rds/mysql/pricing/"
    },
    {
      "Text": "Amazon RDS Extended Support for RDS for MySQL and
PostgreSQL databases",
      "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
extended-support.html"
    },
    {
      "Text": "Amazon RDS Extended Support pricing for Amazon Aurora
PostgreSQL",
```



```

        "Url": "https://aws.amazon.com/rds/aurora/pricing/"
    },
    {
        "Text": "Amazon RDS Extended Support for Aurora PostgreSQL
databases",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/extended-support.html"
    },
    {
        "Text": "Amazon RDS Extended Support pricing for RDS for
PostgreSQL",
        "Url": "https://aws.amazon.com/rds/postgresql/pricing/"
    }
]
}
]
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon レコRDSメンテーションの表示と対応」](#) および [「Amazon Aurora ユーザーガイド」の「Amazon RDSレコメンテーションの表示と対応」](#) を参照してください。 RDS

例 3: 指定された DB インスタンスの DB レコメンテーションを一覧表示するには

次のdescribe-db-recommendations例では、指定された DB インスタンスのすべての DB レコメンテーションを一覧表示します。

```
aws rds describe-db-recommendations \
  --filters Name=dbi-resource-id,Values=database-1
```

出力:

```

{
  "DBRecommendations": [
    {
      "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
      "TypeId": "config_recommendation::old_minor_version",
      "Severity": "informational",
      "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
      "Status": "active",
      "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
      "UpdatedTime": "2024-02-21T23:14:19+00:00",
    }
  ]
}

```

```
    "Detection": "***[resource-name]** is not running the latest minor DB
engine version",
    "Recommendation": "Upgrade to latest engine version",
    "Description": "Your database resources aren't running the latest minor
DB engine version. The latest minor version contains the latest security fixes and
other improvements.",
    "RecommendedActions": [
      {
        "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
        "Operation": "modifyDbInstance",
        "Parameters": [
          {
            "Key": "EngineVersion",
            "Value": "5.7.44"
          },
          {
            "Key": "DBInstanceIdentifier",
            "Value": "database-1"
          }
        ],
        "ApplyModes": [
          "immediately",
          "next-maintenance-window"
        ],
        "Status": "ready",
        "ContextAttributes": [
          {
            "Key": "Recommended value",
            "Value": "5.7.44"
          },
          {
            "Key": "Current engine version",
            "Value": "5.7.42"
          }
        ]
      }
    ],
    "Category": "security",
    "Source": "RDS",
    "TypeDetection": "***[resource-count] resources** are not running the
latest minor DB engine version",
    "TypeRecommendation": "Upgrade to latest engine version",
    "Impact": "Reduced database performance and data security at risk",
```

```

    "AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
which are backward-compatible with earlier minor versions of the same major version
of the DB engine.",
    "Links": [
      {
        "Text": "Upgrading an RDS DB instance engine version",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
      },
      {
        "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon Aurora",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
      },
      {
        "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon RDS",
        "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
      }
    ]
  }
]
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon レコRDSメンデーションの表示と対応」](#)および[「Amazon Aurora ユーザーガイド」の「Amazon RDSレコメンデーションの表示と対応」](#)を参照してください。 RDS

例 4: アクティブな DB レコメンデーションをすべて一覧表示するには

次のdescribe-db-recommendations例では、AWS アカウント内のすべてのアクティブな DB レコメンデーションを一覧表示します。

```
aws rds describe-db-recommendations \
  --filters Name=status,Values=active
```

出力:

```
{
```

```
"DBRecommendations": [
  {
    "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
    "TypeId": "config_recommendation::old_minor_version",
    "Severity": "informational",
    "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
    "Status": "active",
    "CreatedTime": "2024-02-21T23:14:19.292000+00:00",
    "UpdatedTime": "2024-02-21T23:14:19+00:00",
    "Detection": "***[resource-name]** is not running the latest minor DB
engine version",
    "Recommendation": "Upgrade to latest engine version",
    "Description": "Your database resources aren't running the latest minor
DB engine version. The latest minor version contains the latest security fixes and
other improvements.",
    "RecommendedActions": [
      {
        "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
        "Operation": "modifyDbInstance",
        "Parameters": [
          {
            "Key": "EngineVersion",
            "Value": "5.7.44"
          },
          {
            "Key": "DBInstanceIdentifier",
            "Value": "database-1"
          }
        ],
        "ApplyModes": [
          "immediately",
          "next-maintenance-window"
        ],
        "Status": "ready",
        "ContextAttributes": [
          {
            "Key": "Recommended value",
            "Value": "5.7.44"
          },
          {
            "Key": "Current engine version",
            "Value": "5.7.42"
          }
        ]
      }
    ]
  }
]
```

```

    }
  ],
  "Category": "security",
  "Source": "RDS",
  "TypeDetection": "**[resource-count] resources** are not running the
latest minor DB engine version",
  "TypeRecommendation": "Upgrade to latest engine version",
  "Impact": "Reduced database performance and data security at risk",
  "AdditionalInfo": "We recommend that you maintain your database with the
latest DB engine minor version as this version includes the latest security and
functionality fixes. The DB engine minor version upgrades contain only the changes
which are backward-compatible with earlier minor versions of the same major version
of the DB engine.",
  "Links": [
    {
      "Text": "Upgrading an RDS DB instance engine version",
      "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
USER_UpgradeDBInstance.Upgrading.html"
    },
    {
      "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon Aurora",
      "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/blue-green-deployments.html"
    },
    {
      "Text": "Using Amazon RDS Blue/Green Deployments for database
updates for Amazon RDS",
      "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
blue-green-deployments.html"
    }
  ]
}
]
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon レコRDSメンデーションの表示と対応」](#)および[「Amazon Aurora ユーザーガイド」の「Amazon RDSレコメンデーションの表示と対応」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbRecommendations](#)」の「」を参照してください。

describe-db-security-groups

次のコード例は、describe-db-security-groups を使用する方法を示しています。

AWS CLI

DB セキュリティグループを一覧表示するには

次のdescribe-db-security-groups例では、DB セキュリティグループを一覧表示します。

```
aws rds describe-db-security-groups
```

出力:

```
{
  "DBSecurityGroups": [
    {
      "OwnerId": "123456789012",
      "DBSecurityGroupName": "default",
      "DBSecurityGroupDescription": "default",
      "EC2SecurityGroups": [],
      "IPRanges": [],
      "DBSecurityGroupArn": "arn:aws:rds:us-west-1:111122223333:secgrp:default"
    },
    {
      "OwnerId": "123456789012",
      "DBSecurityGroupName": "mysecgroup",
      "DBSecurityGroupDescription": "My Test Security Group",
      "VpcId": "vpc-1234567f",
      "EC2SecurityGroups": [],
      "IPRanges": [],
      "DBSecurityGroupArn": "arn:aws:rds:us-west-1:111122223333:secgrp:mysecgroup"
    }
  ]
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[利用可能な DB セキュリティグループの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeDbSecurityGroups](#)」の「」を参照してください。

describe-db-shard-groups

次の例は、describe-db-shard-groups を使用する方法を説明しています。

AWS CLI

例 1: DB シャードグループを記述するには

次のdescribe-db-shard-groups例では、DB シャードグループの詳細を取得します。

```
aws rds describe-db-shard-groups
```

出力:

```
{
  "DBShardGroups": [
    {
      "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
      "DBShardGroupIdentifier": "limitless-test-shard-grp",
      "DBClusterIdentifier": "limitless-test-cluster",
      "MaxACU": 768.0,
      "ComputeRedundancy": 0,
      "Status": "available",
      "PubliclyAccessible": true,
      "Endpoint": "limitless-test-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"
    },
    {
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
      "DBShardGroupIdentifier": "my-db-shard-group",
      "DBClusterIdentifier": "my-sv2-cluster",
      "MaxACU": 768.0,
      "ComputeRedundancy": 0,
      "Status": "available",
      "PubliclyAccessible": false,
      "Endpoint": "my-sv2-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"
    }
  ]
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスター」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbShardGroups](#)」の「」を参照してください。

describe-db-snapshot-attributes

次のコード例は、describe-db-snapshot-attributes を使用する方法を示しています。

AWS CLI

DB スナップショットの属性名と値を記述するには

次のdescribe-db-snapshot-attributes例では、DB スナップショットの属性名と値について説明します。

```
aws rds describe-db-snapshot-attributes \  
  --db-snapshot-identifier mydbsnapshot
```

出力:

```
{  
  "DBSnapshotAttributesResult": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "123456789012",  
          "210987654321"  
        ]  
      }  
    ]  
  }  
}
```

詳細については、「Amazon RDSユーザーガイド」の「[DB スナップショットの共有](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbSnapshotAttributes](#)」の「」を参照してください。

describe-db-snapshots

次の例は、describe-db-snapshots を使用する方法を説明しています。

AWS CLI

例 1: DB インスタンスの DB スナップショットを記述するには

次の describe-db-snapshots の例は、DB インスタンスの DB スナップショットの詳細を取得します。

```
aws rds describe-db-snapshots \  
--db-snapshot-identifier mydbsnapshot
```

出力:

```
{  
  "DBSnapshots": [  
    {  
      "DBSnapshotIdentifier": "mydbsnapshot",  
      "DBInstanceIdentifier": "mysqldb",  
      "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",  
      "Engine": "mysql",  
      "AllocatedStorage": 20,  
      "Status": "available",  
      "Port": 3306,  
      "AvailabilityZone": "us-east-1f",  
      "VpcId": "vpc-6594f31c",  
      "InstanceCreateTime": "2018-02-08T22:24:55.973Z",  
      "MasterUsername": "mysqladmin",  
      "EngineVersion": "5.6.37",  
      "LicenseModel": "general-public-license",  
      "SnapshotType": "manual",  
      "OptionGroupName": "default:mysql-5-6",  
      "PercentProgress": 100,  
      "StorageType": "gp2",  
      "Encrypted": false,  
      "DBSnapshotArn": "arn:aws:rds:us-  
east-1:123456789012:snapshot:mydbsnapshot",  
      "IAMDatabaseAuthenticationEnabled": false,  
      "ProcessorFeatures": [],  
      "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"  
    }  
  ]  
}
```

```
]
}
```

詳細については、[「Amazon ユーザーガイド」の「DB スナップショットの作成」](#)を参照してください。 RDS

例 2: 手動で作成されたスナップショットの数を調べるには

次のdescribe-db-snapshots例では、--queryオプションのlength演算子を使用して、特定の AWS リージョンで作成された手動スナップショットの数を返します。

```
aws rds describe-db-snapshots \
  --snapshot-type manual \
  --query "length(*[].[DBSnapshots:SnapshotType])" \
  --region eu-central-1
```

出力:

```
35
```

詳細については、「Amazon RDSユーザーガイド」の[「DB スナップショットの作成」](#)を参照してください。

- API 詳細については、「コマンドリファレンス」の[「DescribeDBSnapshots」](#)を参照してください。 AWS CLI

describe-db-subnet-groups

次の例は、describe-db-subnet-groups を使用する方法を説明しています。

AWS CLI

DB サブネットグループを記述するには

次のdescribe-db-subnet-groups例では、指定された DB サブネットグループの詳細を取得します。

```
aws rds describe-db-subnet-groups
```

出力:

```
{
  "DBSubnetGroups": [
    {
      "DBSubnetGroupName": "mydbsubnetgroup",
      "DBSubnetGroupDescription": "My DB Subnet Group",
      "VpcId": "vpc-971c12ee",
      "SubnetGroupStatus": "Complete",
      "Subnets": [
        {
          "SubnetIdentifier": "subnet-d8c8e7f4",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-718fdc7d",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1f"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-cbc8e7e7",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        },
        {
          "SubnetIdentifier": "subnet-0ccde220",
          "SubnetAvailabilityZone": {
            "Name": "us-east-1a"
          },
          "SubnetStatus": "Active"
        }
      ],
      "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123456789012:subgrp:mydbsubnetgroup"
    }
  ]
}
```

詳細については、[Amazon Virtual Private Cloud VPCsと RDS Amazon](#) を参照してください。

RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDbSubnetGroups](#)」の「」を参照してください。

describe-engine-default-cluster-parameters

次のコード例は、describe-engine-default-cluster-parameters を使用する方法を示しています。

AWS CLI

Aurora データベースエンジンのデフォルトのエンジンおよびシステムパラメータ情報を記述するには

次のdescribe-engine-default-cluster-parameters例では、MySQL 5.7 互換の Aurora DB クラスターのデフォルトのエンジンおよびシステムパラメータ情報の詳細を取得します。

```
aws rds describe-engine-default-cluster-parameters \  
--db-parameter-group-family aurora-mysql5.7
```

出力:

```
{  
  "EngineDefaults": {  
    "Parameters": [  
      {  
        "ParameterName": "aurora_load_from_s3_role",  
        "Description": "IAM role ARN used to load data from AWS S3",  
        "Source": "engine-default",  
        "ApplyType": "dynamic",  
        "DataType": "string",  
        "IsModifiable": true,  
        "SupportedEngineModes": [  
          "provisioned"  
        ]  
      },  
      ...some output truncated...  
    ]  
  }  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB パラメータグループと DB クラスターパラメータグループの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEngineDefaultClusterParameters](#)」の「」を参照してください。

describe-engine-default-parameters

次の例は、describe-engine-default-parameters を使用する方法を説明しています。

AWS CLI

データベースエンジンのデフォルトのエンジンおよびシステムパラメータ情報を記述するには、次の describe-engine-default-parameters 例では、MySQL 5.7 DB インスタンスのデフォルトのエンジンおよびシステムパラメータ情報の詳細を取得します。

```
aws rds describe-engine-default-parameters \
  --db-parameter-group-family mysql5.7
```

出力:

```
{
  "EngineDefaults": {
    "Parameters": [
      {
        "ParameterName": "allow-suspicious-udfs",
        "Description": "Controls whether user-defined functions that have
only an xxx symbol for the main function can be loaded",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "boolean",
        "AllowedValues": "0,1",
        "IsModifiable": false
      },
      ...some output truncated...
    ]
  }
}
```

詳細については、「Amazon RDS ユーザーガイド」の「[DB パラメータグループの操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEngineDefaultParameters](#)」の「」を参照してください。

describe-event-categories

次の例は、describe-event-categories を使用する方法を説明しています。

AWS CLI

イベントカテゴリを記述するには

次のdescribe-event-categories例では、使用可能なすべてのイベントソースのイベントカテゴリの詳細を取得します。

```
aws rds describe-event-categories
```

出力:

```
{
  "EventCategoriesMapList": [
    {
      "SourceType": "db-instance",
      "EventCategories": [
        "deletion",
        "read replica",
        "failover",
        "restoration",
        "maintenance",
        "low storage",
        "configuration change",
        "backup",
        "creation",
        "availability",
        "recovery",
        "failure",
        "backtrack",
        "notification"
      ]
    },
    {
      "SourceType": "db-security-group",
      "EventCategories": [
```

```
        "configuration change",
        "failure"
    ]
},
{
    "SourceType": "db-parameter-group",
    "EventCategories": [
        "configuration change"
    ]
},
{
    "SourceType": "db-snapshot",
    "EventCategories": [
        "deletion",
        "creation",
        "restoration",
        "notification"
    ]
},
{
    "SourceType": "db-cluster",
    "EventCategories": [
        "failover",
        "failure",
        "notification"
    ]
},
{
    "SourceType": "db-cluster-snapshot",
    "EventCategories": [
        "backup"
    ]
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEventCategories](#)」の「」を参照してください。

describe-event-subscriptions

次の例は、describe-event-subscriptions を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを記述するには

この例では、現在の AWS アカウントのすべての Amazon RDS イベントサブスクリプションについて説明します。

```
aws rds describe-event-subscriptions
```

出力:

```
{
  "EventSubscriptionsList": [
    {
      "EventCategoriesList": [
        "backup",
        "recovery"
      ],
      "Enabled": true,
      "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-
instance-events",
      "Status": "creating",
      "SourceType": "db-instance",
      "CustomerAwsId": "123456789012",
      "SubscriptionCreationTime": "2018-07-31 23:22:01.893",
      "CustSubscriptionId": "my-instance-events",
      "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events"
    },
    ...some output truncated...
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEventSubscriptions](#)」の「」を参照してください。

describe-events

次のコード例は、describe-events を使用する方法を示しています。

AWS CLI

イベントを記述するには

次のdescribe-events例では、指定された DB インスタンスで発生したイベントの詳細を取得します。

```
aws rds describe-events \  
  --source-identifier test-instance \  
  --source-type db-instance
```

出力:

```
{  
  "Events": [  
    {  
      "SourceType": "db-instance",  
      "SourceIdentifier": "test-instance",  
      "EventCategories": [  
        "backup"  
      ],  
      "Message": "Backing up DB instance",  
      "Date": "2018-07-31T23:09:23.983Z",  
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"  
    },  
    {  
      "SourceType": "db-instance",  
      "SourceIdentifier": "test-instance",  
      "EventCategories": [  
        "backup"  
      ],  
      "Message": "Finished DB Instance backup",  
      "Date": "2018-07-31T23:15:13.049Z",  
      "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

describe-export-tasks

次のコード例は、describe-export-tasks を使用する方法を示しています。

AWS CLI

スナップショットエクスポートタスクを記述するには

次のdescribe-export-tasks例では、Amazon S3 へのスナップショットのエクスポートに関する情報を返します。

```
aws rds describe-export-tasks
```

出力:

```
{
  "ExportTasks": [
    {
      "ExportTaskIdentifier": "test-snapshot-export",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:test-
snapshot",
      "SnapshotTime": "2020-03-02T18:26:28.163Z",
      "TaskStartTime": "2020-03-02T18:57:56.896Z",
      "TaskEndTime": "2020-03-02T19:10:31.985Z",
      "S3Bucket": "mybucket",
      "S3Prefix": "",
      "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
      "Status": "COMPLETE",
      "PercentProgress": 100,
      "TotalExtractedDataInGB": 0
    },
    {
      "ExportTaskIdentifier": "my-s3-export",
      "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-
test",
      "SnapshotTime": "2020-03-27T20:48:42.023Z",
      "S3Bucket": "mybucket",
      "S3Prefix": "",
      "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
abcd0000-7fca-4128-82f2-aabbccddeeff",
      "Status": "STARTING",
      "PercentProgress": 0,
      "TotalExtractedDataInGB": 0
    }
  ]
}
```

```
]
}
```

詳細については、「Amazon RDSユーザーガイド」の[「スナップショットエクスポートのモニタリング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeExportTasks](#)」の「」を参照してください。

describe-global-clusters

次の例は、describe-global-clusters を使用する方法を説明しています。

AWS CLI

グローバル DB クラスターを記述するには

次のdescribe-global-clusters例では、現在の AWS リージョンの Aurora グローバル DB クラスターを一覧表示します。

```
aws rds describe-global-clusters
```

出力:

```
{
  "GlobalClusters": [
    {
      "GlobalClusterIdentifier": "myglobalcluster",
      "GlobalClusterResourceId": "cluster-f5982077e3b5aabb",
      "GlobalClusterArn": "arn:aws:rds::123456789012:global-cluster:myglobalcluster",
      "Status": "available",
      "Engine": "aurora-mysql",
      "EngineVersion": "5.7.mysql_aurora.2.07.2",
      "StorageEncrypted": false,
      "DeletionProtection": false,
      "GlobalClusterMembers": []
    }
  ]
}
```

詳細については、「Amazon [Aurora ユーザーガイド](#)」の「[Aurora グローバルデータベースの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGlobalClusters](#)」の「」を参照してください。

describe-option-group-options

次の例は、describe-option-group-options を使用する方法を説明しています。

AWS CLI

使用可能なすべてのオプションを記述するには

次のdescribe-option-group-options例では、Oracle Database 19c インスタンスの2つのオプションを一覧表示します。

```
aws rds describe-option-group-options \  
  --engine-name oracle-ee \  
  --major-engine-version 19 \  
  --max-items 2
```

出力:

```
{  
  "OptionGroupOptions": [  
    {  
      "Name": "APEX",  
      "Description": "Oracle Application Express Runtime Environment",  
      "EngineName": "oracle-ee",  
      "MajorEngineVersion": "19",  
      "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",  
      "PortRequired": false,  
      "OptionsDependedOn": [],  
      "OptionsConflictsWith": [],  
      "Persistent": false,  
      "Permanent": false,  
      "RequiresAutoMinorEngineVersionUpgrade": false,  
      "VpcOnly": false,  
      "SupportsOptionVersionDowngrade": false,  
      "OptionGroupOptionSettings": [],  
      "OptionGroupOptionVersions": [  
        {
```

```
        "Version": "19.1.v1",
        "IsDefault": true
    },
    {
        "Version": "19.2.v1",
        "IsDefault": false
    }
]
},
{
    "Name": "APEX-DEV",
    "Description": "Oracle Application Express Development Environment",
    "EngineName": "oracle-ee",
    "MajorEngineVersion": "19",
    "MinimumRequiredMinorEngineVersion": "0.0.0.ru-2019-07.rur-2019-07.r1",
    "PortRequired": false,
    "OptionsDependedOn": [
        "APEX"
    ],
    "OptionsConflictsWith": [],
    "Persistent": false,
    "Permanent": false,
    "RequiresAutoMinorEngineVersionUpgrade": false,
    "VpcOnly": false,
    "OptionGroupOptionSettings": []
}
],
"NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[オプショングループのオプションとオプション設定の一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOptionGroupOptions](#)」の「」を参照してください。

describe-option-groups

次のコード例は、describe-option-groups を使用する方法を示しています。

AWS CLI

使用可能なオプショングループを記述するには

次のdescribe-option-groups例では、Oracle Database 19c インスタンスのオプショングループを一覧表示します。

```
aws rds describe-option-groups \  
  --engine-name oracle-ee \  
  --major-engine-version 19
```

出力:

```
{  
  "OptionGroupsList": [  
    {  
      "OptionGroupName": "default:oracle-ee-19",  
      "OptionGroupDescription": "Default option group for oracle-ee 19",  
      "EngineName": "oracle-ee",  
      "MajorEngineVersion": "19",  
      "Options": [],  
      "AllowsVpcAndNonVpcInstanceMemberships": true,  
      "OptionGroupArn": "arn:aws:rds:us-west-1:111122223333:og:default:oracle-  
ee-19"  
    }  
  ]  
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[オプショングループのオプションとオプション設定の一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOptionGroups](#)」の「」を参照してください。

describe-orderable-db-instance-options

次のコード例は、describe-orderable-db-instance-options を使用する方法を示しています。

AWS CLI

順序設定可能な DB インスタンスオプションを記述するには

次のdescribe-orderable-db-instance-options例では、MySQL DB エンジンを実行している DB インスタンスの順序付け可能なオプションの詳細を取得します。

```
aws rds describe-orderable-db-instance-options \  
--engine mysql
```

出力:

```
{  
  "OrderableDBInstanceOptions": [  
    {  
      "MinStorageSize": 5,  
      "ReadReplicaCapable": true,  
      "MaxStorageSize": 6144,  
      "AvailabilityZones": [  
        {  
          "Name": "us-east-1a"  
        },  
        {  
          "Name": "us-east-1b"  
        },  
        {  
          "Name": "us-east-1c"  
        },  
        {  
          "Name": "us-east-1d"  
        }  
      ],  
      "SupportsIops": false,  
      "AvailableProcessorFeatures": [],  
      "MultiAZCapable": true,  
      "DBInstanceClass": "db.m1.large",  
      "Vpc": true,  
      "StorageType": "gp2",  
      "LicenseModel": "general-public-license",  
      "EngineVersion": "5.5.46",  
      "SupportsStorageEncryption": false,  
      "SupportsEnhancedMonitoring": true,  
      "Engine": "mysql",  
      "SupportsIAMDatabaseAuthentication": false,  
      "SupportsPerformanceInsights": false  
    }  
  ]  
  ...some output truncated...  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeOrderableDBInstanceOptions](#)」の「」を参照してください。

describe-pending-maintenance-actions

次の例は、describe-pending-maintenance-actions を使用する方法を説明しています。

AWS CLI

少なくとも 1 つの保留中のメンテナンスアクションを持つリソースを一覧表示するには

次のdescribe-pending-maintenance-actions例では、DB インスタンスの保留中のメンテナンスアクションを一覧表示します。

```
aws rds describe-pending-maintenance-actions
```

出力:

```
{
  "PendingMaintenanceActions": [
    {
      "ResourceIdentifier": "arn:aws:rds:us-west-2:123456789012:cluster:global-db1-cl1",
      "PendingMaintenanceActionDetails": [
        {
          "Action": "system-update",
          "Description": "Upgrade to Aurora PostgreSQL 2.4.2"
        }
      ]
    }
  ]
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[DB インスタンスの維持](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribePendingMaintenanceActions](#)」の「」を参照してください。

describe-reserved-db-instances-offerings

次のコード例は、describe-reserved-db-instances-offerings を使用する方法を示しています。

AWS CLI

予約済み DB インスタンスのオフアリングを記述するには

次のdescribe-reserved-db-instances-offerings例では、 の予約済み DB インスタンス オプションの詳細を取得しますoracle。

```
aws rds describe-reserved-db-instances-offerings \
  --product-description oracle
```

出力:

```
{
  "ReservedDBInstancesOfferings": [
    {
      "CurrencyCode": "USD",
      "UsagePrice": 0.0,
      "ProductDescription": "oracle-se2(li)",
      "ReservedDBInstancesOfferingId": "005bdee3-9ef4-4182-aa0c-58ef7cb6c2f8",
      "MultiAZ": true,
      "DBInstanceClass": "db.m4.xlarge",
      "OfferingType": "Partial Upfront",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.594,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "FixedPrice": 4089.0,
      "Duration": 31536000
    },
    ...some output truncated...
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeReservedDbInstancesOfferings](#)」の「」を参照してください。

describe-reserved-db-instances

次の例は、describe-reserved-db-instances を使用する方法を説明しています。

AWS CLI

予約済み DB インスタンスを記述するには

次のdescribe-reserved-db-instances例では、現在の AWS アカウントで予約された DB インスタンスの詳細を取得します。

```
aws rds describe-reserved-db-instances
```

出力:

```
{
  "ReservedDBInstances": [
    {
      "ReservedDBInstanceId": "myreservedinstance",
      "ReservedDBInstancesOfferingId": "12ab34cd-59af-4b2c-a660-1abcdef23456",
      "DBInstanceClass": "db.t3.micro",
      "StartTime": "2020-06-01T13:44:21.436Z",
      "Duration": 31536000,
      "FixedPrice": 0.0,
      "UsagePrice": 0.0,
      "CurrencyCode": "USD",
      "DBInstanceCount": 1,
      "ProductDescription": "sqlserver-ex(li)",
      "OfferingType": "No Upfront",
      "MultiAZ": false,
      "State": "payment-pending",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.014,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "ReservedDBInstanceArn": "arn:aws:rds:us-west-2:123456789012:ri:myreservedinstance",
      "LeaseId": "a1b2c3d4-6b69-4a59-be89-5e11aa446666"
    }
  ]
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon 用リザーブド DB インスタンス RDS」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReservedDbInstances](#)」の「」を参照してください。

describe-source-regions

次のコード例は、describe-source-regions を使用する方法を示しています。

AWS CLI

ソースリージョンを記述するには

次のdescribe-source-regions例では、すべてのソース AWS リージョンの詳細を取得します。また、自動バックアップは、米国西部 (オレゴン) から送信先 AWS リージョンである米国東部 (バージニア北部) にのみレプリケートできることを示しています。

```
aws rds describe-source-regions \  
  --region us-east-1
```

出力:

```
{  
  "SourceRegions": [  
    {  
      "RegionName": "af-south-1",  
      "Endpoint": "https://rds.af-south-1.amazonaws.com",  
      "Status": "available",  
      "SupportsDBInstanceAutomatedBackupsReplication": false  
    },  
    {  
      "RegionName": "ap-east-1",  
      "Endpoint": "https://rds.ap-east-1.amazonaws.com",  
      "Status": "available",  
      "SupportsDBInstanceAutomatedBackupsReplication": false  
    },  
    {  
      "RegionName": "ap-northeast-1",  
      "Endpoint": "https://rds.ap-northeast-1.amazonaws.com",  
      "Status": "available",  
      "SupportsDBInstanceAutomatedBackupsReplication": true  
    },  
  ],  
}
```

```
{
  "RegionName": "ap-northeast-2",
  "Endpoint": "https://rds.ap-northeast-2.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": true
},
{
  "RegionName": "ap-northeast-3",
  "Endpoint": "https://rds.ap-northeast-3.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": false
},
{
  "RegionName": "ap-south-1",
  "Endpoint": "https://rds.ap-south-1.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": true
},
{
  "RegionName": "ap-southeast-1",
  "Endpoint": "https://rds.ap-southeast-1.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": true
},
{
  "RegionName": "ap-southeast-2",
  "Endpoint": "https://rds.ap-southeast-2.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": true
},
{
  "RegionName": "ap-southeast-3",
  "Endpoint": "https://rds.ap-southeast-3.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": false
},
{
  "RegionName": "ca-central-1",
  "Endpoint": "https://rds.ca-central-1.amazonaws.com",
  "Status": "available",
  "SupportsDBInstanceAutomatedBackupsReplication": true
},
{
  "RegionName": "eu-north-1",
```

```
    "Endpoint": "https://rds.eu-north-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "eu-south-1",
    "Endpoint": "https://rds.eu-south-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "eu-west-1",
    "Endpoint": "https://rds.eu-west-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "eu-west-2",
    "Endpoint": "https://rds.eu-west-2.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "eu-west-3",
    "Endpoint": "https://rds.eu-west-3.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "me-central-1",
    "Endpoint": "https://rds.me-central-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "me-south-1",
    "Endpoint": "https://rds.me-south-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": false
  },
  {
    "RegionName": "sa-east-1",
    "Endpoint": "https://rds.sa-east-1.amazonaws.com",
    "Status": "available",
```

```
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "us-east-2",
    "Endpoint": "https://rds.us-east-2.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "us-west-1",
    "Endpoint": "https://rds.us-west-1.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  },
  {
    "RegionName": "us-west-2",
    "Endpoint": "https://rds.us-west-2.amazonaws.com",
    "Status": "available",
    "SupportsDBInstanceAutomatedBackupsReplication": true
  }
]
}
```

詳細については、「Amazon RDSユーザーガイド」の「[レプリケートされたバックアップに関する情報の検索](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSourceRegions](#)」の「」を参照してください。

describe-valid-db-instance-modifications

次のコード例は、describe-valid-db-instance-modifications を使用する方法を示しています。

AWS CLI

DB インスタンスの有効な変更を記述するには

次のdescribe-valid-db-instance-modifications例では、指定された DB インスタンスの有効な変更の詳細を取得します。

```
aws rds describe-valid-db-instance-modifications \
```

```
--db-instance-identifier test-instance
```

出力:

```
{
  "ValidDBInstanceModificationsMessage": {
    "ValidProcessorFeatures": [],
    "Storage": [
      {
        "StorageSize": [
          {
            "Step": 1,
            "To": 20,
            "From": 20
          },
          {
            "Step": 1,
            "To": 6144,
            "From": 22
          }
        ],
        "ProvisionedIops": [
          {
            "Step": 1,
            "To": 0,
            "From": 0
          }
        ],
        "IopsToStorageRatio": [
          {
            "To": 0.0,
            "From": 0.0
          }
        ],
        "StorageType": "gp2"
      },
      {
        "StorageSize": [
          {
            "Step": 1,
            "To": 6144,
            "From": 100
          }
        ]
      }
    ]
  }
}
```

```
    ],
    "ProvisionedIops": [
      {
        "Step": 1,
        "To": 40000,
        "From": 1000
      }
    ],
    "IopsToStorageRatio": [
      {
        "To": 50.0,
        "From": 1.0
      }
    ],
    "StorageType": "io1"
  },
  {
    "StorageSize": [
      {
        "Step": 1,
        "To": 20,
        "From": 20
      },
      {
        "Step": 1,
        "To": 3072,
        "From": 22
      }
    ],
    "ProvisionedIops": [
      {
        "Step": 1,
        "To": 0,
        "From": 0
      }
    ],
    "IopsToStorageRatio": [
      {
        "To": 0.0,
        "From": 0.0
      }
    ],
    "StorageType": "magnetic"
  }
}
```



```
    ]  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeValidDbInstanceModifications](#)」の「」を参照してください。

download-db-log-file-portion

次のコード例は、download-db-log-file-portion を使用する方法を示しています。

AWS CLI

DB ログファイルをダウンロードするには

次のdownload-db-log-file-portion例では、ログファイルの最新の部分のみをダウンロードし、`tail.txt` という名前のローカルファイルに保存します。

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier test-instance \  
  --log-file-name log.txt \  
  --output text > tail.txt
```

ファイル全体をダウンロードするには、`--starting-token 0`パラメータを含める必要があります。次の例では、出力を `full.txt` という名前のローカルファイルに保存します。

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier test-instance \  
  --log-file-name log.txt \  
  --starting-token 0 \  
  --output text > full.txt
```

保存されたファイルには空白行が含まれている可能性があります。ダウンロード中にログファイルの各部分の最後に表示されます。これは通常、ログファイル分析で問題を引き起こすことはありません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DownloadDbLogFilePortion](#)」の「」を参照してください。

generate-auth-token

次のコード例は、generate-auth-token を使用する方法を示しています。

AWS CLI

認証トークンを生成するには

次のgenerate-db-auth-token例では、IAMデータベース認証で使用する認証トークンを生成します。

```
aws rds generate-db-auth-token \  
  --hostname aurmysql-test.cdgmuiadpid.us-west-2.rds.amazonaws.com \  
  --port 3306 \  
  --region us-east-1 \  
  --username jane_doe
```

出力:

```
aurmysql-test.cdgmuiadpid.us-west-2.rds.amazonaws.com:3306/?  
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-  
Credential=AKIAIESZCNJ30EXAMPLE%2F20180731%2Fus-east-1%2Frds-db%2Faws4_request&X-  
Amz-Date=20180731T235209Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-  
Signature=5a8753ebEXAMPLEa2c724e5667797EXAMPLE9d6ec6e3f427191fa41aeEXAMPLE
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GenerateAuthToken](#)」の「」を参照してください。

generate-db-auth-token

次のコード例は、generate-db-auth-token を使用する方法を示しています。

AWS CLI

IAM認証トークンを生成するには

次のgenerate-db-auth-token例では、データベースに接続するためのIAM認証トークンを生成します。

```
aws rds generate-db-auth-token \  
  --hostname aurmysql-test.cdgmuiadpid.us-west-2.rds.amazonaws.com \  
  --port 3306 \  
  --region us-east-1 \  
  --username jane_doe
```

```
--hostname mydb.123456789012.us-east-1.rds.amazonaws.com \  
--port 3306 \  
--region us-east-1 \  
--username db_user
```

出力:

```
mydb.123456789012.us-east-1.rds.amazonaws.com:3306/?  
Action=connect&DBUser=db_user&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-  
Credential=AKIAIEXAMPLE%2Fus-east-1%2Frds-db%2Faws4_request&X-Amz-  
Date=20210123T011543Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-  
Signature=88987EXAMPLE1EXAMPLE2EXAMPLE3EXAMPLE4EXAMPLE5EXAMPLE6
```

詳細については、「[Amazon RDSユーザーガイド](#)」のIAM「[認証を使用した DB インスタンスへの接続](#)」および「[Amazon Aurora ユーザーガイド](#)」のIAM「[認証を使用した DB クラスターへの接続](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGenerateDbAuthToken](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

Amazon RDSリソースのタグを一覧表示するには

次のlist-tags-for-resource例では、DB インスタンスのすべてのタグを一覧表示します。

```
aws rds list-tags-for-resource \  
--resource-name arn:aws:rds:us-east-1:123456789012:db:orcl1
```

出力:

```
{  
  "TagList": [  
    {  
      "Key": "Environment",  
      "Value": "test"  
    },  
  ],  
}
```

```
{
  "Key": "Name",
  "Value": "MyDatabase"
}
]
```

詳細については、[「Amazon ユーザーガイド」の「Amazon RDSリソースのタグ付け」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

modify-certificates

次のコード例は、`modify-certificates` を使用する方法を示しています。

AWS CLI

新しい DB インスタンスのシステムデフォルトSSL/TLS証明書を一時的に上書きするには

次の`modify-certificates`例では、新しい DB インスタンスのシステムデフォルト SSL/TLS 証明書を一時的に上書きします。

```
aws rds modify-certificates \
  --certificate-identifier rds-ca-2019
```

出力:

```
{
  "Certificate": {
    "CertificateIdentifier": "rds-ca-2019",
    "CertificateType": "CA",
    "Thumbprint": "EXAMPLE123456789012",
    "ValidFrom": "2019-09-19T18:16:53Z",
    "ValidTill": "2024-08-22T17:08:50Z",
    "CertificateArn": "arn:aws:rds:us-east-1::cert:rds-ca-2019",
    "CustomerOverride": true,
    "CustomerOverrideValidTill": "2024-08-22T17:08:50Z"
  }
}
```

詳細については、[「Amazon ユーザーガイド」のSSL「/TLS 証明書のローテーション」](#)および[「Amazon Aurora ユーザーガイド」のSSL「/TLS 証明書のローテーション」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyCertificates](#)」の「」を参照してください。

modify-current-db-cluster-capacity

次のコード例は、`modify-current-db-cluster-capacity` を使用する方法を示しています。

AWS CLI

Aurora Serverless DB クラスターの容量を拡張するには

次の`modify-current-db-cluster-capacity`例では、Aurora Serverless DB クラスターの容量を 8 にスケールリングします。

```
aws rds modify-current-db-cluster-capacity \  
  --db-cluster-identifier mydbcluster \  
  --capacity 8
```

出力:

```
{  
  "DBClusterIdentifier": "mydbcluster",  
  "PendingCapacity": 8,  
  "CurrentCapacity": 1,  
  "SecondsBeforeTimeout": 300,  
  "TimeoutAction": "ForceApplyCapacityChange"  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Aurora Serverless v1 DB クラスター容量を手動でスケールリングする」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyCurrentDbClusterCapacity](#)」の「」を参照してください。

modify-db-cluster-endpoint

次の例は、`modify-db-cluster-endpoint` を使用する方法を説明しています。

AWS CLI

カスタム DB クラスターエンドポイントを変更するには

次のmodify-db-cluster-endpoint例では、指定されたカスタム DB クラスターエンドポイントを変更します。

```
aws rds modify-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint \  
  --static-members dbinstance1 dbinstance2 dbinstance3
```

出力:

```
{  
  "DBClusterEndpointIdentifier": "mycustomendpoint",  
  "DBClusterIdentifier": "mydbcluster",  
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",  
  "Endpoint": "mycustomendpoint.cluster-custom-cnpxample.us-east-1.rds.amazonaws.com",  
  "Status": "modifying",  
  "EndpointType": "CUSTOM",  
  "CustomEndpointType": "READER",  
  "StaticMembers": [  
    "dbinstance1",  
    "dbinstance2",  
    "dbinstance3"  
  ],  
  "ExcludedMembers": [],  
  "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:mycustomendpoint"  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora Connection Management」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbClusterEndpoint](#)」の「」を参照してください。

modify-db-cluster-parameter-group

次の例は、modify-db-cluster-parameter-group を使用する方法を説明しています。

AWS CLI

DB クラスターパラメータグループのパラメータを変更するには

次のmodify-db-cluster-parameter-group例では、DB クラスターパラメータグループのパラメータの値を変更します。

```
aws rds modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterpg \  
  --  
  parameters "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" \  
  \  
  "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

出力:

```
{  
  "DBClusterParameterGroupName": "mydbclusterpg"  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[DB パラメータグループと DB クラスターパラメータグループの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbClusterParameterGroup](#)」の「」を参照してください。

modify-db-cluster-snapshot-attribute

次のコード例は、modify-db-cluster-snapshot-attribute を使用する方法を示しています。

AWS CLI

DB クラスタースナップショット属性を変更するには

次のmodify-db-cluster-snapshot-attribute例では、指定された DB クラスタースナップショット属性を変更します。

```
aws rds modify-db-cluster-snapshot-attribute \  
  --db-cluster-snapshot-identifier myclustersnapshot \  
  --attribute-name restore \  
  \  
  "
```

```
--values-to-add 123456789012
```

出力:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "myclustersnapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012"
        ]
      }
    ]
  }
}
```

詳細については、「[Amazon Aurora ユーザーガイド](#)」の「[DB クラスタースナップショットからの復元](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス ModifyDbClusterSnapshotAttribute](#)」の「」を参照してください。

modify-db-cluster

次の例は、modify-db-cluster を使用する方法を説明しています。

AWS CLI

例 1: DB クラスターを変更するには

次のmodify-db-cluster例では、という名前の DB クラスターのマスターユーザーパスワードを変更cluster-2し、バックアップ保持期間を 14 日間に設定します。--apply-immediately パラメータにより、次のメンテナンスウィンドウまで待たずに、変更がすぐに行われます。

```
aws rds modify-db-cluster \  
  --db-cluster-identifier cluster-2 \  
  --backup-retention-period 14 \  
  --master-user-password newpassword99 \  
  --apply-immediately
```


出力:

```
{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "eu-central-1b",
      "eu-central-1c",
      "eu-central-1a"
    ],
    "BackupRetentionPeriod": 14,
    "DatabaseName": "",
    "DBClusterIdentifier": "cluster-2",
    "DBClusterParameterGroup": "default.aurora5.6",
    "DBSubnetGroup": "default-vpc-2305ca49",
    "Status": "available",
    "EarliestRestorableTime": "2020-06-03T02:07:29.637Z",
    "Endpoint": "cluster-2.cluster-#####.eu-central-1.rds.amazonaws.com",
    "ReaderEndpoint": "cluster-2.cluster-ro-#####.eu-central-1.rds.amazonaws.com",
    "MultiAZ": false,
    "Engine": "aurora",
    "EngineVersion": "5.6.10a",
    "LatestRestorableTime": "2020-06-04T15:11:25.748Z",
    "Port": 3306,
    "MasterUsername": "admin",
    "PreferredBackupWindow": "01:55-02:25",
    "PreferredMaintenanceWindow": "thu:21:14-thu:21:44",
    "ReadReplicaIdentifiers": [],
    "DBClusterMembers": [
      {
        "DBInstanceIdentifier": "cluster-2-instance-1",
        "IsClusterWriter": true,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      }
    ],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-20a5c047",
        "Status": "active"
      }
    ],
    "HostedZoneId": "Z1RLNU0EXAMPLE",
  }
}
```

```

    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:eu-central-1:123456789012:key/
d1bd7c8f-5cdb-49ca-8a62-a1b2c3d4e5f6",
    "DbClusterResourceId": "cluster-AGJ7XI77XVIS6FUXHU1EXAMPLE",
    "DBClusterArn": "arn:aws:rds:eu-central-1:123456789012:cluster:cluster-2",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "ClusterCreateTime": "2020-04-03T14:44:02.764Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": true,
    "CrossAccountClone": false,
    "DomainMemberships": []
  }
}

```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターの変更」](#)を参照してください。

例 2: VPC セキュリティグループを DB クラスターに関連付けるには

次のmodify-db-instance例では、特定のVPCセキュリティグループを関連付け、DB クラスターから DB セキュリティグループを削除します。

```

aws rds modify-db-cluster \
  --db-cluster-identifier dbName \
  --vpc-security-group-ids sg-ID

```

出力:

```

{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-west-2c",
      "us-west-2b",
      "us-west-2a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "dbName",
    "DBClusterParameterGroup": "default.aurora-mysql8.0",

```

```
    "DBSubnetGroup": "default",
    "Status": "available",
    "EarliestRestorableTime": "2024-02-15T01:12:13.966000+00:00",
    "Endpoint": "dbName.cluster-abcdefghji.us-west-2.rds.amazonaws.com",
    "ReaderEndpoint": "dbName.cluster-ro-abcdefghji.us-
west-2.rds.amazonaws.com",
    "MultiAZ": false,
    "Engine": "aurora-mysql",
    "EngineVersion": "8.0.mysql_aurora.3.04.1",
    "LatestRestorableTime": "2024-02-15T02:25:33.696000+00:00",
    "Port": 3306,
    "MasterUsername": "admin",
    "PreferredBackupWindow": "10:59-11:29",
    "PreferredMaintenanceWindow": "thu:08:54-thu:09:24",
    "ReadReplicaIdentifiers": [],
    "DBClusterMembers": [
      {
        "DBInstanceIdentifier": "dbName-instance-1",
        "IsClusterWriter": true,
        "DBClusterParameterGroupStatus": "in-sync",
        "PromotionTier": 1
      }
    ],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-ID",
        "Status": "active"
      }
    ],
    ...output omitted...
  }
}
```

詳細については、「Amazon Aurora ユーザーガイド」の [「セキュリティグループによるアクセスの制御」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyDbCluster](#)」の「」を参照してください。

modify-db-instance

次の例は、modify-db-instance を使用する方法を説明しています。

AWS CLI

例 1: DB インスタンスを変更するには

次のmodify-db-instance例では、オプショングループとパラメータグループを互換性のある Microsoft SQL Server DB インスタンスに関連付けます。--apply-immediately パラメータを使用することで、次のメンテナンスウィンドウを待つことなく、オプショングループとパラメータグループをすぐに関連付けることができます。

```
aws rds modify-db-instance \  
  --db-instance-identifier database-2 \  
  --option-group-name test-se-2017 \  
  --db-parameter-group-name test-sqlserver-se-2017 \  
  --apply-immediately
```

出力:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "database-2",  
    "DBInstanceClass": "db.r4.large",  
    "Engine": "sqlserver-se",  
    "DBInstanceStatus": "available",  
  
    ...output omitted...  
  
    "DBParameterGroups": [  
      {  
        "DBParameterGroupName": "test-sqlserver-se-2017",  
        "ParameterApplyStatus": "applying"  
      }  
    ],  
    "AvailabilityZone": "us-west-2d",  
  
    ...output omitted...  
  
    "MultiAZ": true,  
    "EngineVersion": "14.00.3281.6.v1",  
    "AutoMinorVersionUpgrade": false,  
    "ReadReplicaDBInstanceIdentifiers": [],  
    "LicenseModel": "license-included",  
    "OptionGroupMemberships": [  
      {
```

```

        "OptionGroupName": "test-se-2017",
        "Status": "pending-apply"
    }
],
"CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
"SecondaryAvailabilityZone": "us-west-2c",
"PubliclyAccessible": true,
"StorageType": "gp2",

...output omitted...

"DeletionProtection": false,
"AssociatedRoles": [],
"MaxAllocatedStorage": 1000
}
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon RDS DB インスタンスの変更」](#)を参照してください。 RDS

例 2: VPC セキュリティグループを DB インスタンスに関連付けるには

次のmodify-db-instance例では、特定のVPCセキュリティグループを関連付け、DB インスタンスから DB セキュリティグループを削除します。

```

aws rds modify-db-instance \
  --db-instance-identifier dbName \
  --vpc-security-group-ids sg-ID

```

出力:

```

{
  "DBInstance": {
    "DBInstanceIdentifier": "dbName",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "admin",
    "Endpoint": {
      "Address": "dbName.abcdefghijkl.us-west-2.rds.amazonaws.com",
      "Port": 3306,
      "HostedZoneId": "ABCDEFGHIJK1234"
    }
  }
}

```

```
  },
  "AllocatedStorage": 20,
  "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",
  "PreferredBackupWindow": "11:57-12:27",
  "BackupRetentionPeriod": 7,
  "DBSecurityGroups": [],
  "VpcSecurityGroups": [
    {
      "VpcSecurityGroupId": "sg-ID",
      "Status": "active"
    }
  ],
  ... output omitted ...
  "MultiAZ": false,
  "EngineVersion": "8.0.35",
  "AutoMinorVersionUpgrade": true,
  "ReadReplicaDBInstanceIdentifiers": [],
  "LicenseModel": "general-public-license",

  ... output omitted ...
}
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[セキュリティグループによるアクセスの制御](#)」を参照してください。

- API 詳細については、「[コマンドリファレンス](#)」の「[ModifyDBInstance](#)」を参照してください。AWS CLI

modify-db-parameter-group

次のコード例は、`modify-db-parameter-group` を使用する方法を示しています。

AWS CLI

DB パラメータグループを変更するには

次の `modify-db-parameter-group` の例は、DB パラメータグループの `clr enabled` パラメータの値を変更します。 `--apply-immediately` パラメータを使用することで、次のメンテナンスウィンドウを待つことなく、DB パラメータグループをすぐに変更することができます。

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name test-sqlserver-se-2017 \  
  --apply-immediately
```

```
--parameters "ParameterName='clr
enabled',ParameterValue=1,ApplyMethod=immediate"
```

出力:

```
{
  "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[DB パラメータグループのパラメータの変更](#)」を参照してください。

- API 詳細については、「[コマンドリファレンス](#)」の「[ModifyDBParameterグループ](#)」を参照してください。AWS CLI

modify-db-proxy-endpoint

次の例は、`modify-db-proxy-endpoint` を使用する方法を説明しています。

AWS CLI

RDS データベースの DB プロキシエンドポイントを変更するには

次の`modify-db-proxy-endpoint`例では、DB プロキシエンドポイントを変更`proxyEndpoint`して、読み取りタイムアウトを 65 秒に設定します。

```
aws rds modify-db-proxy-endpoint \
  --db-proxy-endpoint-name proxyEndpoint \
  --cli-read-timeout 65
```

出力:

```
{
  "DBProxyEndpoint":
    {
      "DBProxyEndpointName": "proxyEndpoint",
      "DBProxyEndpointArn": "arn:aws:rds:us-east-1:123456789012:db-proxy-
endpoint:prx-endpoint-0123a01b12345c0ab",
      "DBProxyName": "proxyExample",
      "Status": "available",
      "VpcId": "vpc-1234567",
    }
}
```

```
    "VpcSecurityGroupIds": [  
      "sg-1234"  
    ],  
    "VpcSubnetIds": [  
      "subnetgroup1",  
      "subnetgroup2"  
    ],  
    "Endpoint": "proxyEndpoint.endpoint.proxyExample-ab0cd1efghij.us-east-1.rds.amazonaws.com",  
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",  
    "TargetRole": "READ_WRITE",  
    "IsDefault": "false"  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「プロキシエンドポイントの変更」](#)および「[Amazon Aurora ユーザーガイド」の「プロキシエンドポイントの変更」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「[コマンドリファレンスModifyDbProxyEndpoint](#)」の「」を参照してください。

modify-db-proxy-target-group

次の例は、`modify-db-proxy-target-group` を使用する方法を説明しています。

AWS CLI

DB プロキシエンドポイントを変更するには

次の`modify-db-proxy-target-group`例では、DB プロキシターゲットグループを変更して、最大接続を 80%、最大アイドル接続を 10% に設定します。

```
aws rds modify-db-proxy-target-group \  
  --target-group-name default \  
  --db-proxy-name proxyExample \  
  --connection-pool-config MaxConnectionsPercent=80,MaxIdleConnectionsPercent=10
```

出力:

```
{  
  "DBProxyTargetGroup":
```



```
{
  "DBProxyName": "proxyExample",
  "TargetGroupName": "default",
  "TargetGroupArn": "arn:aws:rds:us-east-1:123456789012:target-group:prx-
tg-0123a01b12345c0ab",
  "IsDefault": true,
  "Status": "available",
  "ConnectionPoolConfig": {
    "MaxConnectionsPercent": 80,
    "MaxIdleConnectionsPercent": 10,
    "ConnectionBorrowTimeout": 120,
    "SessionPinningFilters": []
  },
  "CreateDate": "2023-05-02T18:41:19.495000+00:00",
  "UpdateDate": "2023-05-02T18:41:21.762000+00:00"
}
```

詳細については、[「Amazon ユーザーガイド」のRDS「プロキシの変更」](#) および「Amazon Aurora ユーザーガイド」の[RDS「プロキシの変更」](#)を参照してください。 RDS

- API 詳細については、AWS CLI「コマンドリファレンス[ModifyDbProxyTargetGroup](#)」の「」を参照してください。

modify-db-proxy

次のコード例は、modify-db-proxy を使用する方法を示しています。

AWS CLI

RDS データベースの DB プロキシを変更するには

次のmodify-db-proxy例では、という名前の DB プロキシを変更proxyExampleして、接続 SSLに を要求するようにしています。

```
aws rds modify-db-proxy \
  --db-proxy-name proxyExample \
  --require-tls
```

出力:

```
{
```

```
"DBProxy":
  {
    "DBProxyName": "proxyExample",
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
    "Status": "modifying"
    "EngineFamily": "PostgreSQL",
    "VpcId": "sg-1234567",
    "VpcSecurityGroupIds": [
      "sg-1234"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Auth": "[
      {
        "Description": "proxydescription1",
        "AuthScheme": "SECRETS",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
        "IAMAuth": "DISABLED"
      }
    ]",
    "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
    "RequireTLS": true,
    "IdleClientTimeout": 1800,
    "DebuggingLogging": false,
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
  }
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の[RDS「プロキシの変更](#)」と「[Amazon Aurora ユーザーガイド](#)」の[RDS「プロキシの作成](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスModifyDbProxy](#)」の「」を参照してください。

modify-db-shard-group

次のコード例は、modify-db-shard-group を使用する方法を示しています。

AWS CLI

例 1: DB シャードグループを変更するには

次のmodify-db-shard-group例では、DB シャードグループの最大容量を変更します。

```
aws rds modify-db-shard-group \  
  --db-shard-group-identifier my-db-shard-group \  
  --max-acu 1000
```

出力:

```
{  
  "DBShardGroups": [  
    {  
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",  
      "DBShardGroupIdentifier": "my-db-shard-group",  
      "DBClusterIdentifier": "my-sv2-cluster",  
      "MaxACU": 768.0,  
      "ComputeRedundancy": 0,  
      "Status": "available",  
      "PubliclyAccessible": false,  
      "Endpoint": "my-sv2-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"  
    }  
  ]  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスタ」](#)を参照してください。

例 2: DB シャードグループを記述するには

次のdescribe-db-shard-groups例では、modify-db-shard-group コマンドの実行後に DB シャードグループの詳細を取得します。DB シャードグループの最大容量my-db-shard-groupは、1000 Aurora 容量ユニット () になりましたACUs。

```
aws rds describe-db-shard-groups
```

出力:

```
{
```

```
"DBShardGroups": [
  {
    "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
    "DBShardGroupIdentifier": "limitless-test-shard-grp",
    "DBClusterIdentifier": "limitless-test-cluster",
    "MaxACU": 768.0,
    "ComputeRedundancy": 0,
    "Status": "available",
    "PubliclyAccessible": true,
    "Endpoint": "limitless-test-cluster.limitless-cekyexample.us-
east-2.rds.amazonaws.com"
  },
  {
    "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
    "DBShardGroupIdentifier": "my-db-shard-group",
    "DBClusterIdentifier": "my-sv2-cluster",
    "MaxACU": 1000.0,
    "ComputeRedundancy": 0,
    "Status": "available",
    "PubliclyAccessible": false,
    "Endpoint": "my-sv2-cluster.limitless-cekyexample.us-
east-2.rds.amazonaws.com"
  }
]
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスター」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbShardGroup](#)」の「」を参照してください。

modify-db-snapshot-attribute

次の例は、modify-db-snapshot-attribute を使用する方法を説明しています。

AWS CLI

例 1: 2 つの AWS アカウントで DB スナップショットを復元できるようにするには

次のmodify-db-snapshot-attribute例では、という名前の DB スナップショットを復元するためのアクセス許可を、識別子111122223333とを持つ 4444555566662 つの AWS アカウントに付与しますmydbsnapshot。

```
aws rds modify-db-snapshot-attribute \  
  --db-snapshot-identifier mydbsnapshot \  
  --attribute-name restore \  
  --values-to-add {"111122223333","444455556666"}
```

出力:

```
{  
  "DBSnapshotAttributesResult": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "111122223333",  
          "444455556666"  
        ]  
      }  
    ]  
  }  
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[スナップショットの共有](#)」を参照してください。

例 2: AWS アカウントが DB スナップショットを復元できないようにするには

次のmodify-db-snapshot-attribute例では、という名前の DB スナップショットを復元するアクセス許可を特定の AWS アカウントから削除しますmydbsnapshot。単一のアカウントを指定する場合、アカウント識別子を引用符や中括弧で囲むことはできません。

```
aws rds modify-db-snapshot-attribute \  
  --db-snapshot-identifier mydbsnapshot \  
  --attribute-name restore \  
  --values-to-remove 444455556666
```

出力:

```
{  
  "DBSnapshotAttributesResult": {
```

```
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "111122223333"
        ]
      }
    ]
  }
}
```

詳細については、「Amazon RDSユーザーガイド」の「[スナップショットの共有](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbSnapshotAttribute](#)」の「」を参照してください。

modify-db-snapshot-attributes

次のコード例は、modify-db-snapshot-attributes を使用する方法を示しています。

AWS CLI

DB スナップショット属性を変更するには

次のmodify-db-snapshot-attribute例では、という名前の DB スナップショットを復元するために444455556666、2つのAWSアカウント識別子111122223333とを許可しますmydbsnapshot。

```
aws rds modify-db-snapshot-attribute \
  --db-snapshot-identifier mydbsnapshot \
  --attribute-name restore \
  --values-to-add '["111122223333", "444455556666"]'
```

出力:

```
{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
```

```
{
  "AttributeName": "restore",
  "AttributeValues": [
    "111122223333",
    "444455556666"
  ]
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[スナップショットの共有](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスModifyDbSnapshotAttributes](#)」の「」を参照してください。

modify-db-snapshot

次の例は、modify-db-snapshot を使用する方法を説明しています。

AWS CLI

DB スナップショットを変更するには

次のmodify-db-snapshot例では、`db5-snapshot-upg-test` という名前の PostgreSQL 10.6 スナップショットdb5-snapshot-upg-testを PostgreSQL 11.7 にアップグレードします。新しい DB エンジンバージョンは、スナップショットのアップグレードが完了し、そのステータスが使用可能になった後に表示されます。

```
aws rds modify-db-snapshot \
  --db-snapshot-identifier db5-snapshot-upg-test \
  --engine-version 11.7
```

出力:

```
{
  "DBSnapshot": {
    "DBSnapshotIdentifier": "db5-snapshot-upg-test",
    "DBInstanceIdentifier": "database-5",
    "SnapshotCreateTime": "2020-03-27T20:49:17.092Z",
```

```
"Engine": "postgres",
"AllocatedStorage": 20,
"Status": "upgrading",
"Port": 5432,
"AvailabilityZone": "us-west-2a",
"VpcId": "vpc-2ff27557",
"InstanceCreateTime": "2020-03-27T19:59:04.735Z",
"MasterUsername": "postgres",
"EngineVersion": "10.6",
"LicenseModel": "postgresql-license",
"SnapshotType": "manual",
"OptionGroupName": "default:postgres-11",
"PercentProgress": 100,
"StorageType": "gp2",
"Encrypted": false,
"DBSnapshotArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-
upg-test",
"IAMDatabaseAuthenticationEnabled": false,
"ProcessorFeatures": [],
"DbiResourceId": "db-GJMF75LM42IL6BTFRE4UZJ5YM4"
}
}
```

詳細については、「Amazon RDSユーザーガイド」の [「PostgreSQL DB スナップショットのアップグレード」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyDbSnapshot](#)」の「」を参照してください。

modify-db-subnet-group

次の例は、modify-db-subnet-group を使用する方法を説明しています。

AWS CLI

DB サブネットグループを変更するには

次のmodify-db-subnet-group例では、という名前subnet-08e41f9e230222222のDB サブネットグループに ID を持つサブネットを追加しますmysubnetgroup。既存のサブネットをサブネットグループに保持するには、--subnet-idsオプションに値IDsとして含めます。DB サブネットグループに少なくとも2つの異なるアベイラビリティゾーンを持つサブネットがあることを確認してください。


```
aws rds modify-db-subnet-group \  
  --db-subnet-group-name mysubnetgroup \  
  --subnet-ids  
  ["subnet-0a1dc4e1a6f123456", "subnet-070dd7ecb3aaaaaaaa", "subnet-00f5b198bc0abcdef", "subnet-
```

出力:

```
{  
  "DBSubnetGroup": {  
    "DBSubnetGroupName": "mysubnetgroup",  
    "DBSubnetGroupDescription": "test DB subnet group",  
    "VpcId": "vpc-0f08e7610a1b2c3d4",  
    "SubnetGroupStatus": "Complete",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-08e41f9e230222222",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        },  
        "SubnetStatus": "Active"  
      },  
      {  
        "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaaa",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2b"  
        },  
        "SubnetStatus": "Active"  
      },  
      {  
        "SubnetIdentifier": "subnet-00f5b198bc0abcdef",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2d"  
        },  
        "SubnetStatus": "Active"  
      },  
      {  
        "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2b"  
        },  
        "SubnetStatus": "Active"  
      }  
    ],  
  },  
}
```

```
    "DBSubnetGroupArn": "arn:aws:rds:us-  
west-2:534026745191:subgrp:mysubnetgroup"  
  }  
}
```

詳細については、「Amazon RDSユーザーガイド」の[「ステップ 3: DB サブネットグループを作成する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDbSubnetGroup](#)」の「」を参照してください。

modify-event-subscription

次のコード例は、`modify-event-subscription` を使用する方法を示しています。

AWS CLI

イベントサブスクリプションを変更するには

次の`modify-event-subscription`例では、指定されたイベントサブスクリプションを無効にするため、指定された Amazon Simple Notification Service トピックに通知を発行しなくなります。

```
aws rds modify-event-subscription \  
  --subscription-name my-instance-events \  
  --no-enabled
```

出力:

```
{  
  "EventSubscription": {  
    "EventCategoriesList": [  
      "backup",  
      "recovery"  
    ],  
    "CustomerAwsId": "123456789012",  
    "SourceType": "db-instance",  
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",  
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-  
events",  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",  
    "CustSubscriptionId": "my-instance-events",
```

```
    "Status": "modifying",
    "Enabled": false
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyEventSubscription](#)」の「」を参照してください。

modify-global-cluster

次のコード例は、`modify-global-cluster` を使用する方法を示しています。

AWS CLI

グローバル DB クラスターを変更するには

次の`modify-global-cluster`例では、Aurora My SQL互換のグローバル DB クラスターの削除保護を有効にします。

```
aws rds modify-global-cluster \  
  --global-cluster-identifier myglobalcluster \  
  --deletion-protection
```

出力:

```
{  
  "GlobalCluster": {  
    "GlobalClusterIdentifier": "myglobalcluster",  
    "GlobalClusterResourceId": "cluster-f0e523bfe07aabb",  
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-  
cluster:myglobalcluster",  
    "Status": "available",  
    "Engine": "aurora-mysql",  
    "EngineVersion": "5.7.mysql_aurora.2.07.2",  
    "StorageEncrypted": false,  
    "DeletionProtection": true,  
    "GlobalClusterMembers": []  
  }  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」](#)の「[Aurora グローバルデータベースの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyGlobalCluster](#)」の「」を参照してください。

promote-read-replica-db-cluster

次の例は、promote-read-replica-db-cluster を使用する方法を説明しています。

AWS CLI

DB クラスターリードレプリカを昇格させるには

次のpromote-read-replica-db-cluster例では、指定されたリードレプリカをスタンダード DB クラスターに昇格させます。

```
aws rds promote-read-replica-db-cluster \  
  --db-cluster-identifier mydbcluster-1
```

出力:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1c"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "mydbcluster-1",  
    "...some output truncated..."  
  }  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[リードレプリカを DB クラスターにする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PromoteReadReplicaDbCluster](#)」の「」を参照してください。

promote-read-replica

次のコード例は、promote-read-replica を使用する方法を示しています。

AWS CLI

リードレプリカを昇格させるには

次のpromote-read-replica例では、指定されたリードレプリカをスタンドアロン DB インスタンスに昇格させます。

```
aws rds promote-read-replica \  
  --db-instance-identifier test-instance-repl
```

出力:

```
{  
  "DBInstance": {  
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",  
    "StorageType": "standard",  
    "ReadReplicaSourceDBInstanceIdentifier": "test-instance",  
    "DBInstanceStatus": "modifying",  
    ...some output truncated...  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PromoteReadReplica](#)」の「」を参照してください。

purchase-reserved-db-instance

次の例は、purchase-reserved-db-instance を使用する方法を説明しています。

AWS CLI

リザーブド DB インスタンスサービスを購入するには

次のpurchase-reserved-db-instances-offering例では、リザーブド DB インスタンスサービスを購入します。は、describe-reserved-db-instances-offering コマンドによって返される有効なオファリング ID reserved-db-instances-offering-idである必要があります。

```
aws rds purchase-reserved-db-instances-offering --reserved-db-instances-offering-id
438012d3-4a52-4cc7-b2e3-8dff72e0e706
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PurchaseReservedDbInstance](#)」の「」を参照してください。

purchase-reserved-db-instances-offerings

次の例は、`purchase-reserved-db-instances-offerings` を使用する方法を説明しています。

AWS CLI

例 1: 購入する予約済み DB インスタンスを検索するには

次の `describe-reserved-db-instances-offerings` 例では、`db.t2.micro` インスタンスクラスと 1 年間の利用可能な予約済み MySQL DB インスタンスを一覧表示します。オファリング ID は、リザーブド DB インスタンスの購入に必要です。

```
aws rds describe-reserved-db-instances-offerings \
  --product-description mysql \
  --db-instance-class db.t2.micro \
  --duration 1
```

出力:

```
{
  "ReservedDBInstancesOfferings": [
    {
      "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
      "DBInstanceClass": "db.t2.micro",
      "Duration": 31536000,
      "FixedPrice": 51.0,
      "UsagePrice": 0.0,
      "CurrencyCode": "USD",
      "ProductDescription": "mysql",
      "OfferingType": "Partial Upfront",
      "MultiAZ": false,
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.006,
          "RecurringChargeFrequency": "Hourly"
        }
      ]
    }
  ]
}
```

```

    }
  ]
},
... some output truncated ...
]
}

```

詳細については、[「Amazon ユーザーガイド」の「Amazon のリザーブド DB インスタンス RDS」](#)を参照してください。 RDS

例 2: 予約済み DB インスタンスを購入するには

次のpurchase-reserved-db-instances-offering例は、前の例から予約された DB インスタンス製品を購入する方法を示しています。

```
aws rds purchase-reserved-db-instances-offering --reserved-db-instances-offering-id 8ba30be1-b9ec-447f-8f23-6114e3f4c7b4
```

出力:

```

{
  "ReservedDBInstance": {
    "ReservedDBInstanceId": "ri-2020-06-29-16-54-57-670",
    "ReservedDBInstancesOfferingId": "8ba30be1-b9ec-447f-8f23-6114e3f4c7b4",
    "DBInstanceClass": "db.t2.micro",
    "StartTime": "2020-06-29T16:54:57.670Z",
    "Duration": 31536000,
    "FixedPrice": 51.0,
    "UsagePrice": 0.0,
    "CurrencyCode": "USD",
    "DBInstanceCount": 1,
    "ProductDescription": "mysql",
    "OfferingType": "Partial Upfront",
    "MultiAZ": false,
    "State": "payment-pending",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": 0.006,
        "RecurringChargeFrequency": "Hourly"
      }
    ],
    "ReservedDBInstanceArn": "arn:aws:rds:us-west-2:123456789012:ri:ri-2020-06-29-16-54-57-670"
  }
}

```

```
}  
}
```

詳細については、[「Amazon ユーザーガイド」の「Amazon のリザーブド DB インスタンス RDS」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス [PurchaseReservedDbInstancesOfferings](#)」の「」を参照してください。

reboot-db-instance

次のコード例は、reboot-db-instance を使用する方法を示しています。

AWS CLI

DB インスタンスを再起動するには

次の reboot-db-instance の例は、指定された DB インスタンスを再起動します。

```
aws rds reboot-db-instance \  
  --db-instance-identifier test-mysql-instance
```

出力:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "test-mysql-instance",  
    "DBInstanceClass": "db.t3.micro",  
    "Engine": "mysql",  
    "DBInstanceStatus": "rebooting",  
    "MasterUsername": "admin",  
    "Endpoint": {  
      "Address": "test-mysql-instance.#####.us-  
west-2.rds.amazonaws.com",  
      "Port": 3306,  
      "HostedZoneId": "Z1PVIF0EXAMPLE"  
    },  
    ... output omitted...  
  }  
}
```



```
}
```

詳細については、「Amazon RDSユーザーガイド」の[「DB インスタンスの再起動」](#)を参照してください。

- API 詳細については、「コマンドリファレンス」の[「RebootDBInstance」](#)を参照してください。AWS CLI

reboot-db-shard-group

次のコード例は、reboot-db-shard-group を使用する方法を示しています。

AWS CLI

例 1: DB シャードグループを再起動するには

次のreboot-db-shard-group例では、DB シャードグループを再起動します。

```
aws rds reboot-db-shard-group \  
  --db-shard-group-identifier my-db-shard-group
```

出力:

```
{  
  "DBShardGroups": [  
    {  
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",  
      "DBShardGroupIdentifier": "my-db-shard-group",  
      "DBClusterIdentifier": "my-sv2-cluster",  
      "MaxACU": 1000.0,  
      "ComputeRedundancy": 0,  
      "Status": "available",  
      "PubliclyAccessible": false,  
      "Endpoint": "my-sv2-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"  
    }  
  ]  
}
```

詳細については、[「Amazon Aurora ユーザーガイド」](#)の[「Amazon Aurora DB クラスターまたは Amazon Aurora DB インスタンスの再起動」](#)を参照してください。

例 2: DB シャードグループを記述するには

次のdescribe-db-shard-groups例では、reboot-db-shard-group コマンドの実行後に DB シャードグループの詳細を取得します。DB シャードグループmy-db-shard-groupを再起動しています。

```
aws rds describe-db-shard-groups
```

出力:

```
{
  "DBShardGroups": [
    {
      "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
      "DBShardGroupIdentifier": "limitless-test-shard-grp",
      "DBClusterIdentifier": "limitless-test-cluster",
      "MaxACU": 768.0,
      "ComputeRedundancy": 0,
      "Status": "available",
      "PubliclyAccessible": true,
      "Endpoint": "limitless-test-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"
    },
    {
      "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
      "DBShardGroupIdentifier": "my-db-shard-group",
      "DBClusterIdentifier": "my-sv2-cluster",
      "MaxACU": 1000.0,
      "ComputeRedundancy": 0,
      "Status": "rebooting",
      "PubliclyAccessible": false,
      "Endpoint": "my-sv2-cluster.limitless-cekyexample.us-east-2.rds.amazonaws.com"
    }
  ]
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターまたは Amazon Aurora DB インスタンスの再起動」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootDbShardGroup](#)」の「」を参照してください。

register-db-proxy-targets

次のコード例は、register-db-proxy-targets を使用する方法を示しています。

AWS CLI

DB プロキシをデータベースに登録するには

次のregister-db-proxy-targets例では、データベースとプロキシ間の関連付けを作成します。

```
aws rds register-db-proxy-targets \  
  --db-proxy-name proxyExample \  
  --db-cluster-identifiers database-5
```

出力:

```
{  
  "DBProxyTargets": [  
    {  
      "RdsResourceId": "database-5",  
      "Port": 3306,  
      "Type": "TRACKED_CLUSTER",  
      "TargetHealth": {  
        "State": "REGISTERING"  
      }  
    },  
    {  
      "Endpoint": "database-5instance-1.ab0cd1efghij.us-east-1.rds.amazonaws.com",  
      "RdsResourceId": "database-5",  
      "Port": 3306,  
      "Type": "RDS_INSTANCE",  
      "TargetHealth": {  
        "State": "REGISTERING"  
      }  
    }  
  ]  
}
```

詳細については、[「Amazon ユーザーガイド」のRDS「プロキシの作成」](#)および「Amazon Aurora ユーザーガイド」の[RDS「プロキシの作成」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス [RegisterDbProxyTargets](#)」の「」を参照してください。

remove-from-global-cluster

次のコード例は、`remove-from-global-cluster` を使用する方法を示しています。

AWS CLI

Aurora グローバルデータベースクラスターから Aurora セカンダリクラスターをデタッチするには

次の `remove-from-global-cluster` 例では、Aurora グローバルデータベースクラスターから Aurora セカンダリクラスターをデタッチします。クラスターは読み取り専用から、読み取り/書き込み機能を備えたスタンドアロンクラスターに変わります。

```
aws rds remove-from-global-cluster \  
  --region us-west-2 \  
  --global-cluster-identifier myglobalcluster \  
  --db-cluster-identifier arn:aws:rds:us-west-2:123456789012:cluster:DB-1
```

出力:

```
{  
  "GlobalCluster": {  
    "GlobalClusterIdentifier": "myglobalcluster",  
    "GlobalClusterResourceId": "cluster-abc123def456gh",  
    "GlobalClusterArn": "arn:aws:rds::123456789012:global-  
cluster:myglobalcluster",  
    "Status": "available",  
    "Engine": "aurora-postgresql",  
    "EngineVersion": "10.11",  
    "StorageEncrypted": true,  
    "DeletionProtection": false,  
    "GlobalClusterMembers": [  
      {  
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:js-  
global-cluster",  
        "Readers": [  
          "arn:aws:rds:us-west-2:123456789012:cluster:DB-1"  
        ],  
        "IsWriter": true
```

```
    },
    {
      "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:DB-1",
      "Readers": [],
      "IsWriter": false,
      "GlobalWriteForwardingStatus": "disabled"
    }
  ]
}
```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora グローバルデータベースからクラスターを削除する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveFromGlobalCluster](#)」の「」を参照してください。

remove-option-from-option-group

次の例は、remove-option-from-option-group を使用する方法を説明しています。

AWS CLI

オプショングループからオプションを削除するには

次のremove-option-from-option-group例では、 から OEMオプションを削除しますmyoptiongroup。

```
aws rds remove-option-from-option-group \
  --option-group-name myoptiongroup \
  --options OEM \
  --apply-immediately
```

出力:

```
{
  "OptionGroup": {
    "OptionGroupName": "myoptiongroup",
    "OptionGroupDescription": "Test",
    "EngineName": "oracle-ee",
    "MajorEngineVersion": "19",
    "Options": [],
```

```
"AllowsVpcAndNonVpcInstanceMemberships": true,  
  "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"  
}  
}
```

詳細については、「[Amazon Aurora ユーザーガイド](#)」の「[オプショングループからオプションを削除する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveOptionFromOptionGroup](#)」の「」を参照してください。

remove-role-from-db-cluster

次のコード例は、`remove-role-from-db-cluster` を使用する方法を示しています。

AWS CLI

DB クラスターから AWS Identity and Access Management (IAM) ロールの関連付けを解除するには

次の`remove-role-from-db-cluster`例では、DB クラスターからロールを削除します。

```
aws rds remove-role-from-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Aurora ユーザーガイド](#)」の「[Amazon Aurora MySQL DB クラスターとIAMロールの関連付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveRoleFromDbCluster](#)」の「」を参照してください。

remove-role-from-db-instance

次の例は、`remove-role-from-db-instance` を使用する方法を説明しています。

AWS CLI

DB インスタンスから AWS Identity and Access Management (IAM) ロールの関連付けを解除するには

次のremove-role-from-db-instance例では、という名前の Oracle DB インスタンスrds-s3-integration-roleから という名前のロールを削除しますtest-instance。

```
aws rds remove-role-from-db-instance \  
  --db-instance-identifier test-instance \  
  --feature-name S3_INTEGRATION \  
  --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

このコマンドでは何も出力されません。

詳細については、「Amazon RDSユーザーガイド」の[S3 とのRDSSQLサーバー統合の無効化](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveRoleFromDbInstance](#)」の「」を参照してください。

remove-source-identifier-from-subscription

次の例は、remove-source-identifier-from-subscription を使用する方法を説明しています。

AWS CLI

サブスクリプションからソース識別子を削除するには

次のremove-source-identifier例では、既存のサブスクリプションから指定されたソース識別子を削除します。

```
aws rds remove-source-identifier-from-subscription \  
  --subscription-name my-instance-events \  
  --source-identifier test-instance-repl
```

出力:

```
{  
  "EventSubscription": {  
    "EventSubscriptionArn": "arn:aws:rds:us-east-1:123456789012:es:my-instance-events",  
    "SubscriptionCreationTime": "Tue Jul 31 23:22:01 UTC 2018",  
    "EventCategoriesList": [  
      "backup",  
      "recovery"  
    ]  
  }  
}
```

```
    ],  
    "SnsTopicArn": "arn:aws:sns:us-east-1:123456789012:interesting-events",  
    "Status": "modifying",  
    "CustSubscriptionId": "my-instance-events",  
    "CustomerAwsId": "123456789012",  
    "SourceIdsList": [  
        "test-instance"  
    ],  
    "SourceType": "db-instance",  
    "Enabled": false  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveSourceIdentifierFromSubscription](#)」の「」を参照してください。

remove-tags-from-resource

次の例は、`remove-tags-from-resource` を使用する方法を説明しています。

AWS CLI

リソースからタグを削除する

次の`remove-tags-from-resource`例では、リソースからタグを削除します。

```
aws rds remove-tags-from-resource \  
  --resource-name arn:aws:rds:us-east-1:123456789012:db:mydbinstance \  
  --tag-keys Name Environment
```

このコマンドでは何も出力されません。

詳細については、[「Amazon ユーザーガイド」の「Amazon RDSリソースのタグ付け」](#)および [「Amazon Aurora ユーザーガイド」の「Amazon RDSリソースのタグ付け」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveTagsFromResource](#)」の「」を参照してください。

reset-db-cluster-parameter-group

次の例は、`reset-db-cluster-parameter-group` を使用する方法を説明しています。

AWS CLI

例 1: すべてのパラメータをデフォルト値にリセットするには

次のreset-db-cluster-parameter-group例では、お客様が作成した DB クラスターパラメータグループのすべてのパラメータ値をデフォルト値にリセットします。

```
aws rds reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclpg \  
  --reset-all-parameters
```

出力:

```
{  
  "DBClusterParameterGroupName": "mydbclpg"  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の [「DB パラメータグループと DB クラスターパラメータグループの使用」](#) を参照してください。

例 2: 特定のパラメータをデフォルト値にリセットするには

次のreset-db-cluster-parameter-group例では、特定のパラメータのパラメータ値を、お客様が作成した DB クラスターパラメータグループのデフォルト値にリセットします。

```
aws rds reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclpgy \  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" \  
  "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

出力:

```
{  
  "DBClusterParameterGroupName": "mydbclpgy"  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の [「DB パラメータグループと DB クラスターパラメータグループの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ResetDbClusterParameterGroup](#)」の「」を参照してください。

reset-db-parameter-group

次の例は、reset-db-parameter-group を使用する方法を説明しています。

AWS CLI

例 1: すべてのパラメータをデフォルト値にリセットするには

次のreset-db-parameter-group例では、お客様が作成した DB パラメータグループのすべてのパラメータ値をデフォルト値にリセットします。

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --reset-all-parameters
```

出力:

```
{  
  "DBParameterGroupName": "mypg"  
}
```

詳細については、「Amazon RDSユーザーガイド」の[「DB パラメータグループの使用」](#)および[「Amazon Aurora ユーザーガイド」の「DB パラメータグループと DB クラスターパラメータグループの使用」](#)を参照してください。

例 2: 特定のパラメータをデフォルト値にリセットするには

次のreset-db-parameter-group例では、特定のパラメータのパラメータ値を、お客様が作成した DB パラメータグループのデフォルト値にリセットします。

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" \  
               "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

出力:

```
{  
  "DBParameterGroupName": "mypg"  
}
```

詳細については、「Amazon RDSユーザーガイド」の「[DB パラメータグループの使用](#)」および「[Amazon Aurora ユーザーガイド](#)」の「[DB パラメータグループと DB クラスターパラメータグループの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetDbParameterGroup](#)」の「」を参照してください。

restore-db-cluster-from-s3

次のコード例は、restore-db-cluster-from-s3 を使用する方法を示しています。

AWS CLI

Amazon S3 から Amazon Aurora DB クラスターを復元するには

次のrestore-db-cluster-from-s3例では、Amazon Aurora MySQL バージョン 5.7 互換 DB クラスターを Amazon S3 の MySQL 5.7 DB バックアップファイルから復元します。Amazon S3

```
aws rds restore-db-cluster-from-s3 \  
  --db-cluster-identifier cluster-s3-restore \  
  --engine aurora-mysql \  
  --master-username admin \  
  --master-user-password mypassword \  
  --s3-bucket-name mybucket \  
  --s3-prefix test-backup \  
  --s3-ingestion-role-arn arn:aws:iam::123456789012:role/service-role/TestBackup \  
  --source-engine mysql \  
  --source-engine-version 5.7.28
```

出力:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DBClusterIdentifier": "cluster-s3-restore",  
    "DBClusterParameterGroup": "default.aurora-mysql5.7",
```

```
    "DBSubnetGroup": "default",
    "Status": "creating",
    "Endpoint": "cluster-s3-restore.cluster-co3xyzabc123.us-
west-2.rds.amazonaws.com",
    "ReaderEndpoint": "cluster-s3-restore.cluster-ro-co3xyzabc123.us-
west-2.rds.amazonaws.com",
    "MultiAZ": false,
    "Engine": "aurora-mysql",
    "EngineVersion": "5.7.12",
    "Port": 3306,
    "MasterUsername": "admin",
    "PreferredBackupWindow": "11:15-11:45",
    "PreferredMaintenanceWindow": "thu:12:19-thu:12:49",
    "ReadReplicaIdentifiers": [],
    "DBClusterMembers": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-#####",
        "Status": "active"
      }
    ],
    "HostedZoneId": "Z1PVIF0EXAMPLE",
    "StorageEncrypted": false,
    "DbClusterResourceId": "cluster-SU5THYQQH0WCXZZDGXREXAMPLE",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:cluster-s3-
restore",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "ClusterCreateTime": "2020-07-27T14:22:08.095Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": false,
    "CrossAccountClone": false,
    "DomainMemberships": []
  }
}
```

詳細については、[Amazon S3 バケットを使用した MySQL からのデータの移行](#) を参照してください。

- API 詳細については、「コマンドリファレンス」の[RestoreDbClusterFromS3](#) を参照してください。AWS CLI

restore-db-cluster-from-snapshot

次の例は、restore-db-cluster-from-snapshot を使用する方法を説明しています。

AWS CLI

スナップショットから DB クラスターを復元するには

以下は、PostgreSQL バージョン 10.7 と互換性のある Aurora PostgreSQL DB クラスターを、という名前の DB クラスタースナップショットから restore-db-cluster-from-snapshot 復元します test-instance-snapshot。

```
aws rds restore-db-cluster-from-snapshot \  
  --db-cluster-identifier newdbcluster \  
  --snapshot-identifier test-instance-snapshot \  
  --engine aurora-postgresql \  
  --engine-version 10.7
```

出力:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 7,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "newdbcluster",  
    "DBClusterParameterGroup": "default.aurora-postgresql10",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "newdbcluster.cluster-#####.us-west-2.rds.amazonaws.com",  
    "ReaderEndpoint": "newdbcluster.cluster-ro-#####.us-  
west-2.rds.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "aurora-postgresql",  
    "EngineVersion": "10.7",  
    "Port": 5432,  
    "MasterUsername": "postgres",
```

```
"PreferredBackupWindow": "09:33-10:03",
"PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
"ReadReplicaIdentifiers": [],
"DBClusterMembers": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-#####",
    "Status": "active"
  }
],
"HostedZoneId": "Z1PVIF0EXAMPLE",
"StorageEncrypted": true,
"KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
"DbClusterResourceId": "cluster-5DSB5IFQDDUVAWOUWM1EXAMPLE",
"DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:newdbcluster",
"AssociatedRoles": [],
"IAMDatabaseAuthenticationEnabled": false,
"ClusterCreateTime": "2020-06-05T15:06:58.634Z",
"EngineMode": "provisioned",
"DeletionProtection": false,
"HttpEndpointEnabled": false,
"CopyTagsToSnapshot": false,
"CrossAccountClone": false,
"DomainMemberships": []
}
}
```

詳細については、「[Amazon Aurora ユーザーガイド](#)」の「[DB クラスタースナップショットからの復元](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreDbClusterFromSnapshot](#)」の「」を参照してください。

restore-db-cluster-to-point-in-time

次の例は、restore-db-cluster-to-point-in-time を使用する方法を説明しています。

AWS CLI

DB クラスターを指定された時刻に復元するには

次のrestore-db-cluster-to-point-in-time例では、という名前の DB クラスターを可能な限り最新の時刻database-4に復元します。を復元タイプcopy-on-writeとして使用すると、新しい DB クラスターがソース DB クラスターのクローンとして復元されます。

```
aws rds restore-db-cluster-to-point-in-time \  
  --source-db-cluster-identifier database-4 \  
  --db-cluster-identifier sample-cluster-clone \  
  --restore-type copy-on-write \  
  --use-latest-restorable-time
```

出力:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 7,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "sample-cluster-clone",  
    "DBClusterParameterGroup": "default.aurora-postgresql10",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "sample-cluster-clone.cluster-#####.us-  
west-2.rds.amazonaws.com",  
    "ReaderEndpoint": "sample-cluster-clone.cluster-ro-#####.us-  
west-2.rds.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "aurora-postgresql",  
    "EngineVersion": "10.7",  
    "Port": 5432,  
    "MasterUsername": "postgres",  
    "PreferredBackupWindow": "09:33-10:03",  
    "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",  
    "ReadReplicaIdentifiers": [],  
    "DBClusterMembers": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-#####",  
        "Status": "active"  
      }  
    ]  
  }  
}
```

```
    }
  ],
  "HostedZoneId": "Z1PVIF0EXAMPLE",
  "StorageEncrypted": true,
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
  "DbClusterResourceId": "cluster-BIZ77GDSA2XBSTNPFW1EXAMPLE",
  "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
clone",
  "AssociatedRoles": [],
  "IAMDatabaseAuthenticationEnabled": false,
  "CloneGroupId": "8d19331a-099a-45a4-b4aa-11aa22bb33cc44dd",
  "ClusterCreateTime": "2020-03-10T19:57:38.967Z",
  "EngineMode": "provisioned",
  "DeletionProtection": false,
  "HttpEndpointEnabled": false,
  "CopyTagsToSnapshot": false,
  "CrossAccountClone": false
}
}
```

詳細については、「[Amazon Aurora ユーザーガイド](#)」の「[DB クラスターを指定された時刻に復元する](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスRestoreDbClusterToPointInTime](#)」の「」を参照してください。

restore-db-instance-from-db-snapshot

次の例は、restore-db-instance-from-db-snapshot を使用する方法を説明しています。

AWS CLI

DB スナップショットから DB インスタンスを復元するには

次のrestore-db-instance-from-db-snapshot例では、指定された DB スナップショットから db.t3.small DB インスタンスクラスdb7-new-instanceで という名前の新しい DB インスタンスを作成します。スナップショットが作成されたソース DB インスタンスは、廃止された DB インスタンスクラスを使用するため、アップグレードできません。

```
aws rds restore-db-instance-from-db-snapshot \
  --db-instance-identifier db7-new-instance \
```



```
--db-snapshot-identifier db7-test-snapshot \  
--db-instance-class db.t3.small
```

出力:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "db7-new-instance",  
    "DBInstanceClass": "db.t3.small",  
    "Engine": "mysql",  
    "DBInstanceStatus": "creating",  
  
    ...output omitted...  
  
    "PreferredMaintenanceWindow": "mon:07:37-mon:08:07",  
    "PendingModifiedValues": {},  
    "MultiAZ": false,  
    "EngineVersion": "5.7.22",  
    "AutoMinorVersionUpgrade": true,  
    "ReadReplicaDBInstanceIdentifiers": [],  
    "LicenseModel": "general-public-license",  
  
    ...output omitted...  
  
    "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:db7-new-instance",  
    "IAMDatabaseAuthenticationEnabled": false,  
    "PerformanceInsightsEnabled": false,  
    "DeletionProtection": false,  
    "AssociatedRoles": []  
  }  
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[DB スナップショットからの復元](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RestoreDbInstanceFromDbSnapshot](#)」の「」を参照してください。

restore-db-instance-from-s3

次のコード例は、`restore-db-instance-from-s3` を使用する方法を示しています。

AWS CLI

Amazon S3 のバックアップから DB インスタンスを復元するには

次のrestore-db-instance-from-s3例では、S3 バケット内の既存のバックアップ restored-test-instanceから という名前の新しい DB my-backups インスタンスを作成します。

```
aws rds restore-db-instance-from-s3 \  
  --db-instance-identifier restored-test-instance \  
  --allocated-storage 250 --db-instance-class db.m4.large --engine mysql \  
  --master-username master --master-user-password secret99 \  
  --s3-bucket-name my-backups --s3-ingestion-role-  
arn arn:aws:iam::123456789012:role/my-role \  
  --source-engine mysql --source-engine-version 5.6.27
```

- API 詳細については、「コマンドリファレンス」の[RestoreDbInstanceFromS3](#)を参照してください。AWS CLI

restore-db-instance-to-point-in-time

次の例は、restore-db-instance-to-point-in-time を使用する方法を説明しています。

AWS CLI

例 1: DB インスタンスを特定の時点に復元するには

次のrestore-db-instance-to-point-in-time例ではrestored-test-instance、指定した時刻に という名前の新しい DB インスタンスtest-instanceに復元します。

```
aws rds restore-db-instance-to-point-in-time \  
  --source-db-instance-identifier test-instance \  
  --target-db-instance restored-test-instance \  
  --restore-time 2018-07-30T23:45:00.000Z
```

出力:

```
{  
  "DBInstance": {  
    "AllocatedStorage": 20,  
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:restored-test-  
instance",
```

```

    "DBInstanceStatus": "creating",
    "DBInstanceIdentifier": "restored-test-instance",
    ...some output omitted...
  }
}

```

詳細については、「Amazon RDSユーザーガイド」の「[DB インスタンスを指定された時間に復元する](#)」を参照してください。

例 2: DB インスタンスをレプリケートされたバックアップから指定された時間に復元するには次の `restore-db-instance-to-point-in-time` 例では、Oracle DB インスタンスをレプリケートされた自動バックアップから指定された時間に復元します。

```

aws rds restore-db-instance-to-point-in-time \
  --source-db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfg5rv7replzadausbrktni2bn4example" \
  --target-db-instance-identifier myorclinstance-from-replicated-backup \
  --restore-time 2020-12-08T18:45:00.000Z

```

出力:

```

{
  "DBInstance": {
    "DBInstanceIdentifier": "myorclinstance-from-replicated-backup",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "oracle-se2",
    "DBInstanceStatus": "creating",
    "MasterUsername": "admin",
    "DBName": "ORCL",
    "AllocatedStorage": 20,
    "PreferredBackupWindow": "07:45-08:15",
    "BackupRetentionPeriod": 14,
    ... some output omitted ...
    "DbiResourceId": "db-KGLXG75BGVIWKQT7NQ4EXAMPLE",
    "CACertificateIdentifier": "rds-ca-2019",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
    "MonitoringInterval": 0,
    "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:myorclinstance-from-replicated-backup",
    "IAMDatabaseAuthenticationEnabled": false,

```

```
    "PerformanceInsightsEnabled": false,  
    "DeletionProtection": false,  
    "AssociatedRoles": [],  
    "TagList": []  
  }  
}
```

詳細については、[「Amazon ユーザーガイド」の「レプリケートされたバックアップから指定された時刻への復元」](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreDbInstanceToPointInTime](#)」の「」を参照してください。

start-activity-stream

次の例は、start-activity-stream を使用する方法を説明しています。

AWS CLI

データベースアクティビティストリームを開始するには

次のstart-activity-stream例では、 という名前の Aurora クラスターをモニタリングするための非同期アクティビティストリームを開始します my-pg-cluster。

```
aws rds start-activity-stream \  
  --region us-east-1 \  
  --mode async \  
  --kms-key-id arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-h123-456i789jk0l1 \  
  --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \  
  --apply-immediately
```

出力:

```
{  
  "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-h123-456i789jk0l1",  
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH11JKLM2NOPQ3R4S",  
  "Status": "starting",  
  "Mode": "async",  
  "ApplyImmediately": true  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[データベースアクティビティストリームの開始](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartActivityStream](#)」の「」を参照してください。

start-db-cluster

次のコード例は、start-db-cluster を使用する方法を示しています。

AWS CLI

DB クラスターを起動するには

次のstart-db-cluster例では、DB クラスターとその DB インスタンスを起動します。

```
aws rds start-db-cluster \  
  --db-cluster-identifier mydbcluster
```

出力:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1e",  
      "us-east-1b"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DatabaseName": "mydb",  
    "DBClusterIdentifier": "mydbcluster",  
    "...some output truncated..."  
  }  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[Amazon Aurora DB クラスターの停止と起動](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartDbCluster](#)」の「」を参照してください。

start-db-instance-automated-backups-replication

次の例は、start-db-instance-automated-backups-replication を使用する方法を説明しています。

AWS CLI

クロスリージョン自動バックアップを有効にするには

次のstart-db-instance-automated-backups-replication例では、米国東部 (バージニア北部) リージョンの DB インスタンスから米国西部 (オレゴン) に自動バックアップをレプリケートします。バックアップの保持期間は 14 日間です。

```
aws rds start-db-instance-automated-backups-replication \  
  --region us-west-2 \  
  --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db" \  
  --backup-retention-period 14
```

出力:

```
{  
  "DBInstanceAutomatedBackup": {  
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",  
    "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",  
    "Region": "us-east-1",  
    "DBInstanceIdentifier": "new-orcl-db",  
    "RestoreWindow": {},  
    "AllocatedStorage": 20,  
    "Status": "pending",  
    "Port": 1521,  
    "InstanceCreateTime": "2020-12-04T15:28:31Z",  
    "MasterUsername": "admin",  
    "Engine": "oracle-se2",  
    "EngineVersion": "12.1.0.2.v21",  
    "LicenseModel": "bring-your-own-license",  
    "OptionGroupName": "default:oracle-se2-12-1",  
    "Encrypted": false,  
    "StorageType": "gp2",  
    "IAMDatabaseAuthenticationEnabled": false,  
    "BackupRetentionPeriod": 14,  
    "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfgq5rv7replzadtausbrktni2bn4example"  
  }  
}
```

```
}
```

詳細については、「Amazon RDSユーザーガイド」の[「クロスリージョン自動バックアップの有効化」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartDbInstanceAutomatedBackupsReplication](#)」の「」を参照してください。

start-db-instance

次の例は、start-db-instance を使用する方法を説明しています。

AWS CLI

DB インスタンスを起動するには

次のstart-db-instance例では、指定された DB インスタンスを起動します。

```
aws rds start-db-instance \  
  --db-instance-identifier test-instance
```

出力:

```
{  
  "DBInstance": {  
    "DBInstanceStatus": "starting",  
    ...some output truncated...  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartDbInstance](#)」の「」を参照してください。

start-export-task

次の例は、start-export-task を使用する方法を説明しています。

AWS CLI

スナップショットを Amazon S3 にエクスポートするには

次のstart-export-task例では、 という名前の DB スナップショットを という名前の Amazon S3 バケットdb5-snapshot-testにエクスポートしますmybucket。

```
aws rds start-export-task \  
  --export-task-identifier my-s3-export \  
  --source-arn arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test \  
  --s3-bucket-name mybucket \  
  --iam-role-arn arn:aws:iam::123456789012:role/service-role/ExportRole \  
  --kms-key-id arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-aabbccddeeff
```

出力:

```
{  
  "ExportTaskIdentifier": "my-s3-export",  
  "SourceArn": "arn:aws:rds:us-west-2:123456789012:snapshot:db5-snapshot-test",  
  "SnapshotTime": "2020-03-27T20:48:42.023Z",  
  "S3Bucket": "mybucket",  
  "IamRoleArn": "arn:aws:iam::123456789012:role/service-role/ExportRole",  
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/abcd0000-7fca-4128-82f2-aabbccddeeff",  
  "Status": "STARTING",  
  "PercentProgress": 0,  
  "TotalExtractedDataInGB": 0  
}
```

詳細については、[Amazon S3 バケットへのスナップショットのエクスポート](#)を参照してください。 RDS

- API 詳細については、AWS CLI 「コマンドリファレンス[StartExportTask](#)」の「」を参照してください。

stop-activity-stream

次のコード例は、stop-activity-stream を使用する方法を示しています。

AWS CLI

データベースアクティビティストリームを停止するには

次のstop-activity-stream例では、 という名前の Aurora クラスター内のアクティビティストリームを停止します my-pg-cluster。


```
aws rds stop-activity-stream \  
  --region us-east-1 \  
  --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \  
  --apply-immediately
```

出力:

```
{  
  "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-  
h123-456i789jk011",  
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH11JKLM2NOPQ3R4S",  
  "Status": "stopping"  
}
```

詳細については、「Amazon Aurora ユーザーガイド」の「[アクティビティストリームの停止](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopActivityStream](#)」の「」を参照してください。

stop-db-cluster

次のコード例は、stop-db-cluster を使用する方法を示しています。

AWS CLI

DB クラスターを停止するには

次のstop-db-cluster例では、DB クラスターとその DB インスタンスを停止します。

```
aws rds stop-db-cluster \  
  --db-cluster-identifier mydbcluster
```

出力:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1e",  
      "us-east-1b"  
    ]  
  }  
}
```

```

    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "mydb",
    "DBClusterIdentifier": "mydbcluster",
    ...some output truncated...
  }
}

```

詳細については、[「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターの停止と起動」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopDbCluster](#)」の「」を参照してください。

stop-db-instance-automated-backups-replication

次の例は、stop-db-instance-automated-backups-replication を使用する方法を説明しています。

AWS CLI

自動バックアップのレプリケートを停止するには

以下は、米国西部 (オレゴン) リージョンへの自動バックアップのレプリケーションstop-db-instance-automated-backups-replicationを終了します。レプリケートされたバックアップは、設定されたバックアップ保持期間に従って保持されます。

```

aws rds stop-db-instance-automated-backups-replication \
  --region us-west-2 \
  --source-db-instance-arn "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db"

```

出力:

```

{
  "DBInstanceAutomatedBackup": {
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:new-orcl-db",
    "DbiResourceId": "db-JKIB2GFQ5RV7REPLZA4EXAMPLE",
    "Region": "us-east-1",
    "DBInstanceIdentifier": "new-orcl-db",
    "RestoreWindow": {
      "EarliestTime": "2020-12-04T23:13:21.030Z",
      "LatestTime": "2020-12-07T19:59:57Z"
    }
  }
}

```

```
    },
    "AllocatedStorage": 20,
    "Status": "replicating",
    "Port": 1521,
    "InstanceCreateTime": "2020-12-04T15:28:31Z",
    "MasterUsername": "admin",
    "Engine": "oracle-se2",
    "EngineVersion": "12.1.0.2.v21",
    "LicenseModel": "bring-your-own-license",
    "OptionGroupName": "default:oracle-se2-12-1",
    "Encrypted": false,
    "StorageType": "gp2",
    "IAMDatabaseAuthenticationEnabled": false,
    "BackupRetentionPeriod": 7,
    "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
  }
}
```

詳細については、「Amazon RDSユーザーガイド」の[「自動バックアップレプリケーションの停止」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopDbInstanceAutomatedBackupsReplication](#)」の「」を参照してください。

stop-db-instance

次のコード例は、stop-db-instance を使用する方法を示しています。

AWS CLI

DB インスタンスを停止するには

次のstop-db-instance例では、指定された DB インスタンスを停止します。

```
aws rds stop-db-instance \
  --db-instance-identifier test-instance
```

出力:

```
{
  "DBInstance": {
    "DBInstanceStatus": "stopping",
```

```
    ...some output truncated...
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StopDbInstance](#)」の「」を参照してください。

switchover-blue-green-deployment

次の例は、switchover-blue-green-deployment を使用する方法を説明しています。

AWS CLI

例 1: RDS DB インスタンスのブルー/グリーンデプロイを切り替えるには

次のswitchover-blue-green-deployment例では、指定されたグリーン環境を新しい本番環境として昇格させます。

```
aws rds switchover-blue-green-deployment \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
  --switchover-timeout 300
```

出力:

```
{
  "BlueGreenDeployment": {
    "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
    "BlueGreenDeploymentName": "bgd-cli-test-instance",
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-blhile",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-blhile",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-1",
```

```
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-k5fv7u",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ggsh8m",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-o2vwm0",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "COMPLETED"
    }
],
"Status": "SWITCHOVER_IN_PROGRESS",
"CreateTime": "2022-02-25T22:33:22.225000+00:00"
}
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[ブルー/グリーンデプロイの切り替え](#)」を参照してください。

例 2: Aurora MySQL DB クラスターのブルー/グリーンデプロイを昇格させるには

次のswitchover-blue-green-deployment例では、指定されたグリーン環境を新しい本番環境として昇格させます。

```
aws rds switchover-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \  
  --switchover-timeout 300
```

出力:

```
{  
  "BlueGreenDeployment": {  
    "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",  
    "BlueGreenDeploymentName": "my-blue-green-deployment",  
    "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-  
cluster",  
    "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-  
cluster-green-3ud8z6",  
    "SwitchoverDetails": [  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-  
aurora-mysql-cluster",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-  
aurora-mysql-cluster-green-3ud8z6",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-1",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-1-green-bvxc73",  
        "Status": "AVAILABLE"  
      },  
      {  
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-2",  
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-  
mysql-cluster-2-green-7wc4ie",  
        "Status": "AVAILABLE"  
      }  
    ]  
  }  
}
```

```
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:mysql-aurora-
mysql-cluster-3",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:mysql-aurora-
mysql-cluster-3-green-p4xxkz",
      "Status": "AVAILABLE"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-np1ik1",
      "Status": "AVAILABLE"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-miszlf",
      "Status": "AVAILABLE"
    }
  ],
  "Tasks": [
    {
      "Name": "CREATING_READ_REPLICA_OF_SOURCE",
      "Status": "COMPLETED"
    },
    {
      "Name": "DB_ENGINE_VERSION_UPGRADE",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_CUSTOM_ENDPOINTS",
      "Status": "COMPLETED"
    }
  ],
  "Status": "SWITCHOVER_IN_PROGRESS",
  "CreateTime": "2022-02-25T22:38:49.522000+00:00"
}
```

```
}

```

詳細については、「Amazon Aurora ユーザーガイド」の「[ブルー/グリーンデプロイの切り替え](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SwitchoverBlueGreenDeployment](#)」の「」を参照してください。

を使用した Amazon RDS Data Service の例 AWS CLI

次のコード例は、Amazon RDS Data Service AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-execute-statement

次のコード例は、batch-execute-statement を使用する方法を示しています。

AWS CLI

バッチSQLステートメントを実行するには

次のbatch-execute-statement例では、パラメータが設定されたデータの配列に対してバッチSQLステートメントを実行します。

```
aws rds-data batch-execute-statement \
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
  --database "mydb" \
  --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
```



```
--sql "insert into mytable values (:id, :val)" \  
--parameter-sets "[[{"name": \"id\", \"value\": {\"longValue\": 1}}, {\"name\":  
\"val\", \"value\": {\"stringValue\": \"ValueOne\"}},  
  [ {\"name\": \"id\", \"value\": {\"longValue\": 2}}, {\"name\": \"val\",  
\"value\": {\"stringValue\": \"ValueTwo\"}},  
  [ {\"name\": \"id\", \"value\": {\"longValue\": 3}}, {\"name\": \"val\",  
\"value\": {\"stringValue\": \"ValueThree\"}}]]]"
```

このコマンドでは何も出力されません。

詳細については、「[Amazon RDSユーザーガイド](#)」の「[Aurora Serverless APIのデータの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchExecuteStatement](#)」の「」を参照してください。

begin-transaction

次の例は、begin-transaction を使用する方法を説明しています。

AWS CLI

SQLトランザクションを開始するには

次のbegin-transaction例では、SQLトランザクションを開始します。

```
aws rds-data begin-transaction \  
--resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
--database "mydb" \  
--secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret"
```

出力:

```
{  
  "transactionId": "ABC1234567890xyz"  
}
```

詳細については、「[Amazon RDSユーザーガイド](#)」の「[Aurora Serverless APIのデータの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BeginTransaction](#)」の「」を参照してください。

commit-transaction

次のコード例は、commit-transaction を使用する方法を示しています。

AWS CLI

SQLトランザクションをコミットするには

次のcommit-transaction例では、指定されたSQLトランザクションを終了し、そのトランザクションの一部として行った変更をコミットします。

```
aws rds-data commit-transaction \  
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
  --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \  
  --transaction-id "ABC1234567890xyz"
```

出力:

```
{  
  "transactionStatus": "Transaction Committed"  
}
```

詳細については、「Amazon RDSユーザーガイド」の「[Aurora Serverless APIのデータの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CommitTransaction](#)」の「」を参照してください。

execute-statement

次のコード例は、execute-statement を使用する方法を示しています。

AWS CLI

例 1: トランザクションの一部であるSQLステートメントを実行するには

次のexecute-statement例では、トランザクションの一部であるSQLステートメントを実行します。

```
aws rds-data execute-statement \  
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
  --database "mydb" \  
  --statement "SELECT * FROM mytable"
```

```
--secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \  
--sql "update mytable set quantity=5 where id=201" \  
--transaction-id "ABC1234567890xyz"
```

出力:

```
{  
  "numberOfRecordsUpdated": 1  
}
```

例 2: パラメータを使用して SQL ステートメントを実行するには

次の `execute-statement` 例では、パラメータを使用して SQL ステートメントを実行します。

```
aws rds-data execute-statement \  
  --resource-arn "arn:aws:rds:us-east-1:123456789012:cluster:mydbcluster" \  
  --database "mydb" \  
  --secret-arn "arn:aws:secretsmanager:us-east-1:123456789012:secret:mysecret" \  
  --sql "insert into mytable values (:id, :val)" \  
  --parameters "[{\"name\": \"id\", \"value\": {\"longValue\": 1}}, {\"name\":  
  \"val\", \"value\": {\"stringValue\": \"value1\"}}]"
```

出力:

```
{  
  "numberOfRecordsUpdated": 1  
}
```

詳細については、「[Amazon RDS ユーザーガイド](#)」の「[Aurora Serverless API のデータの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス ExecuteStatement](#)」の「」を参照してください。

rollback-transaction

次のコード例は、`rollback-transaction` を使用する方法を示しています。

AWS CLI

SQL トランザクションをロールバックするには

次のrollback-transaction例では、指定されたSQLトランザクションをロールバックします。

```
aws rds-data rollback-transaction \  
  --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \  
  --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \  
  --transaction-id "ABC1234567890xyz"
```

出力:

```
{  
  "transactionStatus": "Rollback Complete"  
}
```

詳細については、「Amazon RDSユーザーガイド」の「[Aurora Serverless APIのデータの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RollbackTransaction](#)」の「」を参照してください。

を使用した Amazon RDS Performance Insights の例 AWS CLI

次のコード例は、Amazon RDS Performance Insights AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-dimension-keys

次の例は、describe-dimension-keys を使用する方法を説明しています。

AWS CLI

ディメンションキーを記述するには

この例では、すべての待機イベントの名前をリクエストします。データは、イベント名と、指定された期間におけるそれらのイベントの集計値によって要約されます。

コマンド:

```
aws pi describe-dimension-keys --service-type RDS --identifier db-LKCG0BK26374TPTDFX0IWVCP --start-time 1527026400 --end-time 1527080400 --metric db.load.avg --group-by '{"Group": "db.wait_event"}
```

出力:

```
{
  "AlignedEndTime": 1.5270804E9,
  "AlignedStartTime": 1.5270264E9,
  "Keys": [
    {
      "Dimensions": {"db.wait_event.name": "wait/synch/mutex/innodb/aurora_lock_thread_slot_futex"},
      "Total": 0.05906906851195666
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/io/aurora_redo_log_flush"},
      "Total": 0.015824722186149193
    },
    {
      "Dimensions": {"db.wait_event.name": "CPU"},
      "Total": 0.008014396230265477
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/io/aurora_respond_to_client"},
      "Total": 0.0036361612526204477
    },
    {
      "Dimensions": {"db.wait_event.name": "wait/io/table/sql/handler"},
      "Total": 0.0019108398419382965
    },
    {

```

```

        "Dimensions": {"db.wait_event.name": "wait/synch/cond/mysys/
my_thread_var::suspend"},
        "Total": 8.533847837782684E-4
    },
    {
        "Dimensions": {"db.wait_event.name": "wait/io/file/csv/data"},
        "Total": 6.864181956477376E-4
    },
    {
        "Dimensions": {"db.wait_event.name": "Unknown"},
        "Total": 3.895887056379051E-4
    },
    {
        "Dimensions": {"db.wait_event.name": "wait/synch/mutex/sql/
FILE_AS_TABLE::LOCK_shim_lists"},
        "Total": 3.710368625122906E-5
    },
    {
        "Dimensions": {"db.wait_event.name": "wait/lock/table/sql/handler"},
        "Total": 0
    }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDimensionKeys](#)」の「」を参照してください。

get-resource-metrics

次のコード例は、get-resource-metrics を使用する方法を示しています。

AWS CLI

リソースメトリクスを取得するには

この例では、db.wait_event デイメンショングループと、そのグループ内の db.wait_event.name デイメンションのデータポイントをリクエストします。レスポンスでは、関連するデータポイントは、リクエストされたデイメンション (db.wait_event.name) によってグループ化されます。

コマンド:

```
aws pi get-resource-metrics --service-type RDS --identifier db-LKCG0BK26374TPTDFX0IWVCP  
PM --start-time 1527026400 --end-time 1527080400 --period-  
in-seconds 300 --metric db.load.avg --metric-queries file://metric-queries.json
```

の引数は、JSON ファイルに `--metric-queries` 保存されます `metric-queries.json`。ファイルの内容は次のとおりです。

```
[  
  {  
    "Metric": "db.load.avg",  
    "GroupBy": {  
      "Group": "db.wait_event"  
    }  
  }  
]
```

出力:

```
{  
  "AlignedEndTime": 1.5270804E9,  
  "AlignedStartTime": 1.5270264E9,  
  "Identifier": "db-LKCG0BK26374TPTDFX0IWVCP",  
  "MetricList": [  
    {  
      "Key": {  
        "Metric": "db.load.avg"  
      },  
      "DataPoints": [  
        {  
          "Timestamp": 1527026700.0,  
          "Value": 1.3533333333333333  
        },  
        {  
          "Timestamp": 1527027000.0,  
          "Value": 0.88  
        },  
        <...remaining output omitted...>  
      ]  
    },  
    {  
      "Key": {  
        "Metric": "db.load.avg",
```

```
        "Dimensions": {
            "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
        }
    },
    "DataPoints": [
        {
            "Timestamp": 1527026700.0,
            "Value": 0.8566666666666667
        },
        {
            "Timestamp": 1527027000.0,
            "Value": 0.8633333333333333
        },
        <...remaining output omitted...>
    ],
    <...remaining output omitted...>
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceMetrics](#)」の「」を参照してください。

を使用した Amazon Redshift の例 AWS CLI

次のコード例は、Amazon Redshift AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-reserved-node-exchange

次の例は、accept-reserved-node-exchange を使用する方法を説明しています。

AWS CLI

予約済みノード交換を受け入れるには

次のaccept-reserved-node-exchange例では、DC1リザーブドノードとDC2リザーブドノードの交換を受け入れます。

```
aws redshift accept-reserved-node-exchange /  
  --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE /  
  --target-reserved-node-offering-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

出力:

```
{  
  "ExchangedReservedNode": {  
    "ReservedNodeId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",  
    "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",  
    "NodeType": "dc2.large",  
    "StartTime": "2019-12-06T21:17:26Z",  
    "Duration": 31536000,  
    "FixedPrice": 0.0,  
    "UsagePrice": 0.0,  
    "CurrencyCode": "USD",  
    "NodeCount": 1,  
    "State": "exchanging",  
    "OfferingType": "All Upfront",  
    "RecurringCharges": [  
      {  
        "RecurringChargeAmount": 0.0,  
        "RecurringChargeFrequency": "Hourly"  
      }  
    ],  
    "ReservedNodeOfferingType": "Regular"  
  }  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「を使用したリザーブドノードのアップグレード AWS CLI」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptReservedNodeExchange](#)」の「」を参照してください。

authorize-cluster-security-group-ingress

次のコード例は、authorize-cluster-security-group-ingress を使用する方法を示しています。

AWS CLI

EC2 セキュリティへのアクセスを許可する GroupThis の例は、名前付き Amazon EC2 セキュリティグループへのアクセスを許可します。コマンド：

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

CIDR rangeThis 例へのアクセスを許可すると、CIDR range.Command へのアクセスが許可されます。

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --cidrip 192.168.100.100/32
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AuthorizeClusterSecurityGroupIngress](#)」の「」を参照してください。

authorize-snapshot-access

次の例は、authorize-snapshot-access を使用する方法を説明しています。

AWS CLI

AWS アカウントが復元することを許可する SnapshotThis サンプルは、スナップショットを復元444455556666することを AWS アカウントに許可しますmy-snapshot-id。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift authorize-snapshot-access --snapshot-id my-snapshot-id --account-with-restore-access 444455556666
```

結果:

```
{
  "Snapshot": {
    "Status": "available",
    "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
    "EstimatedSecondsToCompletion": 0,
    "AvailabilityZone": "us-east-1a",
    "ClusterVersion": "1.0",
    "MasterUsername": "adminuser",
    "Encrypted": false,
    "OwnerAccount": "111122223333",
    "BackupProgressInMegabytes": 11.0,
    "ElapsedTimeInSeconds": 0,
    "DBName": "dev",
    "CurrentBackupRateInMegabytesPerSecond": 0.1534,
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "ActualIncrementalBackupSizeInMegabytes": 11.0,
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "mycluster",
    "TotalBackupSizeInMegabytes": 20.0,
    "Port": 5439,
    "NumberOfNodes": 2,
    "SnapshotIdentifier": "my-snapshot-id"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AuthorizeSnapshotAccess](#)」の「」を参照してください。

batch-delete-cluster-snapshots

次の例は、batch-delete-cluster-snapshots を使用する方法を説明しています。

AWS CLI

クラスタースナップショットのセットを削除するには

次のbatch-delete-cluster-snapshots例では、一連の手動クラスタースナップショットを削除します。

```
aws redshift batch-delete-cluster-snapshots \  
  --  
  identifiers SnapshotIdentifier=mycluster-2019-11-06-14-12 SnapshotIdentifier=mycluster-2019-
```

出力:

```
{  
  "Resources": [  
    "mycluster-2019-11-06-14-12",  
    "mycluster-2019-11-06-14-20"  
  ]  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift スナップショット」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDeleteClusterSnapshots](#)」の「」を参照してください。

batch-modify-cluster-snapshots

次のコード例は、batch-modify-cluster-snapshots を使用する方法を示しています。

AWS CLI

クラスタースナップショットのセットを変更するには

次のbatch-modify-cluster-snapshots例では、一連のクラスタースナップショットの設定を変更します。

```
aws redshift batch-modify-cluster-snapshots \  
  --snapshot-identifier-list mycluster-2019-11-06-16-31 mycluster-2019-11-06-16-32  
 \  
  --manual-snapshot-retention-period 30
```

出力:

```
{  
  "Resources": [  

```

```

    "mycluster-2019-11-06-16-31",
    "mycluster-2019-11-06-16-32"
  ],
  "Errors": [],
  "ResponseMetadata": {
    "RequestId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "x-amzn-requestid": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",
      "content-type": "text/xml",
      "content-length": "480",
      "date": "Sat, 07 Dec 2019 00:36:09 GMT",
      "connection": "keep-alive"
    },
    "RetryAttempts": 0
  }
}

```

詳細については、[「Amazon Redshift クラスター管理ガイド」](#)の「Amazon Redshift スナップショット」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchModifyClusterSnapshots](#)」の「」を参照してください。

cancel-resize

次の例は、cancel-resize を使用する方法を説明しています。

AWS CLI

クラスターのサイズ変更をキャンセルするには

次のcancel-resize例では、クラスターの従来のサイズ変更オペレーションをキャンセルします。

```

aws redshift cancel-resize \
  --cluster-identifier mycluster

```

出力:

```

{
  "TargetNodeType": "dc2.large",
  "TargetNumberOfNodes": 2,

```

```
"TargetClusterType": "multi-node",
"Status": "CANCELLING",
"ResizeType": "ClassicResize",
"TargetEncryptionType": "NONE"
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift でのクラスターのサイズ変更」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelResize](#)」の「」を参照してください。

copy-cluster-snapshot

次のコード例は、copy-cluster-snapshot を使用する方法を示しています。

AWS CLI

Get a Description of All Cluster Versions This の例は、すべてのクラスターバージョンの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift copy-cluster-snapshot --source-snapshot-identifier
cm:examplecluster-2013-01-22-19-27-58 --target-snapshot-identifier my-saved-
snapshot-copy
```

結果:

```
{
  "Snapshot": {
    "Status": "available",
    "SnapshotCreateTime": "2013-01-22T19:27:58.931Z",
    "AvailabilityZone": "us-east-1c",
    "ClusterVersion": "1.0",
    "MasterUsername": "adminuser",
    "DBName": "dev",
    "ClusterCreateTime": "2013-01-22T19:23:59.368Z",
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "examplecluster",
    "Port": 5439,
    "NumberOfNodes": "2",
    "SnapshotIdentifier": "my-saved-snapshot-copy"
  }
}
```

```
    },
    "ResponseMetadata": {
      "RequestId": "3b279691-64e3-11e2-bec0-17624ad140dd"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyClusterSnapshot](#)」の「」を参照してください。

create-cluster-parameter-group

次の例は、`create-cluster-parameter-group` を使用する方法を説明しています。

AWS CLI

クラスターパラメータの作成 GroupThis の例は、新しいクラスターパラメータグループを作成します。コマンド：

```
aws redshift create-cluster-parameter-group --parameter-group-name
myclusterparametergroup --parameter-group-family redshift-1.0 --description "My
first cluster parameter group"
```

結果:

```
{
  "ClusterParameterGroup": {
    "ParameterGroupFamily": "redshift-1.0",
    "Description": "My first cluster parameter group",
    "ParameterGroupName": "myclusterparametergroup"
  },
  "ResponseMetadata": {
    "RequestId": "739448f0-64cc-11e2-8f7d-3b939af52818"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateClusterParameterGroup](#)」の「」を参照してください。

create-cluster-security-group

次のコード例は、`create-cluster-security-group` を使用する方法を示しています。

AWS CLI

クラスターセキュリティ GroupThis の例を作成すると、新しいクラスターセキュリティグループが作成されます。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group"
```

結果:

```
{
  "create_cluster_security_group_response": {
    "create_cluster_security_group_result": {
      "cluster_security_group": {
        "description": "This is my cluster security group",
        "owner_id": "300454760768",
        "cluster_security_group_name": "mysecuritygroup",
        "ec2_security_groups": \[],
        "ip_ranges": \[]
      }
    },
    "response_metadata": {
      "request_id": "5df486a0-343a-11e2-b0d8-d15d0ef48549"
    }
  }
}
```

--output text オプションコマンドを使用して、同じ情報をテキスト形式で取得することもできます。

--output text オプションコマンド:

オプションコマンド:

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group" --output text
```

結果:

```
This is my cluster security group 300454760768 mysecuritygroup
```



```
a0c0bfab-343a-11e2-95d2-c3dc9fe8ab57
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateClusterSecurityGroup](#)」の「」を参照してください。

create-cluster-snapshot

次のコード例は、create-cluster-snapshot を使用する方法を示しています。

AWS CLI

クラスターの作成 SnapshotThis の例は、新しいクラスタースナップショットを作成します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift create-cluster-snapshot --cluster-identifier mycluster --snapshot-identifier my-snapshot-id
```

結果:

```
{
  "Snapshot": {
    "Status": "creating",
    "SnapshotCreateTime": "2013-01-22T22:20:33.548Z",
    "AvailabilityZone": "us-east-1a",
    "ClusterVersion": "1.0",
    "MasterUsername": "adminuser",
    "DBName": "dev",
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "mycluster",
    "Port": 5439,
    "NumberOfNodes": "2",
    "SnapshotIdentifier": "my-snapshot-id"
  },
  "ResponseMetadata": {
    "RequestId": "f024d1a5-64e1-11e2-88c5-53eb05787dfb"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateClusterSnapshot](#)」の「」を参照してください。

create-cluster-subnet-group

次のコード例は、create-cluster-subnet-group を使用する方法を示しています。

AWS CLI

クラスターサブネットの作成 GroupThis の例は、新しいクラスターサブネットグループを作成します。コマンド：

```
aws redshift create-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
--description "My subnet group" --subnet-ids subnet-763fdd1c
```

結果:

```
{
  "ClusterSubnetGroup": {
    "Subnets": [
      {
        "SubnetStatus": "Active",
        "SubnetIdentifier": "subnet-763fdd1c",
        "SubnetAvailabilityZone": {
          "Name": "us-east-1a"
        }
      }
    ],
    "VpcId": "vpc-7e3fdd14",
    "SubnetGroupStatus": "Complete",
    "Description": "My subnet group",
    "ClusterSubnetGroupName": "mysubnetgroup"
  },
  "ResponseMetadata": {
    "RequestId": "500b8ce2-698f-11e2-9790-fd67517fb6fd"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateClusterSubnetGroup](#)」の「」を参照してください。

create-cluster

次の例は、create-cluster を使用する方法を説明しています。

AWS CLI

最小でクラスターを作成する ParametersThis の例は、パラメータの最小限のセットでクラスターを作成します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster
```

結果:

```
{
  "Cluster": {
    "NodeType": "dw.hs1.xlarge",
    "ClusterVersion": "1.0",
    "PubliclyAccessible": "true",
    "MasterUsername": "adminuser",
    "ClusterParameterGroups": [
      {
        "ParameterApplyStatus": "in-sync",
        "ParameterGroupName": "default.redshift-1.0"
      }
    ],
    "ClusterSecurityGroups": [
      {
        "Status": "active",
        "ClusterSecurityGroupName": "default"
      }
    ],
    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": [],
    "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "creating",
    "ClusterIdentifier": "mycluster",
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {
      "MasterUserPassword": "\*****"
    }
  },
  "ResponseMetadata": {
    "RequestId": "7cf4bcfc-64dd-11e2-bea9-49e0ce183f07"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCluster](#)」の「」を参照してください。

create-event-subscription

次のコード例は、create-event-subscription を使用する方法を示しています。

AWS CLI

イベントの通知サブスクリプションを作成するには

次のcreate-event-subscription例では、イベント通知サブスクリプションを作成します。

```
aws redshift create-event-subscription \  
  --subscription-name mysubscription \  
  --sns-topic-arn arn:aws:sns:us-west-2:123456789012:MySNSTopic \  
  --source-type cluster \  
  --source-ids mycluster
```

出力:

```
{  
  "EventSubscription": {  
    "CustomerAwsId": "123456789012",  
    "CustSubscriptionId": "mysubscription",  
    "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNSTopic",  
    "Status": "active",  
    "SubscriptionCreationTime": "2019-12-09T20:05:19.365Z",  
    "SourceType": "cluster",  
    "SourceIdsList": [  
      "mycluster"  
    ],  
    "EventCategoriesList": [],  
    "Severity": "INFO",  
    "Enabled": true,  
    "Tags": []  
  }  
}
```

詳細については、[「Amazon Redshift クラスタ管理ガイド」の「Amazon Redshift イベント通知のサブスクリプション」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateEventSubscription](#)」の「」を参照してください。

create-hsm-client-certificate

次の例は、create-hsm-client-certificate を使用する方法を説明しています。

AWS CLI

HSMクライアント証明書を作成するには

次のcreate-hsm-client-certificate例では、クラスターが に接続するために使用できるHSMクライアント証明書を生成しますHSM。

```
aws redshift create-hsm-client-certificate \  
--hsm-client-certificate-identifier myhsmclientcert
```

出力:

```
{  
  "HsmClientCertificate": {  
    "HsmClientCertificateIdentifier": "myhsmclientcert",  
    "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----  
MIICiEXAMPLECQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC  
VVMxCzAJBgNVBAsTEXAMPLEEwDgYDVQQHEwDTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAsTC01BTSBDb25EXAMPLEIwEAYDVQQDEw1UZXR0Q21sYWMxHZA  
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb2EXAMPLETEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBEXAMPLEMRAwDgYD  
EXAMPLETZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBDb25z  
b2x1MRIwEAEXAMPLEEw1UZXR0Q21sYWMxHZAAdBgkqhkiG9w0BCQEWEG5vb251QGft  
YXpvbi5jb20wgZ8wDQYJKEXAMLEAQEBBQADgY0AMIGJAoGBAMak0dn+a4GmWIWJ  
21uUSfwfEvySwTc2XADZ4nB+BLYgVIk6EXAMPLE3G93vUEI03IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugEXAMPLEEzZswY6786m86gpE  
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEEXAMPLEEAtCu4  
nUHVvxYUEXAMPLEEh8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJI1J00zbhNYS5f6GEXAMPLE10ZxBHjJnyp3780D8uTs7fLvjx79LjStB  
NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",  
    "Tags": []  
  }  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift API アクセス許可リファレンス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateHsmClientCertificate](#)」の「」を参照してください。

create-hsm-configuration

次の例は、create-hsm-configuration を使用する方法を説明しています。

AWS CLI

HSM設定を作成するには

次のcreate-hsm-configuration例では、クラスターがハードウェアセキュリティモジュール () にデータベース暗号化キーを保存して使用するために必要な情報を含む指定されたHSM設定を作成しますHSM。

```
aws redshift create-hsm-configuration /  
  --hsm-configuration-identifier myhsmconnection  
  --description "My HSM connection"  
  --hsm-ip-address 192.0.2.09  
  --hsm-partition-name myhsmpartition /  
  --hsm-partition-password A1b2c3d4 /  
  --hsm-server-public-certificate myhsmclientcert
```

出力:

```
{  
  "HsmConfiguration": {  
    "HsmConfigurationIdentifier": "myhsmconnection",  
    "Description": "My HSM connection",  
    "HsmIpAddress": "192.0.2.09",  
    "HsmPartitionName": "myhsmpartition",  
    "Tags": []  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateHsmConfiguration](#)」の「」を参照してください。

create-snapshot-copy-grant

次の例は、create-snapshot-copy-grant を使用する方法を説明しています。

AWS CLI

スナップショットコピー許可を作成するには

次のcreate-snapshot-copy-grant例では、スナップショットコピー許可を作成し、コピー先の AWS リージョンでコピーされたスナップショットを暗号化します。

```
aws redshift create-snapshot-copy-grant \  
  --snapshot-copy-grant-name mysnapshotcopygrantname
```

出力:

```
{  
  "SnapshotCopyGrant": {  
    "SnapshotCopyGrantName": "mysnapshotcopygrantname",  
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/  
bPxRfih3yCo8nvbEXAMPLEKEY",  
    "Tags": []  
  }  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift データベース暗号化」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSnapshotCopyGrant](#)」の「」を参照してください。

create-snapshot-schedule

次のコード例は、create-snapshot-schedule を使用する方法を示しています。

AWS CLI

スナップショットスケジュールを作成するには

次のcreate-snapshot-schedule例では、指定された説明と 12 時間ごとのレートでスナップショットスケジュールを作成します。

```
aws redshift create-snapshot-schedule \  
  --schedule-definitions "rate(12 hours)" \  
  --schedule-identifier mysnapshotschedule \  
  --schedule-description "My schedule description"
```

出力:

```
{  
  "ScheduleDefinitions": [  
    "rate(12 hours)"  
  ],  
  "ScheduleIdentifier": "mysnapshotschedule",  
  "ScheduleDescription": "My schedule description",  
  "Tags": []  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「自動スナップショットスケジュール」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSnapshotSchedule](#)」の「」を参照してください。

create-tags

次のコード例は、create-tags を使用する方法を示しています。

AWS CLI

クラスターのタグを作成するには

次のcreate-tags例では、指定されたタグキーと値のペアを指定されたクラスターに追加します。

```
aws redshift create-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tags "Key"="mytags","Value"="tag1"
```

このコマンドでは、出力が生成されません。

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift でのリソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTags](#)」の「」を参照してください。

delete-cluster-parameter-group

次の例は、delete-cluster-parameter-group を使用する方法を説明しています。

AWS CLI

クラスターパラメータを削除する GroupThis の例は、クラスターパラメータグループを削除します。コマンド：

```
aws redshift delete-cluster-parameter-group --parameter-group-name
myclusterparametergroup
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteClusterParameterGroup](#)」の「」を参照してください。

delete-cluster-security-group

次の例は、delete-cluster-security-group を使用する方法を説明しています。

AWS CLI

クラスターセキュリティを削除する GroupThis の例は、クラスターセキュリティグループを削除します。コマンド：

```
aws redshift delete-cluster-security-group --cluster-security-group-name
mysecuritygroup
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteClusterSecurityGroup](#)」の「」を参照してください。

delete-cluster-snapshot

次のコード例は、delete-cluster-snapshot を使用する方法を示しています。

AWS CLI

クラスターを削除する SnapshotThis の例は、クラスタースナップショットを削除します。コマンド :

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteClusterSnapshot](#)」の「」を参照してください。

delete-cluster-subnet-group

次の例は、delete-cluster-subnet-group を使用する方法を説明しています。

AWS CLI

クラスターサブネットを削除する GroupThis 例でクラスターサブネットグループを削除します。コマンド :

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

結果:

```
{
  "ResponseMetadata": {
    "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteClusterSubnetGroup](#)」の「」を参照してください。

delete-cluster

次のコード例は、delete-cluster を使用する方法を示しています。

AWS CLI

最終クラスターがないクラスターを削除する SnapshotThis の例は、クラスターを削除し、データの削除を強制するため、最終クラスタースナップショットは作成されません。コマンド :

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-snapshot
```

クラスターを削除する。最終クラスターを許可する SnapshotThis の例はクラスターを削除しますが、最終クラスタースナップショットを指定します。コマンド：

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCluster](#)」の「」を参照してください。

delete-event-subscription

次の例は、delete-event-subscription を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを削除するには

次のdelete-event-subscription例では、指定されたイベント通知サブスクリプションを削除します。

```
aws redshift delete-event-subscription \  
  --subscription-name mysubscription
```

このコマンドでは、出力が生成されません。

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift イベント通知のサブスクリプション」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEventSubscription](#)」の「」を参照してください。

delete-hsm-client-certificate

次の例は、delete-hsm-client-certificate を使用する方法を説明しています。

AWS CLI

HSMクライアント証明書を削除するには

次のdelete-hsm-client-certificate例では、HSMクライアント証明書を削除します。

```
aws redshift delete-hsm-client-certificate \  
  --hsm-client-certificate-identifier myhsmClientcert
```

このコマンドでは、出力が生成されません。

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift API アクセス許可リファレンス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteHsmClientCertificate](#)」の「」を参照してください。

delete-hsm-configuration

次のコード例は、delete-hsm-configuration を使用する方法を示しています。

AWS CLI

HSM設定を削除するには

次のdelete-hsm-configuration例では、現在の AWS アカウントから指定されたHSM設定を削除します。

```
aws redshift delete-hsm-configuration \  
  --hsm-configuration-identifier myhsmconnection
```

このコマンドでは、出力が生成されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteHsmConfiguration](#)」の「」を参照してください。

delete-scheduled-action

次のコード例は、delete-scheduled-action を使用する方法を示しています。

AWS CLI

スケジュールされたアクションを削除するには

次のdelete-scheduled-action例では、指定されたスケジュールされたアクションを削除します。

```
aws redshift delete-scheduled-action \  
  --scheduled-action-name myscheduledaction
```

このコマンドでは、出力が生成されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteScheduledAction](#)」の「」を参照してください。

delete-snapshot-copy-grant

次のコード例は、delete-snapshot-copy-grant を使用する方法を示しています。

AWS CLI

スナップショットコピー許可を削除するには

次のdelete-snapshot-copy-grant例では、指定されたスナップショットコピー許可を削除します。

```
aws redshift delete-snapshot-copy-grant \  
  --snapshot-copy-grant-name mysnapshotcopygrantname
```

このコマンドでは、出力が生成されません。

詳細については、[「Amazon Redshift クラスター管理ガイド」](#)の「[Amazon Redshift データベース暗号化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSnapshotCopyGrant](#)」の「」を参照してください。

delete-snapshot-schedule

次のコード例は、delete-snapshot-schedule を使用する方法を示しています。

AWS CLI

スナップショットスケジュールを削除するには

次のdelete-snapshot-schedule例では、指定されたスナップショットスケジュールを削除します。スケジュールを削除する前に、クラスターの関連付けを解除する必要があります。

```
aws redshift delete-snapshot-schedule \  
  --snapshot-schedule-name mysnapshotschedule
```

```
--schedule-identifier mysnapshotschedule
```

このコマンドでは、出力が生成されません。

詳細については、[「Amazon Redshift クラスター管理ガイド」の「自動スナップショットスケジュール」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSnapshotSchedule](#)」の「」を参照してください。

delete-tags

次の例は、delete-tags を使用する方法を説明しています。

AWS CLI

クラスターからタグを削除するには

次のdelete-tags例では、指定されたクラスターから指定されたキー名のタグを削除します。

```
aws redshift delete-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tag-keys "clustertagkey" "clustertagvalue"
```

このコマンドでは、出力が生成されません。

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift でのリソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTags](#)」の「」を参照してください。

describe-account-attributes

次の例は、describe-account-attributes を使用する方法を説明しています。

AWS CLI

AWS アカウントの属性を記述するには

次のdescribe-account-attributes例では、呼び出し元の AWS アカウントにアタッチされた属性を表示します。

```
aws redshift describe-account-attributes
```

出力:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "max-defer-maintenance-duration",
      "AttributeValues": [
        {
          "AttributeValue": "45"
        }
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAccountAttributes](#)」の「」を参照してください。

describe-cluster-db-revisions

次のコード例は、describe-cluster-db-revisions を使用する方法を示しています。

AWS CLI

クラスターの DB リビジョンを記述するには

次のdescribe-cluster-db-revisions例は、指定されたクラスターのClusterDbRevisionオブジェクトの配列の詳細を示しています。

```
aws redshift describe-cluster-db-revisions \
  --cluster-identifier mycluster
```

出力:

```
{
  "ClusterDbRevisions": [
    {
      "ClusterIdentifier": "mycluster",
      "CurrentDatabaseRevision": "11420",
```

```

        "DatabaseRevisionReleaseDate": "2019-11-22T16:43:49.597Z",
        "RevisionTargets": []
    }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeClusterDbRevisions](#)」の「」を参照してください。

describe-cluster-parameter-groups

次の例は、describe-cluster-parameter-groups を使用する方法を説明しています。

AWS CLI

Get a Description of All Cluster Parameter Groups This の例は、アカウントのすべてのクラスターパラメータグループの説明を列ヘッダーとともに返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-cluster-parameter-groups
```

結果:

```

{
  "ParameterGroups": [
    {
      "ParameterGroupFamily": "redshift-1.0",
      "Description": "My first cluster parameter group",
      "ParameterGroupName": "myclusterparametergroup"
    }
  ],
  "ResponseMetadata": {
    "RequestId": "8ceb8f6f-64cc-11e2-bea9-49e0ce183f07"
  }
}

```

--output text オプションコマンドを使用して、同じ情報をテキスト形式で取得することもできます。

--output text オプションコマンド:

オプションコマンド:


```
aws redshift describe-cluster-parameter-groups --output text
```

結果:

```
redshift-1.0      My first cluster parameter group      myclusterparametergroup
RESPONSEMETADATA 9e665a36-64cc-11e2-8f7d-3b939af52818
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeClusterParameterGroups](#)」の「」を参照してください。

describe-cluster-parameters

次の例は、describe-cluster-parameters を使用する方法を説明しています。

AWS CLI

指定されたクラスターパラメータのパラメータを取得する GroupThis の例は、名前付きパラメータグループのパラメータを取得します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup
```

結果:

```
{
  "Parameters": [
    {
      "Description": "Sets the display format for date and time values.",
      "DataType": "string",
      "IsModifiable": true,
      "Source": "engine-default",
      "ParameterValue": "ISO, MDY",
      "ParameterName": "datestyle"
    },
    {
      "Description": "Sets the number of digits displayed for floating-point
values",
      "DataType": "integer",
      "IsModifiable": true,
      "AllowedValues": "-15-2",
      "Source": "engine-default",
```

```

    "ParameterValue": "0",
    "ParameterName": "extra_float_digits"
  },
  (...remaining output omitted...)
]
}

```

--output text オプションコマンドを使用して、同じ情報をテキスト形式で取得することもできます。

--output text オプションコマンド:

オプションコマンド:

```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup --output text
```

結果:

```

RESPONSEMETADATA    cdac40aa-64cc-11e2-9e70-918437dd236d
Sets the display format for date and time values.  string True    engine-default
ISO, MDY          datestyle
Sets the number of digits displayed for floating-point values  integer True
-15-2  engine-default 0      extra_float_digits
This parameter applies a user-defined label to a group of queries that are run
during the same session..  string True    engine-default  default query_group
require ssl for all databaseconnections  boolean True    true,false    engine-
default  false  require_ssl
Sets the schema search order for names that are not schema-qualified.  string
True    engine-default $user, public  search_path
Aborts any statement that takes over the specified number of milliseconds.  integer
True    engine-default 0      statement_timeout
wlm json configuration  string True    engine-default
\ [{"query_concurrency":5}]      wlm_json_configuration

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeClusterParameters](#)」の「」を参照してください。

describe-cluster-security-groups

次の例は、describe-cluster-security-groups を使用する方法を説明しています。

AWS CLI

Get a Description of All Cluster Security Groups This の例は、アカウントのすべてのクラスターセキュリティグループの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-cluster-security-groups
```

結果:

```
{
  "ClusterSecurityGroups": [
    {
      "OwnerId": "100447751468",
      "Description": "default",
      "ClusterSecurityGroupName": "default",
      "EC2SecurityGroups": \[],
      "IPRanges": [
        {
          "Status": "authorized",
          "CIDRIP": "0.0.0.0/0"
        }
      ]
    },
    {
      "OwnerId": "100447751468",
      "Description": "This is my cluster security group",
      "ClusterSecurityGroupName": "mysecuritygroup",
      "EC2SecurityGroups": \[],
      "IPRanges": \[]
    },
    (...remaining output omitted...)
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeClusterSecurityGroups](#)」の「」を参照してください。

describe-cluster-snapshots

次のコード例は、describe-cluster-snapshots を使用する方法を示しています。

AWS CLI

Get a Description of All Cluster Snapshots This の例は、アカウントのすべてのクラスタースナップショットの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-cluster-snapshots
```

結果:

```
{
  "Snapshots": [
    {
      "Status": "available",
      "SnapshotCreateTime": "2013-07-17T22:02:22.852Z",
      "EstimatedSecondsToCompletion": -1,
      "AvailabilityZone": "us-east-1a",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
      "Encrypted": false,
      "OwnerAccount": "111122223333",
      "BackupProgressInMegabytes": 20.0,
      "ElapsedTimeInSeconds": 0,
      "DBName": "dev",
      "CurrentBackupRateInMegabytesPerSecond": 0.0,
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "ActualIncrementalBackupSizeInMegabytes"; 20.0
      "SnapshotType": "automated",
      "NodeType": "dw.hs1.xlarge",
      "ClusterIdentifier": "mycluster",
      "Port": 5439,
      "TotalBackupSizeInMegabytes": 20.0,
      "NumberOfNodes": "2",
      "SnapshotIdentifier": "cm:mycluster-2013-01-22-22-04-18"
    },
    {
      "EstimatedSecondsToCompletion": 0,
      "OwnerAccount": "111122223333",
      "CurrentBackupRateInMegabytesPerSecond": 0.1534,
      "ActualIncrementalBackupSizeInMegabytes"; 11.0,
      "NumberOfNodes": "2",
      "Status": "available",
      "ClusterVersion": "1.0",
      "MasterUsername": "adminuser",
```

```
"AccountsWithRestoreAccess": [
  {
    "AccountID": "444455556666"
  } ],
"TotalBackupSizeInMegabytes": 20.0,
"DBName": "dev",
"BackupProgressInMegabytes": 11.0,
"ClusterCreateTime": "2013-01-22T21:59:29.559Z",
"ElapsedTimeInSeconds": 0,
"ClusterIdentifier": "mycluster",
"SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
"AvailabilityZone": "us-east-1a",
"NodeType": "dw.hs1.xlarge",
"Encrypted": false,
"SnapshotType": "manual",
"Port": 5439,
"SnapshotIdentifier": "my-snapshot-id"
} ]
}
(...remaining output omitted...)
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClusterSnapshots](#)」の「」を参照してください。

describe-cluster-subnet-groups

次のコード例は、describe-cluster-subnet-groups を使用する方法を示しています。

AWS CLI

Get a Description of All Cluster Subnet GroupsThis の例は、すべてのクラスターサブネットグループの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-cluster-subnet-groups
```

結果:

```
{
  "ClusterSubnetGroups": [
    {
      "Subnets": [
```

```
{
  "SubnetStatus": "Active",
  "SubnetIdentifier": "subnet-763fdd1c",
  "SubnetAvailabilityZone": {
    "Name": "us-east-1a"
  }
},
{
  "VpcId": "vpc-7e3fdd14",
  "SubnetGroupStatus": "Complete",
  "Description": "My subnet group",
  "ClusterSubnetGroupName": "mysubnetgroup"
}
],
"ResponseMetadata": {
  "RequestId": "37fa8c89-6990-11e2-8f75-ab4018764c77"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClusterSubnetGroups](#)」の「」を参照してください。

describe-cluster-tracks

次のコード例は、describe-cluster-tracks を使用する方法を示しています。

AWS CLI

クラスタトラックを記述するには

次のdescribe-cluster-tracks例では、使用可能なメンテナンストラックの詳細を表示します。

```
aws redshift describe-cluster-tracks \
  --maintenance-track-name current
```

出力:

```
{
  "MaintenanceTracks": [
    {
```

```

    "MaintenanceTrackName": "current",
    "DatabaseVersion": "1.0.11420",
    "UpdateTargets": [
      {
        "MaintenanceTrackName": "preview_features",
        "DatabaseVersion": "1.0.11746",
        "SupportedOperations": [
          {
            "OperationName": "restore-from-cluster-snapshot"
          }
        ]
      },
      {
        "MaintenanceTrackName": "trailing",
        "DatabaseVersion": "1.0.11116",
        "SupportedOperations": [
          {
            "OperationName": "restore-from-cluster-snapshot"
          },
          {
            "OperationName": "modify-cluster"
          }
        ]
      }
    ]
  }
]
}

```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「クラスターメンテナンストラックの選択」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClusterTracks](#)」の「」を参照してください。

describe-cluster-versions

次の例は、describe-cluster-versions を使用する方法を説明しています。

AWS CLI

Get a Description of All Cluster Versions This の例は、すべてのクラスターバージョンの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-cluster-versions
```

結果:

```
{
  "ClusterVersions": [
    {
      "ClusterVersion": "1.0",
      "Description": "Initial release",
      "ClusterParameterGroupFamily": "redshift-1.0"
    } ],
  "ResponseMetadata": {
    "RequestId": "16a53de3-64cc-11e2-bec0-17624ad140dd"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClusterVersions](#)」の「」を参照してください。

describe-clusters

次の例は、describe-clusters を使用する方法を説明しています。

AWS CLI

Get a Description of All Clusters This サンプルは、アカウントのすべてのクラスターの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド :

```
aws redshift describe-clusters
```

結果:

```
{
  "Clusters": [
    {
      "NodeType": "dw.hs1.xlarge",
      "Endpoint": {
        "Port": 5439,
        "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
      },
      "ClusterVersion": "1.0",
```



```
"PubliclyAccessible": "true",
"MasterUsername": "adminuser",
"ClusterParameterGroups": [
  {
    "ParameterApplyStatus": "in-sync",
    "ParameterGroupName": "default.redshift-1.0"
  } ],
"ClusterSecurityGroups": [
  {
    "Status": "active",
    "ClusterSecurityGroupName": "default"
  } ],
"AllowVersionUpgrade": true,
"VpcSecurityGroups": \[],
"AvailabilityZone": "us-east-1a",
"ClusterCreateTime": "2013-01-22T21:59:29.559Z",
"PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
"AutomatedSnapshotRetentionPeriod": 1,
"ClusterStatus": "available",
"ClusterIdentifier": "mycluster",
"DBName": "dev",
"NumberOfNodes": 2,
"PendingModifiedValues": {}
} ],
"ResponseMetadata": {
  "RequestId": "65b71cac-64df-11e2-8f5b-e90bd6c77476"
}
}
```

--output text オプションコマンドを使用して、同じ情報をテキスト形式で取得することもできます。

--output text オプションコマンド:

オプションコマンド:

```
aws redshift describe-clusters --output text
```

結果:

```
dw.hs1.xlarge      1.0      true      adminuser      True      us-east-1a
2013-01-22T21:59:29.559Z      sat:03:30-sat:04:00      1      available
mycluster          dev      2
```

```
ENDPOINT      5439      mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com
in-sync       default.redshift-1.0
active        default
PENDINGMODIFIEDVALUES
RESPONSEMETADATA 934281a8-64df-11e2-b07c-f7fbdd006c67
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeClusters](#)」の「」を参照してください。

describe-default-cluster-parameters

次の例は、describe-default-cluster-parameters を使用する方法を説明しています。

AWS CLI

Get a Description of Default Cluster ParametersThis の例は、redshift-1.0 ファミリーのデフォルトクラスターパラメータの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-default-cluster-parameters --parameter-group-family
redshift-1.0
```

結果:

```
{
  "DefaultClusterParameters": {
    "ParameterGroupFamily": "redshift-1.0",
    "Parameters": [
      {
        "Description": "Sets the display format for date and time values.",
        "DataType": "string",
        "IsModifiable": true,
        "Source": "engine-default",
        "ParameterValue": "ISO, MDY",
        "ParameterName": "datestyle"
      },
      {
        "Description": "Sets the number of digits displayed for floating-point
values",
        "DataType": "integer",
        "IsModifiable": true,
        "AllowedValues": "-15-2",
```

```

        "Source": "engine-default",
        "ParameterValue": "0",
        "ParameterName": "extra_float_digits"
    },
    (...remaining output omitted...)
]
}
}

```

有効なパラメータグループファミリーのリストを表示するには、`describe-cluster-parameter-groups` コマンドを使用します。

`describe-cluster-parameter-groups` コマンド。

コマンド。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDefaultClusterParameters](#)」の「」を参照してください。

describe-event-categories

次のコード例は、`describe-event-categories` を使用する方法を示しています。

AWS CLI

クラスターのイベントカテゴリを記述するには

次の `describe-event-categories` 例では、クラスターのイベントカテゴリの詳細を表示します。

```
aws redshift describe-event-categories \
  --source-type cluster
```

出力:

```
{
  "EventCategoriesMapList": [
    {
      "SourceType": "cluster",
      "Events": [
        {
          "EventId": "REDSHIFT-EVENT-2000",
          "EventCategories": [

```

```

        "management"
      ],
      "EventDescription": "Cluster <cluster name> created at <time in
UTC>.",
      "Severity": "INFO"
    },
    {
      "EventId": "REDSHIFT-EVENT-2001",
      "EventCategories": [
        "management"
      ],
      "EventDescription": "Cluster <cluster name> deleted at <time in
UTC>.",
      "Severity": "INFO"
    },
    {
      "EventId": "REDSHIFT-EVENT-3625",
      "EventCategories": [
        "monitoring"
      ],
      "EventDescription": "The cluster <cluster name> can't be resumed
with its previous elastic network interface <ENI id>. We will allocate a new
elastic network interface and associate it with the cluster node.",
      "Severity": "INFO"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEventCategories](#)」の「」を参照してください。

describe-event-subscriptions

次の例は、describe-event-subscriptions を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを記述するには

次のdescribe-event-subscriptions例では、指定されたサブスクリプションのイベント通知サブスクリプションを表示します。

```
aws redshift describe-event-subscriptions \  
--subscription-name mysubscription
```

出力:

```
{  
  "EventSubscriptionsList": [  
    {  
      "CustomerAwsId": "123456789012",  
      "CustSubscriptionId": "mysubscription",  
      "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",  
      "Status": "active",  
      "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",  
      "SourceIdsList": [],  
      "EventCategoriesList": [  
        "management"  
      ],  
      "Severity": "ERROR",  
      "Enabled": true,  
      "Tags": []  
    }  
  ]  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift イベント通知のサブスクリプション」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEventSubscriptions](#)」の「」を参照してください。

describe-events

次のコード例は、describe-events を使用する方法を示しています。

AWS CLI

すべてのイベントを記述この例ではすべてのイベントを返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-events
```

結果:

```
{
  "Events": [
    {
      "Date": "2013-01-22T19:17:03.640Z",
      "SourceIdentifier": "myclusterparametergroup",
      "Message": "Cluster parameter group myclusterparametergroup has been
created.",
      "SourceType": "cluster-parameter-group"
    } ],
  "ResponseMetadata": {
    "RequestId": "9f056111-64c9-11e2-9390-ff04f2c1e638"
  }
}
```

--output text オプションコマンドを使用して、同じ情報をテキスト形式で取得することもできます。

--output text オプションコマンド:

オプションコマンド:

```
aws redshift describe-events --output text
```

結果:

```
2013-01-22T19:17:03.640Z    myclusterparametergroup Cluster parameter group
myclusterparametergroup has been created.    cluster-parameter-group
RESPONSEMETADATA    8e5fe765-64c9-11e2-bce3-e56f52c50e17
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeEvents](#)」の「」を参照してください。

describe-hsm-client-certificates

次のコード例は、describe-hsm-client-certificates を使用する方法を示しています。

AWS CLI

HSMクライアント証明書を記述するには

次のdescribe-hsm-client-certificates例では、指定されたHSMクライアント証明書の詳細を表示します。

```
aws redshift describe-hsm-client-certificates \  
  --hsm-client-certificate-identifier myhsmclientcert
```

出力:

```
{  
  "HsmClientCertificates": [  
    {  
      "HsmClientCertificateIdentifier": "myhsmclientcert",  
      "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----\  
EXAMPLECAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMCMC  
VVMxCzAJBgNVBAEXAMPLERAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBA5TC01BTSBDb25zEXAMPLEwEAYDVQQDEw1UZXR0Q21sYWMxHzAd  
BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5jb20wHhEXAMPLEDI1MjA0EXAMPLEN  
EXAMPLE0MjA0NTIxWjCBiDELMAKGA1UEBhMCMCVVMxCzAJBgNVBAgTAldBMRAwDgYD  
VQQHEwdTZWF0dGEXAMPLEQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z  
b2x1MRIwEAYDVQQDEw1UZXR0Q21sEXAMPLEdBgkqhkiG9w0BCQEWEG5vb251QGFT  
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIEEXAMPLEMaK0dn+a4GmWIWJ  
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY67EXAMPLEE  
EXAMPLEZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUHVvXUntneD9EXAMPLE6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJI1J00zbhNYS5f6GuoEEXAMPLEEBHjJnyp3780D8uTs7fLvjx79LjSTb  
NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\  
n",  
      "Tags": []  
    }  
  ]  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift API アクセス許可リファレンス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeHsmClientCertificates](#)」の「」を参照してください。

describe-hsm-configurations

次のコード例は、describe-hsm-configurations を使用する方法を示しています。

AWS CLI

HSM設定を記述するには

次のdescribe-hsm-configurations例では、呼び出し元の AWS アカウントで使用可能な HSM設定の詳細を表示します。

```
aws redshift describe-hsm-configurations \  
  --hsm-configuration-identifier myhsmconnection
```

出力:

```
{  
  "HsmConfigurations": [  
    {  
      "HsmConfigurationIdentifier": "myhsmconnection",  
      "Description": "My HSM connection",  
      "HsmIpAddress": "192.0.2.09",  
      "HsmPartitionName": "myhsmpartition",  
      "Tags": []  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeHsmConfigurations](#)」の「」を参照してください。

describe-logging-status

次のコード例は、describe-logging-status を使用する方法を示しています。

AWS CLI

クラスターのログ記録ステータスを記述するには

次のdescribe-logging-status例は、クエリや接続試行などの情報がクラスターに対してログに記録されているかどうかを示しています。

```
aws redshift describe-logging-status \  
  --cluster-identifier mycluster
```


出力:

```
{
  "LoggingEnabled": false
}
```

詳細については、「Amazon Redshift クラスター管理ガイド」の「[データベース監査ログ記録](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLoggingStatus](#)」の「」を参照してください。

describe-node-configuration-options

次のコード例は、describe-node-configuration-options を使用する方法を示しています。

AWS CLI

ノード設定オプションを記述するには

次のdescribe-node-configuration-options例は、ノードタイプ、ノード数、指定されたクラスタースナップショットのディスク使用量など、可能なノード設定のプロパティを示しています。

```
aws redshift describe-node-configuration-options \
  --action-type restore-cluster \
  --snapshot-identifier rs:mycluster-2019-12-09-16-42-43
```

出力:

```
{
  "NodeConfigurationOptionList": [
    {
      "NodeType": "dc2.large",
      "NumberOfNodes": 2,
      "EstimatedDiskUtilizationPercent": 19.61
    },
    {
      "NodeType": "dc2.large",
      "NumberOfNodes": 4,
      "EstimatedDiskUtilizationPercent": 9.96
    },
  ],
}
```

```
{
  "NodeType": "ds2.xlarge",
  "NumberOfNodes": 2,
  "EstimatedDiskUtilizationPercent": 1.53
},
{
  "NodeType": "ds2.xlarge",
  "NumberOfNodes": 4,
  "EstimatedDiskUtilizationPercent": 0.78
}
]
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift リザーブドノードの購入」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeNodeConfigurationOptions](#)」の「」を参照してください。

describe-orderable-cluster-options

次の例は、describe-orderable-cluster-options を使用する方法を説明しています。

AWS CLI

すべての注文可能なクラスターの説明 OptionsThis の例は、注文可能なすべてのクラスターオプションの説明を返します。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift describe-orderable-cluster-options
```

結果:

```
{
  "OrderableClusterOptions": [
    {
      "NodeType": "dw.hs1.8xlarge",
      "AvailabilityZones": [
        { "Name": "us-east-1a" },
        { "Name": "us-east-1b" },
        { "Name": "us-east-1c" } ],
      "ClusterVersion": "1.0",
      "ClusterType": "multi-node"
    }
  ]
}
```

```

    },
    {
      "NodeType": "dw.hs1.xlarge",
      "AvailabilityZones": [
        { "Name": "us-east-1a" },
        { "Name": "us-east-1b" },
        { "Name": "us-east-1c" } ],
      "ClusterVersion": "1.0",
      "ClusterType": "multi-node"
    },
    {
      "NodeType": "dw.hs1.xlarge",
      "AvailabilityZones": [
        { "Name": "us-east-1a" },
        { "Name": "us-east-1b" },
        { "Name": "us-east-1c" } ],
      "ClusterVersion": "1.0",
      "ClusterType": "single-node"
    } ],
  "ResponseMetadata": {
    "RequestId": "f6000035-64cb-11e2-9135-ff82df53a51a"
  }
}

```

--output text オプションコマンドを使用して、同じ情報をテキスト形式で取得することもできます。

--output text オプションコマンド:

オプションコマンド:

```
aws redshift describe-orderable-cluster-options --output text
```

結果:

```

dw.hs1.8xlarge      1.0      multi-node
us-east-1a
us-east-1b
us-east-1c
dw.hs1.xlarge      1.0      multi-node
us-east-1a
us-east-1b
us-east-1c

```

```
dw.hs1.xlarge      1.0      single-node
us-east-1a
us-east-1b
us-east-1c
RESPONSEMETADATA  e648696b-64cb-11e2-bec0-17624ad140dd
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOrderableClusterOptions](#)」の「」を参照してください。

describe-reserved-node-offerings

次の例は、describe-reserved-node-offerings を使用する方法を説明しています。

AWS CLI

リザーブドノードの説明 OfferingsThis の例は、購入可能なリザーブドノードサービスをすべて示しています。コマンド：

```
aws redshift describe-reserved-node-offerings
```

結果:

```
{
  "ReservedNodeOfferings": [
    {
      "OfferingType": "Heavy Utilization",
      "FixedPrice": "",
      "NodeType": "dw.hs1.xlarge",
      "UsagePrice": "",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": "",
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "Duration": 31536000,
      "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
    },
    {
      "OfferingType": "Heavy Utilization",
      "FixedPrice": "",
      "NodeType": "dw.hs1.8xlarge",
      "UsagePrice": "",

```

```

    "RecurringCharges": [
      {
        "RecurringChargeAmount": "",
        "RecurringChargeFrequency": "Hourly"
      } ],
    "Duration": 31536000,
    "ReservedNodeOfferingId": "e5a2ff3b-352d-4a9c-ad7d-373c4cab5dd2"
  },
  ...remaining output omitted...
],
"ResponseMetadata": {
  "RequestId": "8b1a1a43-75ff-11e2-9666-e142fe91ddd1"
}
}

```

予約済みノードサービスを購入する場合は、有効な `purchase-reserved-node-offering` を使用して `ReservedNodeOfferingId` を呼び出すことができます。

`purchase-reserved-node-offering` は有効な `ReservedNodeOfferingId` を使用します。

は有効な `ReservedNodeOfferingId` を使用します。

`ReservedNodeOfferingId`.

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeReservedNodeOfferings](#)」の「」を参照してください。

describe-reserved-nodes

次の例は、`describe-reserved-nodes` を使用する方法を説明しています。

AWS CLI

「リザーブド」の説明 `NodesThis` 「リザーブド」の例は、購入したリザーブドノードオフリングを示しています。コマンド：

```
aws redshift describe-reserved-nodes
```

結果:

```
{
```

```

"ResponseMetadata": {
  "RequestId": "bc29ce2e-7600-11e2-9949-4b361e7420b7"
},
"ReservedNodes": [
  {
    "OfferingType": "Heavy Utilization",
    "FixedPrice": "",
    "NodeType": "dw.hs1.xlarge",
    "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
    "UsagePrice": "",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": "",
        "RecurringChargeFrequency": "Hourly"
      } ],
    "NodeCount": 1,
    "State": "payment-pending",
    "StartTime": "2013-02-13T17:08:39.051Z",
    "Duration": 31536000,
    "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeReservedNodes](#)」の「」を参照してください。

describe-resize

次のコード例は、describe-resize を使用する方法を示しています。

AWS CLI

ResizeThis 説明例は、クラスターの最新のサイズ変更について説明しています。リクエストは、.dw.hs1.8xlargeCommand タイプの 3 つのノードに対して行われました。

```
aws redshift describe-resize --cluster-identifier mycluster
```

結果:

```
{
  "Status": "NONE",
```

```
"TargetClusterType": "multi-node",
"TargetNodeType": "dw.hs1.8xlarge",
"ResponseMetadata": {
  "RequestId": "9f52b0b4-7733-11e2-aa9b-318b2909bd27"
},
"TargetNumberOfNodes": "3"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeResize](#)」の「」を参照してください。

describe-scheduled-actions

次の例は、describe-scheduled-actions を使用する方法を説明しています。

AWS CLI

スケジュールされたアクションを記述するには

次のdescribe-scheduled-actions例では、現在スケジュールされているアクションの詳細を表示します。

```
aws redshift describe-scheduled-actions
```

出力:

```
{
  "ScheduledActions": [
    {
      "ScheduledActionName": "resizecluster",
      "TargetAction": {
        "ResizeCluster": {
          "ClusterIdentifier": "mycluster",
          "NumberOfNodes": 4,
          "Classic": false
        }
      },
      "Schedule": "at(2019-12-10T00:07:00)",
      "IamRole": "arn:aws:iam:123456789012:role/myRedshiftRole",
      "State": "ACTIVE",
      "NextInvocations": [
```

```

        "2019-12-10T00:07:00Z"
      ]
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeScheduledActions](#)」の「」を参照してください。

describe-snapshot-copy-grants

次のコード例は、describe-snapshot-copy-grants を使用する方法を示しています。

AWS CLI

スナップショットコピー許可を記述するには

次のdescribe-snapshot-copy-grants例では、指定されたクラスタースナップショットコピー許可の詳細を表示します。

```

aws redshift describe-snapshot-copy-grants \
  --snapshot-copy-grant-name mysnapshotcopygrantname

```

出力:

```

{
  "SnapshotCopyGrants": [
    {
      "SnapshotCopyGrantName": "mysnapshotcopygrantname",
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/bPxRfih3yCo8nvnvEXAMPLEKEY",
      "Tags": []
    }
  ]
}

```

詳細については、「[Amazon Redshift クラスター管理ガイド](#)」の「[Amazon Redshift データベース暗号化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSnapshotCopyGrants](#)」の「」を参照してください。

describe-snapshot-schedules

次の例は、describe-snapshot-schedules を使用する方法を説明しています。

AWS CLI

スナップショットスケジュールを記述するには

次のdescribe-snapshot-schedules例では、指定されたクラスタースナップショットスケジュールの詳細を表示します。

```
aws redshift describe-snapshot-schedules \  
  --cluster-identifier mycluster \  
  --schedule-identifier mynapshotschedule
```

出力:

```
{  
  "SnapshotSchedules": [  
    {  
      "ScheduleDefinitions": [  
        "rate(12 hours)"  
      ],  
      "ScheduleIdentifier": "mynapshotschedule",  
      "ScheduleDescription": "My schedule description",  
      "Tags": [],  
      "AssociatedClusterCount": 1,  
      "AssociatedClusters": [  
        {  
          "ClusterIdentifier": "mycluster",  
          "ScheduleAssociationState": "ACTIVE"  
        }  
      ]  
    }  
  ]  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「自動スナップショットスケジュール」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSnapshotSchedules](#)」の「」を参照してください。

describe-storage

次の例は、describe-storage を使用する方法を説明しています。

AWS CLI

ストレージを記述するには

次のdescribe-storage例では、アカウントのバックアップストレージとプロビジョナルストレージサイズの詳細を表示します。

```
aws redshift describe-storage
```

出力:

```
{
  "TotalBackupSizeInMegaBytes": 193149.0,
  "TotalProvisionedStorageInMegaBytes": 655360.0
}
```

詳細については、「Amazon Redshift クラスター管理ガイド」の[「スナップショットストレージの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStorage](#)」の「」を参照してください。

describe-table-restore-status

次のコード例は、describe-table-restore-status を使用する方法を示しています。

AWS CLI

クラスタースナップショットからのテーブル復元リクエストのステータスを記述するには

次のdescribe-table-restore-status例では、指定されたクラスターに対して行われたテーブル復元リクエストの詳細を表示します。

```
aws redshift describe-table-restore-status /
  --cluster-identifier mycluster
```

出力:

```
{
  "TableRestoreStatusDetails": [
    {
      "TableRestoreRequestId": "z1116630-0e80-46f4-ba86-bd9670411ebd",
      "Status": "IN_PROGRESS",
      "RequestTime": "2019-12-27T18:22:12.257Z",
      "ClusterIdentifier": "mycluster",
      "SnapshotIdentifier": "mysnapshotid",
      "SourceDatabaseName": "dev",
      "SourceSchemaName": "public",
      "SourceTableName": "mytable",
      "TargetDatabaseName": "dev",
      "TargetSchemaName": "public",
      "NewTableName": "mytable-clone"
    }
  ]
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「スナップショットからテーブルを復元する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTableRestoreStatus](#)」の「」を参照してください。

describe-tags

次のコード例は、describe-tags を使用する方法を示しています。

AWS CLI

タグを記述するには

次のdescribe-tags例では、指定されたタグ名と値に関連付けられた、指定されたクラスターのリソースを表示します。

```
aws redshift describe-tags \
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
  --tag-keys clustertagkey \
  --tag-values clustertagvalue
```

出力:

```
{
  "TaggedResources": [
    {
      "Tag": {
        "Key": "clustertagkey",
        "Value": "clustertagvalue"
      },
      "ResourceName": "arn:aws:redshift:us-
west-2:123456789012:cluster:mycluster",
      "ResourceType": "cluster"
    }
  ]
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift でのリソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

disable-snapshot-copy

次のコード例は、`disable-snapshot-copy` を使用する方法を示しています。

AWS CLI

クラスターのスナップショットコピーを無効にするには

次の`disable-snapshot-copy`例では、指定されたクラスターのスナップショットの自動コピーを無効にします。

```
aws redshift disable-snapshot-copy \
  --cluster-identifier mycluster
```

出力:

```
{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "ClusterStatus": "available",
    "ClusterAvailabilityStatus": "Available",
```

```
"MasterUsername": "adminuser",
"DBName": "dev",
"Endpoint": {
  "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
  "Port": 5439
},
"ClusterCreateTime": "2019-12-05T18:44:36.991Z",
"AutomatedSnapshotRetentionPeriod": 3,
"ManualSnapshotRetentionPeriod": -1,
"ClusterSecurityGroups": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sh-i9b431cd",
    "Status": "active"
  }
],
"ClusterParameterGroups": [
  {
    "ParameterGroupName": "default.redshift-1.0",
    "ParameterApplyStatus": "in-sync"
  }
],
"ClusterSubnetGroupName": "default",
"VpcId": "vpc-b1fel7t9",
"AvailabilityZone": "us-west-2f",
"PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
"PendingModifiedValues": {
  "NodeType": "dc2.large",
  "NumberOfNodes": 2,
  "ClusterType": "multi-node"
},
"ClusterVersion": "1.0",
"AllowVersionUpgrade": true,
"NumberOfNodes": 4,
"PubliclyAccessible": false,
"Encrypted": false,
"Tags": [
  {
    "Key": "mytags",
    "Value": "tag1"
  }
],
"EnhancedVpcRouting": false,
"IamRoles": [
```

```
    {
      "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
      "ApplyStatus": "in-sync"
    }
  ],
  "MaintenanceTrackName": "current",
  "DeferredMaintenanceWindows": [],
  "ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
  "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
  "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
}
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「別の AWS リージョンへのスナップショットのコピー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableSnapshotCopy](#)」の「」を参照してください。

enable-snapshot-copy

次のコード例は、enable-snapshot-copy を使用する方法を示しています。

AWS CLI

クラスターのスナップショットコピーを有効にするには

次のenable-snapshot-copy例では、指定されたクラスターのスナップショットの自動コピーを有効にします。

```
aws redshift enable-snapshot-copy \
  --cluster-identifier mycluster \
  --destination-region us-west-1
```

出力:

```
{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "ClusterStatus": "available",
    "ClusterAvailabilityStatus": "Available",
    "MasterUsername": "adminuser",
```

```
"DBName": "dev",
"Endpoint": {
  "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
  "Port": 5439
},
"ClusterCreateTime": "2019-12-05T18:44:36.991Z",
"AutomatedSnapshotRetentionPeriod": 3,
"ManualSnapshotRetentionPeriod": -1,
"ClusterSecurityGroups": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sh-f4c731cd",
    "Status": "active"
  }
],
"ClusterParameterGroups": [
  {
    "ParameterGroupName": "default.redshift-1.0",
    "ParameterApplyStatus": "in-sync"
  }
],
"ClusterSubnetGroupName": "default",
"VpcId": "vpc-b1ael7t9",
"AvailabilityZone": "us-west-2f",
"PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
"PendingModifiedValues": {
  "NodeType": "dc2.large",
  "NumberOfNodes": 2,
  "ClusterType": "multi-node"
},
"ClusterVersion": "1.0",
"AllowVersionUpgrade": true,
"NumberOfNodes": 4,
"PubliclyAccessible": false,
"Encrypted": false,
"ClusterSnapshotCopyStatus": {
  "DestinationRegion": "us-west-1",
  "RetentionPeriod": 7,
  "ManualSnapshotRetentionPeriod": -1
},
"Tags": [
  {
    "Key": "mytags",
    "Value": "tag1"
  }
]
```

```
    }
  ],
  "EnhancedVpcRouting": false,
  "IamRoles": [
    {
      "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
      "ApplyStatus": "in-sync"
    }
  ],
  "MaintenanceTrackName": "current",
  "DeferredMaintenanceWindows": [],
  "ExpectedNextSnapshotScheduleTime": "2019-12-10T04:42:43.390Z",
  "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
  "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
}
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「別の AWS リージョンへのスナップショットのコピー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableSnapshotCopy](#)」の「」を参照してください。

get-cluster-credentials

次の例は、get-cluster-credentials を使用する方法を説明しています。

AWS CLI

AWS アカウントのクラスター認証情報を取得するには

次のget-cluster-credentials例では、Amazon Redshift データベースへのアクセスを有効にする一時的な認証情報を取得します。

```
aws redshift get-cluster-credentials \  
  --db-user adminuser --db-name dev \  
  --cluster-identifier mycluster
```

出力:

```
{  
  "DbUser": "IAM:adminuser",
```



```
"DbPassword": "AMAFUyyuros/QjxPTtgzcsuQsqzIasdzJEN04aCtWDzXx109d6UmpkBtvEqFly/  
EXAMPLE==",  
  "Expiration": "2019-12-10T17:25:05.770Z"  
}
```

詳細については、[「Amazon Redshift を使用したIAMデータベース認証情報の生成CLI」](#)または [API「Amazon Redshift クラスター管理ガイド」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetClusterCredentials](#)」の「」を参照してください。

get-reserved-node-exchange-offerings

次のコード例は、get-reserved-node-exchange-offerings を使用する方法を示しています。

AWS CLI

予約済みノード交換サービスを取得するには

次のget-reserved-node-exchange-offerings例では、指定されたDC1リザーブドノードDC2ReservedNodeOfferingsに一致する の配列を取得します。

```
aws redshift get-reserved-node-exchange-offerings \  
  --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

出力:

```
{  
  "ReservedNodeOfferings": [  
    {  
      "ReservedNodeOfferingId": "12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE",  
      "NodeType": "dc2.large",  
      "Duration": 31536000,  
      "FixedPrice": 0.0,  
      "UsagePrice": 0.0,  
      "CurrencyCode": "USD",  
      "OfferingType": "All Upfront",  
      "RecurringCharges": [  
        {  
          "RecurringChargeAmount": 0.0,  
          "RecurringChargeFrequency": "Hourly"  
        }  
      ],  
    }  
  ],  
}
```

```

        "ReservedNodeOfferingType": "Regular"
    }
]
}

```

詳細については、「[Amazon Redshift クラスター管理ガイド](#)」の「[を使用したリザーブドノードのアップグレード AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetReservedNodeExchangeOfferings](#)」の「」を参照してください。

modify-cluster-iam-roles

次のコード例は、`modify-cluster-iam-roles` を使用する方法を示しています。

AWS CLI

クラスターのIAMロールを変更するには

次の`modify-cluster-iam-roles`例では、指定された AWS IAMクラスターから指定されたロールを削除します。

```

aws redshift modify-cluster-iam-roles \
  --cluster-identifier mycluster \
  --remove-iam-roles arn:aws:iam::123456789012:role/myRedshiftRole

```

出力:

```

{
  "Cluster": {
    "ClusterIdentifier": "mycluster",
    "NodeType": "dc2.large",
    "ClusterStatus": "available",
    "ClusterAvailabilityStatus": "Available",
    "MasterUsername": "adminuser",
    "DBName": "dev",
    "Endpoint": {
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",
      "Port": 5439
    },
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",
    "AutomatedSnapshotRetentionPeriod": 3,
    "ManualSnapshotRetentionPeriod": -1,
  }
}

```

```
"ClusterSecurityGroups": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sh-f9b731sd",
    "Status": "active"
  }
],
"ClusterParameterGroups": [
  {
    "ParameterGroupName": "default.redshift-1.0",
    "ParameterApplyStatus": "in-sync"
  }
],
"ClusterSubnetGroupName": "default",
"VpcId": "vpc-b2fal7t9",
"AvailabilityZone": "us-west-2f",
"PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
"PendingModifiedValues": {
  "NodeType": "dc2.large",
  "NumberOfNodes": 2,
  "ClusterType": "multi-node"
},
"ClusterVersion": "1.0",
"AllowVersionUpgrade": true,
"NumberOfNodes": 4,
"PubliclyAccessible": false,
"Encrypted": false,
"ClusterSnapshotCopyStatus": {
  "DestinationRegion": "us-west-1",
  "RetentionPeriod": 7,
  "ManualSnapshotRetentionPeriod": -1
},
"Tags": [
  {
    "Key": "mytags",
    "Value": "tag1"
  }
],
"EnhancedVpcRouting": false,
"IamRoles": [],
"MaintenanceTrackName": "current",
"DeferredMaintenanceWindows": [],
"ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
"ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
```

```
    "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
  }
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift でのアイデンティティベースのポリシー \(IAM ポリシー\) の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyClusterIamRoles](#)」の「」を参照してください。

modify-cluster-maintenance

次のコード例は、`modify-cluster-maintenance` を使用する方法を示しています。

AWS CLI

クラスターのメンテナンスを変更するには

次の`modify-cluster-maintenance`例では、指定されたクラスターのメンテナンスを 30 日間延期します。

```
aws redshift modify-cluster-maintenance \  
  --cluster-identifier mycluster \  
  --defer-maintenance \  
  --defer-maintenance-duration 30
```

出力:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "available",  
    "ClusterAvailabilityStatus": "Available",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
  },  
}
```

```
"ClusterSecurityGroups": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sh-a1a123ab",
    "Status": "active"
  }
],
"ClusterParameterGroups": [
  {
    "ParameterGroupName": "default.redshift-1.0",
    "ParameterApplyStatus": "in-sync"
  }
],
"ClusterSubnetGroupName": "default",
"VpcId": "vpc-b1ael7t9",
"AvailabilityZone": "us-west-2f",
"PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
"PendingModifiedValues": {
  "NodeType": "dc2.large",
  "NumberOfNodes": 2,
  "ClusterType": "multi-node"
},
"ClusterVersion": "1.0",
"AllowVersionUpgrade": true,
"NumberOfNodes": 4,
"PubliclyAccessible": false,
"Encrypted": false,
"ClusterSnapshotCopyStatus": {
  "DestinationRegion": "us-west-1",
  "RetentionPeriod": 7,
  "ManualSnapshotRetentionPeriod": -1
},
"Tags": [
  {
    "Key": "mytags",
    "Value": "tag1"
  }
],
"EnhancedVpcRouting": false,
"IamRoles": [],
"MaintenanceTrackName": "current",
"DeferredMaintenanceWindows": [
  {
    "DeferMaintenanceIdentifier": "dfm-mUdVIIFcT1B4SGhw6fyF",
```

```

        "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
        "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
    }
],
"ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
"ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
"NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
}
}

```

詳細については、「Amazon Redshift [クラスター管理ガイド](#)」の「[クラスターメンテナンス](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスModifyClusterMaintenance](#)」の「」を参照してください。

modify-cluster-parameter-group

次の例は、`modify-cluster-parameter-group` を使用する方法を説明しています。

AWS CLI

パラメータグループのパラメータを変更する

次の`modify-cluster-parameter-group`例では、ワークロード管理の `wlm_json_configuration` パラメータを変更します。以下に示すJSON内容を含むファイルからパラメータを受け入れます。

```

aws redshift modify-cluster-parameter-group \
  --parameter-group-name myclusterparametergroup \
  --parameters file://modify_pg.json

```

`modify_pg.json` の内容:

```

[
  {
    "ParameterName": "wlm_json_configuration",
    "ParameterValue": "[{\"user_group\": \"example_user_group1\", \"query_group\": \"example_query_group1\", \"query_concurrency\": 7}, {\"query_concurrency\": 5}]"
  }
]

```

出力:

```
{
  "ParameterGroupStatus": "Your parameter group has been updated but changes won't
  get applied until you reboot the associated Clusters.",
  "ParameterGroupName": "myclusterparametergroup",
  "ResponseMetadata": {
    "RequestId": "09974cc0-64cd-11e2-bea9-49e0ce183f07"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyClusterParameterGroup](#)」の「」を参照してください。

modify-cluster-snapshot-schedule

次の例は、`modify-cluster-snapshot-schedule` を使用する方法を説明しています。

AWS CLI

クラスタースナップショットのスケジュールを変更するには

次の`modify-cluster-snapshot-schedule`例では、指定されたクラスターから指定されたスナップショットスケジュールを削除します。

```
aws redshift modify-cluster-snapshot-schedule \
  --cluster-identifier mycluster \
  --schedule-identifier mynapsshotschedule \
  --disassociate-schedule
```

このコマンドでは、出力が生成されません。

詳細については、[「Amazon Redshift クラスター管理ガイド」の「自動スナップショットスケジュール」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyClusterSnapshotSchedule](#)」の「」を参照してください。

modify-cluster-snapshot

次の例は、`modify-cluster-snapshot` を使用する方法を説明しています。

AWS CLI

クラスタースナップショットを変更するには

次のmodify-cluster-snapshot例では、指定されたクラスタースナップショットの手動保持期間設定を 10 日の値に設定します。

```
aws redshift modify-cluster-snapshot \  
--snapshot-identifier mycluster-2019-11-06-16-32 \  
--manual-snapshot-retention-period 10
```

出力:

```
{  
  "Snapshot": {  
    "SnapshotIdentifier": "mycluster-2019-11-06-16-32",  
    "ClusterIdentifier": "mycluster",  
    "SnapshotCreateTime": "2019-12-07T00:34:05.633Z",  
    "Status": "available",  
    "Port": 5439,  
    "AvailabilityZone": "us-west-2f",  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "MasterUsername": "adminuser",  
    "ClusterVersion": "1.0",  
    "SnapshotType": "manual",  
    "NodeType": "dc2.large",  
    "NumberOfNodes": 2,  
    "DBName": "dev",  
    "VpcId": "vpc-b1cel7t9",  
    "Encrypted": false,  
    "EncryptedWithHSM": false,  
    "OwnerAccount": "123456789012",  
    "TotalBackupSizeInMegaBytes": 64384.0,  
    "ActualIncrementalBackupSizeInMegaBytes": 24.0,  
    "BackupProgressInMegaBytes": 24.0,  
    "CurrentBackupRateInMegaBytesPerSecond": 13.0011,  
    "EstimatedSecondsToCompletion": 0,  
    "ElapsedTimeInSeconds": 1,  
    "Tags": [  
      {  
        "Key": "mytagkey",  
        "Value": "mytagvalue"  
      }  
    ]  
  }  
}
```



```
    ],  
    "EnhancedVpcRouting": false,  
    "MaintenanceTrackName": "current",  
    "ManualSnapshotRetentionPeriod": 10,  
    "ManualSnapshotRemainingDays": 6,  
    "SnapshotRetentionStartTime": "2019-12-07T00:34:07.479Z"  
  }  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift スナップショット」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyClusterSnapshot](#)」の「」を参照してください。

modify-cluster-subnet-group

次のコード例は、`modify-cluster-subnet-group` を使用する方法を示しています。

AWS CLI

クラスターサブネット内のサブネットを変更する GroupThis 例には、キャッシュサブネットグループ内のサブネットのリストを変更する方法が示されています。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift modify-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup  
--subnet-ids subnet-763fdd1 subnet-ac830e9
```

結果:

```
{  
  "ClusterSubnetGroup":  
  {  
    "Subnets": [  
      {  
        "SubnetStatus": "Active",  
        "SubnetIdentifier": "subnet-763fdd1c",  
        "SubnetAvailabilityZone":  
          { "Name": "us-east-1a" }  
      },  
      {
```

```
    "SubnetStatus": "Active",
    "SubnetIdentifier": "subnet-ac830e9",
    "SubnetAvailabilityZone":
      { "Name": "us-east-1b" }
  } ],
  "VpcId": "vpc-7e3fdd14",
  "SubnetGroupStatus": "Complete",
  "Description": "My subnet group",
  "ClusterSubnetGroupName": "mysubnetgroup"
},
"ResponseMetadata": {
  "RequestId": "8da93e89-8372-f936-93a8-873918938197a"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyClusterSubnetGroup](#)」の「」を参照してください。

modify-cluster

次の例は、modify-cluster を使用する方法を説明しています。

AWS CLI

セキュリティグループと を関連付ける ClusterThis の例は、クラスターセキュリティグループを指定されたクラスターに関連付ける方法を示しています。コマンド：

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups
mysecuritygroup
```

のメンテナンスウィンドウを変更すると、クラスターの毎週の優先メンテナンスウィンドウを、日曜日の午後 11 時 15 分から月曜日の午前 3 時 15 分まで 4 時間以上のウィンドウに変更する方法 ClusterThis が表示されます。コマンド：

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-
window Sun:23:15-Mon:03:15
```

この例では、クラスターのマスターパスワードを変更する方法 ClusterThis を示しています。コマンド：

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password
A1b2c3d4
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyCluster](#)」の「」を参照してください。

modify-event-subscription

次の例は、`modify-event-subscription` を使用する方法を説明しています。

AWS CLI

イベントサブスクリプションを変更するには

次の`modify-event-subscription`例では、指定されたイベント通知サブスクリプションを無効にします。

```
aws redshift modify-event-subscription \  
  --subscription-name mysubscription \  
  --no-enabled
```

出力:

```
{  
  "EventSubscription": {  
    "CustomerAwsId": "123456789012",  
    "CustSubscriptionId": "mysubscription",  
    "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",  
    "Status": "active",  
    "SubscriptionCreationTime": "2019-12-09T21:50:21.332Z",  
    "SourceIdsList": [],  
    "EventCategoriesList": [  
      "management"  
    ],  
    "Severity": "ERROR",  
    "Enabled": false,  
    "Tags": []  
  }  
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift イベント通知のサブスクリプション」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyEventSubscription](#)」の「」を参照してください。

modify-scheduled-action

次のコード例は、`modify-scheduled-action` を使用する方法を示しています。

AWS CLI

スケジュールされたアクションを変更するには

次の`modify-scheduled-action`例では、指定された既存のスケジュールされたアクションに説明を追加します。

```
aws redshift modify-scheduled-action \  
  --scheduled-action-name myscheduledaction \  
  --scheduled-action-description "My scheduled action"
```

出力:

```
{  
  "ScheduledActionName": "myscheduledaction",  
  "TargetAction": {  
    "ResizeCluster": {  
      "ClusterIdentifier": "mycluster",  
      "NumberOfNodes": 2,  
      "Classic": false  
    }  
  },  
  "Schedule": "at(2019-12-25T00:00:00)",  
  "IamRole": "arn:aws:iam::123456789012:role/myRedshiftRole",  
  "ScheduledActionDescription": "My scheduled action",  
  "State": "ACTIVE",  
  "NextInvocations": [  
    "2019-12-25T00:00:00Z"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ModifyScheduledAction](#)」の「」を参照してください。

modify-snapshot-copy-retention-period

次のコード例は、modify-snapshot-copy-retention-period を使用する方法を示しています。

AWS CLI

スナップショットコピーの保持期間を変更するには

次のmodify-snapshot-copy-retention-period例では、送信元 AWS リージョンからコピーされた後、送信先 AWS リージョンに指定されたクラスターのスナップショットを保持する日数を変更します。

```
aws redshift modify-snapshot-copy-retention-period \  
  --cluster-identifier mycluster \  
  --retention-period 15
```

出力:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "available",  
    "ClusterAvailabilityStatus": "Available",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswquae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-a1a123ab",  
        "Status": "active"  
      }  
    ]  
  }  
}
```

```
    }
  ],
  "ClusterParameterGroups": [
    {
      "ParameterGroupName": "default.redshift-1.0",
      "ParameterApplyStatus": "in-sync"
    }
  ],
  "ClusterSubnetGroupName": "default",
  "VpcId": "vpc-b1fet7t9",
  "AvailabilityZone": "us-west-2f",
  "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
  "PendingModifiedValues": {
    "NodeType": "dc2.large",
    "NumberOfNodes": 2,
    "ClusterType": "multi-node"
  },
  "ClusterVersion": "1.0",
  "AllowVersionUpgrade": true,
  "NumberOfNodes": 4,
  "PubliclyAccessible": false,
  "Encrypted": false,
  "ClusterSnapshotCopyStatus": {
    "DestinationRegion": "us-west-1",
    "RetentionPeriod": 15,
    "ManualSnapshotRetentionPeriod": -1
  },
  "Tags": [
    {
      "Key": "mytags",
      "Value": "tag1"
    }
  ],
  "EnhancedVpcRouting": false,
  "IamRoles": [],
  "MaintenanceTrackName": "current",
  "DeferredMaintenanceWindows": [
    {
      "DeferMaintenanceIdentifier": "dfm-mUdVSfDcT1F4SGhw6fyF",
      "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
      "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
    }
  ],
  "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
```

```
}  
}
```

詳細については、「Amazon Redshift クラスター管理ガイド」の[「スナップショットスケジュール形式」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifySnapshotCopyRetentionPeriod](#)」の「」を参照してください。

modify-snapshot-schedule

次のコード例は、modify-snapshot-schedule を使用する方法を示しています。

AWS CLI

スナップショットのスケジュールを変更するには

次のmodify-snapshot-schedule例では、指定されたスナップショットスケジュールのレートを 10 時間ごとに変更します。

```
aws redshift modify-snapshot-schedule \  
  --schedule-identifier mynapshotschedule \  
  --schedule-definitions "rate(10 hours)"
```

出力:

```
{  
  "ScheduleDefinitions": [  
    "rate(10 hours)"  
  ],  
  "ScheduleIdentifier": "mynapshotschedule",  
  "ScheduleDescription": "My schedule description",  
  "Tags": []  
}
```

詳細については、「Amazon Redshift クラスター管理ガイド」の[「スナップショットスケジュール形式」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifySnapshotSchedule](#)」の「」を参照してください。

purchase-reserved-node-offering

次の例は、purchase-reserved-node-offering を使用する方法を説明しています。

AWS CLI

リザーブド購入 NodeThis の例は、リザーブドノードサービスを購入する方法を示しています。reserved-node-offering-id は、describe-reserved-node-offerings.Command を呼び出して取得します。

```
aws redshift purchase-reserved-node-offering --reserved-node-offering-id ceb6a579-cf4c-4343-be8b-d832c45ab51c
```

結果:

```
{
  "ReservedNode": {
    "OfferingType": "Heavy Utilization",
    "FixedPrice": "",
    "NodeType": "dw.hs1.xlarge",
    "ReservedNodeId": "1ba8e2e3-bc01-4d65-b35d-a4a3e931547e",
    "UsagePrice": "",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": "",
        "RecurringChargeFrequency": "Hourly"
      }
    ],
    "NodeCount": 1,
    "State": "payment-pending",
    "StartTime": "2013-02-13T17:08:39.051Z",
    "Duration": 31536000,
    "ReservedNodeOfferingId": "ceb6a579-cf4c-4343-be8b-d832c45ab51c"
  },
  "ResponseMetadata": {
    "RequestId": "01bda7bf-7600-11e2-b605-2568d7396e7f"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PurchaseReservedNodeOffering](#)」の「」を参照してください。

reboot-cluster

次の例は、reboot-cluster を使用する方法を説明しています。

AWS CLI

サンプルを再起動すると、クラスターが ClusterThis 再起動されます。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift reboot-cluster --cluster-identifier mycluster
```

結果:

```
{
  "Cluster": {
    "NodeType": "dw.hs1.xlarge",
    "Endpoint": {
      "Port": 5439,
      "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
    },
    "ClusterVersion": "1.0",
    "PubliclyAccessible": "true",
    "MasterUsername": "adminuser",
    "ClusterParameterGroups": [
      {
        "ParameterApplyStatus": "in-sync",
        "ParameterGroupName": "default.redshift-1.0"
      }
    ],
    "ClusterSecurityGroups": [
      {
        "Status": "active",
        "ClusterSecurityGroupName": "default"
      }
    ],
    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": \[],
    "AvailabilityZone": "us-east-1a",
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "rebooting",
    "ClusterIdentifier": "mycluster",
```

```
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {}
  },
  "ResponseMetadata": {
    "RequestId": "61c8b564-64e8-11e2-8f7d-3b939af52818"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootCluster](#)」の「」を参照してください。

reset-cluster-parameter-group

次のコード例は、reset-cluster-parameter-group を使用する方法を示しています。

AWS CLI

パラメータのパラメータのリセット GroupThis の例は、パラメータグループのすべてのパラメータをリセットする方法を示しています。コマンド：

```
aws redshift reset-cluster-parameter-group --parameter-group-name
myclusterparametergroup --reset-all-parameters
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetClusterParameterGroup](#)」の「」を参照してください。

resize-cluster

次の例は、resize-cluster を使用する方法を説明しています。

AWS CLI

クラスターのサイズを変更するには

次のresize-cluster例では、指定されたクラスターのサイズを変更します。

```
aws redshift resize-cluster \  
  --cluster-identifier mycluster \  
  --cluster-type multi-node \  
  --node-type dc2.large \  
  --node-count 3
```

```
--number-of-nodes 6 \  
--classic
```

出力:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "resizing",  
    "ClusterAvailabilityStatus": "Modifying",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-a1a123ab",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {  
        "ParameterGroupName": "default.redshift-1.0",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "ClusterSubnetGroupName": "default",  
    "VpcId": "vpc-a1abc1a1",  
    "AvailabilityZone": "us-west-2f",  
    "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",  
    "PendingModifiedValues": {  
      "NodeType": "dc2.large",  
      "NumberOfNodes": 6,  
      "ClusterType": "multi-node"  
    },  
    "ClusterVersion": "1.0",  
  }  
}
```

```
"AllowVersionUpgrade": true,
"NumberOfNodes": 4,
"PubliclyAccessible": false,
"Encrypted": false,
"ClusterSnapshotCopyStatus": {
  "DestinationRegion": "us-west-1",
  "RetentionPeriod": 15,
  "ManualSnapshotRetentionPeriod": -1
},
"Tags": [
  {
    "Key": "mytags",
    "Value": "tag1"
  }
],
"EnhancedVpcRouting": false,
"IamRoles": [],
"MaintenanceTrackName": "current",
"DeferredMaintenanceWindows": [
  {
    "DeferMaintenanceIdentifier": "dfm-mUdVCfDcT1B4SGhw6fyF",
    "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
    "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
  }
],
"NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z",
"ResizeInfo": {
  "ResizeType": "ClassicResize",
  "AllowCancelResize": true
}
}
```

詳細については、「Amazon Redshift [クラスター管理ガイド](#)」の「[クラスターのサイズ変更](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス](#)[ResizeCluster](#)」の「」を参照してください。

restore-from-cluster-snapshot

次のコード例は、restore-from-cluster-snapshot を使用する方法を示しています。

AWS CLI

クラスターを復元する SnapshotThis 例から、スナップショットからクラスターを復元します。
コマンド:

```
aws redshift restore-from-cluster-snapshot --cluster-identifier mycluster-clone --
snapshot-identifier my-snapshot-id
```

結果:

```
{
  "Cluster": {
    "NodeType": "dw.hs1.xlarge",
    "ClusterVersion": "1.0",
    "PubliclyAccessible": "true",
    "MasterUsername": "adminuser",
    "ClusterParameterGroups": [
      {
        "ParameterApplyStatus": "in-sync",
        "ParameterGroupName": "default.redshift-1.0"
      }
    ],
    "ClusterSecurityGroups": [
      {
        "Status": "active",
        "ClusterSecurityGroupName": "default"
      }
    ],
    "AllowVersionUpgrade": true,
    "VpcSecurityGroups": \[],
    "PreferredMaintenanceWindow": "sun:23:15-mon:03:15",
    "AutomatedSnapshotRetentionPeriod": 1,
    "ClusterStatus": "creating",
    "ClusterIdentifier": "mycluster-clone",
    "DBName": "dev",
    "NumberOfNodes": 2,
    "PendingModifiedValues": {}
  },
  "ResponseMetadata": {
    "RequestId": "77fd512b-64e3-11e2-8f5b-e90bd6c77476"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RestoreFromClusterSnapshot](#)」の「」を参照してください。

restore-table-from-cluster-snapshot

次のコード例は、restore-table-from-cluster-snapshot を使用する方法を示しています。

AWS CLI

クラスタースナップショットからテーブルを復元するには

次のrestore-table-from-cluster-snapshot例では、指定されたクラスタースナップショット内の指定されたテーブルから新しいテーブルを作成します。

```
aws redshift restore-table-from-cluster-snapshot /
  --cluster-identifier mycluster /
  --snapshot-identifier mycluster-2019-11-19-16-17 /
  --source-database-name dev /
  --source-schema-name public /
  --source-table-name mytable /
  --target-database-name dev /
  --target-schema-name public /
  --new-table-name mytable-clone
```

出力:

```
{
  "TableRestoreStatus": {
    "TableRestoreRequestId": "a123a12b-abc1-1a1a-a123-a1234ab12345",
    "Status": "PENDING",
    "RequestTime": "2019-12-20T00:20:16.402Z",
    "ClusterIdentifier": "mycluster",
    "SnapshotIdentifier": "mycluster-2019-11-19-16-17",
    "SourceDatabaseName": "dev",
    "SourceSchemaName": "public",
    "SourceTableName": "mytable",
    "TargetDatabaseName": "dev",
    "TargetSchemaName": "public",
    "NewTableName": "mytable-clone"
  }
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「スナップショットからテーブルを復元する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreTableFromClusterSnapshot](#)」の「」を参照してください。

revoke-cluster-security-group-ingress

次のコード例は、`revoke-cluster-security-group-ingress` を使用する方法を示しています。

AWS CLI

EC2 セキュリティからのアクセスの取り消し GroupThis の例は、名前付き Amazon EC2 セキュリティグループへのアクセスを取り消します。コマンド：

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

CIDR rangeThis 例へのアクセスを取り消すと、CIDR 範囲へのアクセスが取り消されます。コマンド：

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --cidrip 192.168.100.100/32
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RevokeClusterSecurityGroupIngress](#)」の「」を参照してください。

revoke-snapshot-access

次の例は、`revoke-snapshot-access` を使用する方法を説明しています。

AWS CLI

AWS アカウントの認証を取り消して復元する SnapshotThis の例は、スナップショットを復元444455556666する AWS アカウントの認証を取り消しますmy-snapshot-id。デフォルトでは、出力は JSON 形式です。コマンド：

```
aws redshift revoke-snapshot-access --snapshot-id my-snapshot-id --account-with-restore-access 444455556666
```

結果:

```
{
  "Snapshot": {
    "Status": "available",
    "SnapshotCreateTime": "2013-07-17T22:04:18.947Z",
    "EstimatedSecondsToCompletion": 0,
    "AvailabilityZone": "us-east-1a",
    "ClusterVersion": "1.0",
    "MasterUsername": "adminuser",
    "Encrypted": false,
    "OwnerAccount": "111122223333",
    "BackupProgressInMegabytes": 11.0,
    "ElapsedTimeInSeconds": 0,
    "DBName": "dev",
    "CurrentBackupRateInMegabytesPerSecond": 0.1534,
    "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
    "ActualIncrementalBackupSizeInMegabytes": 11.0,
    "SnapshotType": "manual",
    "NodeType": "dw.hs1.xlarge",
    "ClusterIdentifier": "mycluster",
    "TotalBackupSizeInMegabytes": 20.0,
    "Port": 5439,
    "NumberOfNodes": 2,
    "SnapshotIdentifier": "my-snapshot-id"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RevokeSnapshotAccess](#)」の「」を参照してください。

rotate-encryption-key

次の例は、rotate-encryption-key を使用する方法を説明しています。

AWS CLI

クラスターの暗号化キーをローテーションするには

次のrotate-encryption-key例では、指定されたクラスターの暗号化キーをローテーションします。

```
aws redshift rotate-encryption-key \  
  --cluster-identifier mycluster
```

出力:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "rotating-keys",  
    "ClusterAvailabilityStatus": "Modifying",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-10T19:25:45.886Z",  
    "AutomatedSnapshotRetentionPeriod": 30,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sh-a1a123ab",  
        "Status": "active"  
      }  
    ],  
    "ClusterParameterGroups": [  
      {  
        "ParameterGroupName": "default.redshift-1.0",  
        "ParameterApplyStatus": "in-sync"  
      }  
    ],  
    "ClusterSubnetGroupName": "default",  
    "VpcId": "vpc-a1abc1a1",  
    "AvailabilityZone": "us-west-2a",  
    "PreferredMaintenanceWindow": "sat:16:00-sat:16:30",  
    "PendingModifiedValues": {},  
    "ClusterVersion": "1.0",  
    "AllowVersionUpgrade": true,  
  }  
}
```

```
    "NumberOfNodes": 2,
    "PubliclyAccessible": false,
    "Encrypted": true,
    "Tags": [],
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
    "EnhancedVpcRouting": false,
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
        "ApplyStatus": "in-sync"
      }
    ],
    "MaintenanceTrackName": "current",
    "DeferredMaintenanceWindows": [],
    "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
  }
}
```

詳細については、[「Amazon Redshift クラスター管理ガイド」の「Amazon Redshift データベース暗号化」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RotateEncryptionKey](#)」の「」を参照してください。

を使用した Amazon Rekognition の例 AWS CLI

次のコード例は、Amazon Rekognition AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

compare-faces

次の例は、compare-faces を使用する方法を説明しています。

詳細については、「[イメージ内の顔を比較する](#)」を参照してください。

AWS CLI

2 つの画像内の顔を比較するには

次の compare-faces コマンドは、Amazon S3 バケットに保存されている 2 つの画像の顔を比較します。

```
aws rekognition compare-faces \  
  --source-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"source.jpg"}}' \  
  --target-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"target.jpg"}}'
```

出力:

```
{  
  "UnmatchedFaces": [],  
  "FaceMatches": [  
    {  
      "Face": {  
        "BoundingBox": {  
          "Width": 0.12368916720151901,  
          "Top": 0.16007372736930847,  
          "Left": 0.5901257991790771,  
          "Height": 0.25140416622161865  
        },  
        "Confidence": 100.0,  
        "Pose": {  
          "Yaw": -3.7351467609405518,  
          "Roll": -0.10309021919965744,  
          "Pitch": 0.8637830018997192  
        },  
        "Quality": {  
          "Sharpness": 95.51618957519531,  
          "Brightness": 65.29893493652344  
        },  
        "Landmarks": [  
          {  
            "Type": "Forehead",  
            "X": 0.5901257991790771,  
            "Y": 0.16007372736930847  
          },  
          {  
            "Type": "LeftEye",  
            "X": 0.6501257991790771,  
            "Y": 0.22007372736930847  
          },  
          {  
            "Type": "RightEye",  
            "X": 0.7101257991790771,  
            "Y": 0.22007372736930847  
          },  
          {  
            "Type": "LeftMouth",  
            "X": 0.6501257991790771,  
            "Y": 0.32007372736930847  
          },  
          {  
            "Type": "RightMouth",  
            "X": 0.7101257991790771,  
            "Y": 0.32007372736930847  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
        {
            "Y": 0.26721030473709106,
            "X": 0.6204193830490112,
            "Type": "eyeLeft"
        },
        {
            "Y": 0.26831310987472534,
            "X": 0.6776827573776245,
            "Type": "eyeRight"
        },
        {
            "Y": 0.3514654338359833,
            "X": 0.6241428852081299,
            "Type": "mouthLeft"
        },
        {
            "Y": 0.35258132219314575,
            "X": 0.6713621020317078,
            "Type": "mouthRight"
        },
        {
            "Y": 0.3140771687030792,
            "X": 0.6428444981575012,
            "Type": "nose"
        }
    ]
},
"Similarity": 100.0
}
],
"SourceImageFace": {
    "BoundingBox": {
        "Width": 0.12368916720151901,
        "Top": 0.16007372736930847,
        "Left": 0.5901257991790771,
        "Height": 0.25140416622161865
    },
    "Confidence": 100.0
}
}
```

詳細については、「Amazon Rekognition デイベロツパーガイド」の「[イメージ間の顔の比較](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CompareFaces](#)」の「」を参照してください。

create-collection

次の例は、create-collection を使用する方法を説明しています。

詳細については、「[コレクションを作成する](#)」を参照してください。

AWS CLI

コレクションを作成するには

次の create-collection コマンドは、指定された名前のコレクションを作成します。

```
aws rekognition create-collection \  
  --collection-id "MyCollection"
```

出力:

```
{  
  "CollectionArn": "aws:rekognition:us-west-2:123456789012:collection/  
MyCollection",  
  "FaceModelVersion": "4.0",  
  "StatusCode": 200  
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[コレクションの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCollection](#)」の「」を参照してください。

create-stream-processor

次の例は、create-stream-processor を使用する方法を説明しています。

AWS CLI

新しいストリームプロセッサを作成するには

次のcreate-stream-processor例では、指定された設定で新しいストリームプロセッサを作成します。

```
aws rekognition create-stream-processor --name my-stream-processor \  
  --input '{"KinesisVideoStream":{"Arn":"arn:aws:kinesisvideo:us-  
west-2:123456789012:stream/macwebcam/1530559711205"}}' \  
  --stream-processor-output '{"KinesisDataStream":{"Arn":"arn:aws:kinesis:us-  
west-2:123456789012:stream/AmazonRekognitionRekStream"}}' \  
  --role-arn arn:aws:iam::123456789012:role/AmazonRekognitionDetect \  
  --settings '{"FaceSearch":  
{"CollectionId":"MyCollection","FaceMatchThreshold":85.5}}'
```

出力:

```
{  
  "StreamProcessorArn": "arn:aws:rekognition:us-  
west-2:123456789012:streamprocessor/my-stream-processor"  
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」の「ストリーミングビデオの使用」](#)を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateStreamProcessor](#)」の「」を参照してください。

delete-collection

次のコード例は、delete-collection を使用する方法を示しています。

詳細については、「[コレクションを削除する](#)」を参照してください。

AWS CLI

コレクションを削除するには

次の delete-collection コマンドは、指定されたコレクションを削除します。

```
aws rekognition delete-collection \  
  --collection-id MyCollection
```

出力:

```
{
  "StatusCode": 200
}
```

詳細については、「Amazon Rekognition デイベロッパーガイド」の「[コレクションの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCollection](#)」の「」を参照してください。

delete-faces

次のコード例は、delete-faces を使用する方法を示しています。

詳細については、「[コレクションから顔を削除する](#)」を参照してください。

AWS CLI

コレクションから顔を削除するには

次の delete-faces コマンドは、コレクションから指定された顔を削除します。

```
aws rekognition delete-faces \
  --collection-id MyCollection
  --face-ids '["0040279c-0178-436e-b70a-e61b074e96b0"]'
```

出力:

```
{
  "DeletedFaces": [
    "0040279c-0178-436e-b70a-e61b074e96b0"
  ]
}
```

詳細については、「Amazon Rekognition デイベロッパーガイド」の「[コレクションからの顔の削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFaces](#)」の「」を参照してください。

delete-stream-processor

次のコード例は、delete-stream-processor を使用する方法を示しています。

AWS CLI

ストリームプロセッサを削除するには

次のdelete-stream-processorコマンドは、指定されたストリームプロセッサを削除します。

```
aws rekognition delete-stream-processor \  
  --name my-stream-processor
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Rekognition デベロッパーガイド」](#)の「[ストリーミングビデオの使用](#)」を参照してください。 Amazon Rekognition

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteStreamProcessor](#)」の「」を参照してください。

describe-collection

次の例は、describe-collection を使用する方法を説明しています。

詳細については、「[コレクションを定義する](#)」を参照してください。

AWS CLI

コレクションを記述するには

次の describe-collection の例は、指定されたコレクションの詳細を表示します。

```
aws rekognition describe-collection \  
  --collection-id MyCollection
```

出力:

```
{  
  "FaceCount": 200,  
  "CreationTimestamp": 1569444828.274,
```



```
"CollectionARN": "arn:aws:rekognition:us-west-2:123456789012:collection/MyCollection",
  "FaceModelVersion": "4.0"
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[コレクションの定義](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCollection](#)」の「」を参照してください。

describe-stream-processor

次の例は、describe-stream-processor を使用する方法を説明しています。

AWS CLI

ストリームプロセッサに関する情報を取得するには

次のdescribe-stream-processorコマンドは、指定されたストリームプロセッサの詳細を表示します。

```
aws rekognition describe-stream-processor \
  --name my-stream-processor
```

出力:

```
{
  "Status": "STOPPED",
  "Name": "my-stream-processor",
  "LastUpdateTimestamp": 1532449292.712,
  "Settings": {
    "FaceSearch": {
      "FaceMatchThreshold": 80.0,
      "CollectionId": "my-collection"
    }
  },
  "RoleArn": "arn:aws:iam::123456789012:role/AmazonRekognitionDetectStream",
  "StreamProcessorArn": "arn:aws:rekognition:us-west-2:123456789012:streamprocessor/my-stream-processor",
  "Output": {
    "KinesisDataStream": {
```

```

    "Arn": "arn:aws:kinesis:us-west-2:123456789012:stream/
AmazonRekognitionRekStream"
  },
  "Input": {
    "KinesisVideoStream": {
      "Arn": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/
macwebcam/123456789012"
    }
  },
  "CreationTimestamp": 1532449292.712
}

```

詳細については、[「Amazon Rekognition デベロッパーガイド」の「ストリーミングビデオの使用」](#)を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStreamProcessor](#)」の「」を参照してください。

detect-faces

次の例は、detect-faces を使用する方法を説明しています。

詳細については、「[イメージ内の顔を検出する](#)」を参照してください。

AWS CLI

画像内の顔を検出するには

次の detect-faces コマンドは、Amazon S3 バケットに保存されている指定された画像の顔を検出します。

```

aws rekognition detect-faces \
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"MyFriend.jpg"}}' \
  --attributes "ALL"

```

出力:

```

{
  "FaceDetails": [
    {
      "Confidence": 100.0,
      "Eyeglasses": {

```

```
    "Confidence": 98.91107940673828,
    "Value": false
  },
  "Sunglasses": {
    "Confidence": 99.7966537475586,
    "Value": false
  },
  "Gender": {
    "Confidence": 99.56611633300781,
    "Value": "Male"
  },
  "Landmarks": [
    {
      "Y": 0.26721030473709106,
      "X": 0.6204193830490112,
      "Type": "eyeLeft"
    },
    {
      "Y": 0.26831310987472534,
      "X": 0.6776827573776245,
      "Type": "eyeRight"
    },
    {
      "Y": 0.3514654338359833,
      "X": 0.6241428852081299,
      "Type": "mouthLeft"
    },
    {
      "Y": 0.35258132219314575,
      "X": 0.6713621020317078,
      "Type": "mouthRight"
    },
    {
      "Y": 0.3140771687030792,
      "X": 0.6428444981575012,
      "Type": "nose"
    },
    {
      "Y": 0.24662546813488007,
      "X": 0.6001564860343933,
      "Type": "leftEyeBrowLeft"
    },
    {
      "Y": 0.24326619505882263,
```

```
    "X": 0.6303644776344299,  
    "Type": "leftEyeBrowRight"  
  },  
  {  
    "Y": 0.23818562924861908,  
    "X": 0.6146903038024902,  
    "Type": "leftEyeBrowUp"  
  },  
  {  
    "Y": 0.24373626708984375,  
    "X": 0.6640064716339111,  
    "Type": "rightEyeBrowLeft"  
  },  
  {  
    "Y": 0.24877218902111053,  
    "X": 0.7025929093360901,  
    "Type": "rightEyeBrowRight"  
  },  
  {  
    "Y": 0.23938551545143127,  
    "X": 0.6823262572288513,  
    "Type": "rightEyeBrowUp"  
  },  
  {  
    "Y": 0.265746533870697,  
    "X": 0.6112898588180542,  
    "Type": "leftEyeLeft"  
  },  
  {  
    "Y": 0.2676128149032593,  
    "X": 0.6317071914672852,  
    "Type": "leftEyeRight"  
  },  
  {  
    "Y": 0.262735515832901,  
    "X": 0.6201658248901367,  
    "Type": "leftEyeUp"  
  },  
  {  
    "Y": 0.27025148272514343,  
    "X": 0.6206279993057251,  
    "Type": "leftEyeDown"  
  },  
  {
```

```
        "Y": 0.268223375082016,  
        "X": 0.6658390760421753,  
        "Type": "rightEyeLeft"  
    },  
    {  
        "Y": 0.2672517001628876,  
        "X": 0.687832236289978,  
        "Type": "rightEyeRight"  
    },  
    {  
        "Y": 0.26383838057518005,  
        "X": 0.6769183874130249,  
        "Type": "rightEyeUp"  
    },  
    {  
        "Y": 0.27138751745224,  
        "X": 0.676596462726593,  
        "Type": "rightEyeDown"  
    },  
    {  
        "Y": 0.32283174991607666,  
        "X": 0.6350004076957703,  
        "Type": "noseLeft"  
    },  
    {  
        "Y": 0.3219289481639862,  
        "X": 0.6567046642303467,  
        "Type": "noseRight"  
    },  
    {  
        "Y": 0.3420318365097046,  
        "X": 0.6450609564781189,  
        "Type": "mouthUp"  
    },  
    {  
        "Y": 0.3664324879646301,  
        "X": 0.6455618143081665,  
        "Type": "mouthDown"  
    },  
    {  
        "Y": 0.26721030473709106,  
        "X": 0.6204193830490112,  
        "Type": "leftPupil"  
    },  
    },
```

```
{
  "Y": 0.26831310987472534,
  "X": 0.6776827573776245,
  "Type": "rightPupil"
},
{
  "Y": 0.26343393325805664,
  "X": 0.5946047306060791,
  "Type": "upperJawlineLeft"
},
{
  "Y": 0.3543180525302887,
  "X": 0.6044883728027344,
  "Type": "midJawlineLeft"
},
{
  "Y": 0.4084877669811249,
  "X": 0.6477024555206299,
  "Type": "chinBottom"
},
{
  "Y": 0.3562754988670349,
  "X": 0.707981526851654,
  "Type": "midJawlineRight"
},
{
  "Y": 0.26580461859703064,
  "X": 0.7234612107276917,
  "Type": "upperJawlineRight"
}
],
"Pose": {
  "Yaw": -3.7351467609405518,
  "Roll": -0.10309021919965744,
  "Pitch": 0.8637830018997192
},
"Emotions": [
  {
    "Confidence": 8.74203109741211,
    "Type": "SURPRISED"
  },
  {
    "Confidence": 2.501944065093994,
    "Type": "ANGRY"
  }
]
```

```
    },
    {
      "Confidence": 0.7378743290901184,
      "Type": "DISGUSTED"
    },
    {
      "Confidence": 3.5296201705932617,
      "Type": "HAPPY"
    },
    {
      "Confidence": 1.7162904739379883,
      "Type": "SAD"
    },
    {
      "Confidence": 9.518536567687988,
      "Type": "CONFUSED"
    },
    {
      "Confidence": 0.45474427938461304,
      "Type": "FEAR"
    },
    {
      "Confidence": 72.79895782470703,
      "Type": "CALM"
    }
  ],
  "AgeRange": {
    "High": 48,
    "Low": 32
  },
  "EyesOpen": {
    "Confidence": 98.93987274169922,
    "Value": true
  },
  "BoundingBox": {
    "Width": 0.12368916720151901,
    "Top": 0.16007372736930847,
    "Left": 0.5901257991790771,
    "Height": 0.25140416622161865
  },
  "Smile": {
    "Confidence": 93.4493179321289,
    "Value": false
  },
},
```

```
    "MouthOpen": {
      "Confidence": 90.53053283691406,
      "Value": false
    },
    "Quality": {
      "Sharpness": 95.51618957519531,
      "Brightness": 65.29893493652344
    },
    "Mustache": {
      "Confidence": 89.85221099853516,
      "Value": false
    },
    "Beard": {
      "Confidence": 86.1991195678711,
      "Value": true
    }
  }
]
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[イメージ内の顔の検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectFaces](#)」の「」を参照してください。

detect-labels

次のコード例は、detect-labels を使用する方法を示しています。

詳細については、「[イメージ内のラベルを検出する](#)」を参照してください。

AWS CLI

画像内のラベルを検出するには

次の detect-labels の例は、Amazon S3 バケットに保存されている画像内のシーンとオブジェクトを検出します。

```
aws rekognition detect-labels \
  --image '{"S3Object":{"Bucket":"bucket","Name":"image"}}'
```

出力:


```
{
  "Labels": [
    {
      "Instances": [],
      "Confidence": 99.15271759033203,
      "Parents": [
        {
          "Name": "Vehicle"
        },
        {
          "Name": "Transportation"
        }
      ],
      "Name": "Automobile"
    },
    {
      "Instances": [],
      "Confidence": 99.15271759033203,
      "Parents": [
        {
          "Name": "Transportation"
        }
      ],
      "Name": "Vehicle"
    },
    {
      "Instances": [],
      "Confidence": 99.15271759033203,
      "Parents": [],
      "Name": "Transportation"
    },
    {
      "Instances": [
        {
          "BoundingBox": {
            "Width": 0.10616336017847061,
            "Top": 0.5039216876029968,
            "Left": 0.0037978808395564556,
            "Height": 0.18528179824352264
          },
          "Confidence": 99.15271759033203
        }
      ],
    }
  ]
}
```

```
    "BoundingBox": {
      "Width": 0.2429988533258438,
      "Top": 0.5251884460449219,
      "Left": 0.7309805154800415,
      "Height": 0.21577216684818268
    },
    "Confidence": 99.1286392211914
  },
  {
    "BoundingBox": {
      "Width": 0.14233611524105072,
      "Top": 0.5333095788955688,
      "Left": 0.6494812965393066,
      "Height": 0.15528248250484467
    },
    "Confidence": 98.48368072509766
  },
  {
    "BoundingBox": {
      "Width": 0.11086395382881165,
      "Top": 0.5354844927787781,
      "Left": 0.10355594009160995,
      "Height": 0.10271988064050674
    },
    "Confidence": 96.45606231689453
  },
  {
    "BoundingBox": {
      "Width": 0.06254628300666809,
      "Top": 0.5573825240135193,
      "Left": 0.46083059906959534,
      "Height": 0.053911514580249786
    },
    "Confidence": 93.65448760986328
  },
  {
    "BoundingBox": {
      "Width": 0.10105438530445099,
      "Top": 0.534368634223938,
      "Left": 0.5743985772132874,
      "Height": 0.12226245552301407
    },
    "Confidence": 93.06217193603516
  },
},
```

```
{
  "BoundingBox": {
    "Width": 0.056389667093753815,
    "Top": 0.5235804319381714,
    "Left": 0.9427769780158997,
    "Height": 0.17163699865341187
  },
  "Confidence": 92.6864013671875
},
{
  "BoundingBox": {
    "Width": 0.06003860384225845,
    "Top": 0.5441341400146484,
    "Left": 0.22409997880458832,
    "Height": 0.06737709045410156
  },
  "Confidence": 90.4227066040039
},
{
  "BoundingBox": {
    "Width": 0.02848697081208229,
    "Top": 0.5107086896896362,
    "Left": 0,
    "Height": 0.19150497019290924
  },
  "Confidence": 86.65286254882812
},
{
  "BoundingBox": {
    "Width": 0.04067881405353546,
    "Top": 0.5566273927688599,
    "Left": 0.316415935754776,
    "Height": 0.03428703173995018
  },
  "Confidence": 85.36471557617188
},
{
  "BoundingBox": {
    "Width": 0.043411049991846085,
    "Top": 0.5394920110702515,
    "Left": 0.18293385207653046,
    "Height": 0.0893595889210701
  },
  "Confidence": 82.21705627441406
}
```

```
    },
    {
      "BoundingBox": {
        "Width": 0.031183116137981415,
        "Top": 0.5579366683959961,
        "Left": 0.2853088080883026,
        "Height": 0.03989990055561066
      },
      "Confidence": 81.0157470703125
    },
    {
      "BoundingBox": {
        "Width": 0.031113790348172188,
        "Top": 0.5504819750785828,
        "Left": 0.2580395042896271,
        "Height": 0.056484755128622055
      },
      "Confidence": 56.13441467285156
    },
    {
      "BoundingBox": {
        "Width": 0.08586374670267105,
        "Top": 0.5438792705535889,
        "Left": 0.5128012895584106,
        "Height": 0.08550430089235306
      },
      "Confidence": 52.37760925292969
    }
  ],
  "Confidence": 99.15271759033203,
  "Parents": [
    {
      "Name": "Vehicle"
    },
    {
      "Name": "Transportation"
    }
  ],
  "Name": "Car"
},
{
  "Instances": [],
  "Confidence": 98.9914321899414,
  "Parents": [],
```

```
    "Name": "Human"
  },
  {
    "Instances": [
      {
        "BoundingBox": {
          "Width": 0.19360728561878204,
          "Top": 0.35072067379951477,
          "Left": 0.43734854459762573,
          "Height": 0.2742200493812561
        },
        "Confidence": 98.9914321899414
      },
      {
        "BoundingBox": {
          "Width": 0.03801717236638069,
          "Top": 0.5010883808135986,
          "Left": 0.9155802130699158,
          "Height": 0.06597328186035156
        },
        "Confidence": 85.02790832519531
      }
    ],
    "Confidence": 98.9914321899414,
    "Parents": [],
    "Name": "Person"
  },
  {
    "Instances": [],
    "Confidence": 93.24951934814453,
    "Parents": [],
    "Name": "Machine"
  },
  {
    "Instances": [
      {
        "BoundingBox": {
          "Width": 0.03561960905790329,
          "Top": 0.6468243598937988,
          "Left": 0.7850857377052307,
          "Height": 0.08878646790981293
        },
        "Confidence": 93.24951934814453
      }
    ],
```

```
{
  "BoundingBox": {
    "Width": 0.02217046171426773,
    "Top": 0.6149078607559204,
    "Left": 0.04757237061858177,
    "Height": 0.07136218994855881
  },
  "Confidence": 91.5025863647461
},
{
  "BoundingBox": {
    "Width": 0.016197510063648224,
    "Top": 0.6274210214614868,
    "Left": 0.6472989320755005,
    "Height": 0.04955997318029404
  },
  "Confidence": 85.14686584472656
},
{
  "BoundingBox": {
    "Width": 0.020207518711686134,
    "Top": 0.6348286867141724,
    "Left": 0.7295016646385193,
    "Height": 0.07059963047504425
  },
  "Confidence": 83.34547424316406
},
{
  "BoundingBox": {
    "Width": 0.020280985161662102,
    "Top": 0.6171894669532776,
    "Left": 0.08744934946298599,
    "Height": 0.05297485366463661
  },
  "Confidence": 79.9981460571289
},
{
  "BoundingBox": {
    "Width": 0.018318990245461464,
    "Top": 0.623889148235321,
    "Left": 0.6836880445480347,
    "Height": 0.06730121374130249
  },
  "Confidence": 78.87144470214844
}
```

```
    },
    {
      "BoundingBox": {
        "Width": 0.021310249343514442,
        "Top": 0.6167286038398743,
        "Left": 0.004064912907779217,
        "Height": 0.08317798376083374
      },
      "Confidence": 75.89361572265625
    },
    {
      "BoundingBox": {
        "Width": 0.03604431077837944,
        "Top": 0.7030032277107239,
        "Left": 0.9254803657531738,
        "Height": 0.04569442570209503
      },
      "Confidence": 64.402587890625
    },
    {
      "BoundingBox": {
        "Width": 0.009834849275648594,
        "Top": 0.5821820497512817,
        "Left": 0.28094568848609924,
        "Height": 0.01964157074689865
      },
      "Confidence": 62.79907989501953
    },
    {
      "BoundingBox": {
        "Width": 0.01475677452981472,
        "Top": 0.6137543320655823,
        "Left": 0.5950819253921509,
        "Height": 0.039063986390829086
      },
      "Confidence": 59.40483474731445
    }
  ],
  "Confidence": 93.24951934814453,
  "Parents": [
    {
      "Name": "Machine"
    }
  ]
],
```

```
    "Name": "Wheel"
  },
  {
    "Instances": [],
    "Confidence": 92.61514282226562,
    "Parents": [],
    "Name": "Road"
  },
  {
    "Instances": [],
    "Confidence": 92.37877655029297,
    "Parents": [
      {
        "Name": "Person"
      }
    ],
    "Name": "Sport"
  },
  {
    "Instances": [],
    "Confidence": 92.37877655029297,
    "Parents": [
      {
        "Name": "Person"
      }
    ],
    "Name": "Sports"
  },
  {
    "Instances": [
      {
        "BoundingBox": {
          "Width": 0.12326609343290329,
          "Top": 0.6332163214683533,
          "Left": 0.44815489649772644,
          "Height": 0.058117982000112534
        },
        "Confidence": 92.37877655029297
      }
    ],
    "Confidence": 92.37877655029297,
    "Parents": [
      {
        "Name": "Person"
      }
    ]
  }
}
```



```
    },
    {
      "Name": "Sport"
    }
  ],
  "Name": "Skateboard"
},
{
  "Instances": [],
  "Confidence": 90.62931060791016,
  "Parents": [
    {
      "Name": "Person"
    }
  ],
  "Name": "Pedestrian"
},
{
  "Instances": [],
  "Confidence": 88.81334686279297,
  "Parents": [],
  "Name": "Asphalt"
},
{
  "Instances": [],
  "Confidence": 88.81334686279297,
  "Parents": [],
  "Name": "Tarmac"
},
{
  "Instances": [],
  "Confidence": 88.23201751708984,
  "Parents": [],
  "Name": "Path"
},
{
  "Instances": [],
  "Confidence": 80.26520538330078,
  "Parents": [],
  "Name": "Urban"
},
{
  "Instances": [],
  "Confidence": 80.26520538330078,
```

```
    "Parents": [
      {
        "Name": "Building"
      },
      {
        "Name": "Urban"
      }
    ],
    "Name": "Town"
  },
  {
    "Instances": [],
    "Confidence": 80.26520538330078,
    "Parents": [],
    "Name": "Building"
  },
  {
    "Instances": [],
    "Confidence": 80.26520538330078,
    "Parents": [
      {
        "Name": "Building"
      },
      {
        "Name": "Urban"
      }
    ],
    "Name": "City"
  },
  {
    "Instances": [],
    "Confidence": 78.37934875488281,
    "Parents": [
      {
        "Name": "Car"
      },
      {
        "Name": "Vehicle"
      },
      {
        "Name": "Transportation"
      }
    ],
    "Name": "Parking Lot"
  }
}
```

```
  },
  {
    "Instances": [],
    "Confidence": 78.37934875488281,
    "Parents": [
      {
        "Name": "Car"
      },
      {
        "Name": "Vehicle"
      },
      {
        "Name": "Transportation"
      }
    ],
    "Name": "Parking"
  },
  {
    "Instances": [],
    "Confidence": 74.37590026855469,
    "Parents": [
      {
        "Name": "Building"
      },
      {
        "Name": "Urban"
      },
      {
        "Name": "City"
      }
    ],
    "Name": "Downtown"
  },
  {
    "Instances": [],
    "Confidence": 69.84622955322266,
    "Parents": [
      {
        "Name": "Road"
      }
    ],
    "Name": "Intersection"
  },
  {
```

```
"Instances": [],
"Confidence": 57.68518829345703,
"Parents": [
  {
    "Name": "Sports Car"
  },
  {
    "Name": "Car"
  },
  {
    "Name": "Vehicle"
  },
  {
    "Name": "Transportation"
  }
],
"Name": "Coupe"
},
{
  "Instances": [],
  "Confidence": 57.68518829345703,
  "Parents": [
    {
      "Name": "Car"
    },
    {
      "Name": "Vehicle"
    },
    {
      "Name": "Transportation"
    }
  ],
  "Name": "Sports Car"
},
{
  "Instances": [],
  "Confidence": 56.59492111206055,
  "Parents": [
    {
      "Name": "Path"
    }
  ],
  "Name": "Sidewalk"
},
```

```
{
  "Instances": [],
  "Confidence": 56.59492111206055,
  "Parents": [
    {
      "Name": "Path"
    }
  ],
  "Name": "Pavement"
},
{
  "Instances": [],
  "Confidence": 55.58770751953125,
  "Parents": [
    {
      "Name": "Building"
    },
    {
      "Name": "Urban"
    }
  ],
  "Name": "Neighborhood"
}
],
"LabelModelVersion": "2.0"
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[イメージ内のラベルの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectLabels](#)」の「」を参照してください。

detect-moderation-labels

次の例は、detect-moderation-labels を使用する方法を説明しています。

詳細については、「[不適切なイメージを検出する](#)」を参照してください。

AWS CLI

画像内の安全でないコンテンツを検出するには

次の `detect-moderation-labels` コマンドは、Amazon S3 バケットに保存されている指定された画像の安全でないコンテンツを検出します。

```
aws rekognition detect-moderation-labels \  
  --image "S3Object={Bucket=MyImageS3Bucket,Name=gun.jpg}"
```

出力:

```
{  
  "ModerationModelVersion": "3.0",  
  "ModerationLabels": [  
    {  
      "Confidence": 97.29618072509766,  
      "ParentName": "Violence",  
      "Name": "Weapon Violence"  
    },  
    {  
      "Confidence": 97.29618072509766,  
      "ParentName": "",  
      "Name": "Violence"  
    }  
  ]  
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[不適切なイメージの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectModerationLabels](#)」の「」を参照してください。

detect-text

次の例は、`detect-text` を使用する方法を説明しています。

詳細については、「[イメージ内のテキストを検出する](#)」を参照してください。

AWS CLI

画像内のテキストを検出するには

次の `detect-text` コマンドは、指定された画像内のテキストを検出します。

```
aws rekognition detect-text \  
--image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePicture.jpg"}}'
```

出力:

```
{  
  "TextDetections": [  
    {  
      "Geometry": {  
        "BoundingBox": {  
          "Width": 0.24624845385551453,  
          "Top": 0.28288066387176514,  
          "Left": 0.391388863325119,  
          "Height": 0.022687450051307678  
        },  
        "Polygon": [  
          {  
            "Y": 0.28288066387176514,  
            "X": 0.391388863325119  
          },  
          {  
            "Y": 0.2826388478279114,  
            "X": 0.6376373171806335  
          },  
          {  
            "Y": 0.30532628297805786,  
            "X": 0.637677013874054  
          },  
          {  
            "Y": 0.305568128824234,  
            "X": 0.39142853021621704  
          }  
        ]  
      },  
      "Confidence": 94.35709381103516,  
      "DetectedText": "ESTD 1882",  
      "Type": "LINE",  
      "Id": 0  
    },  
    {  
      "Geometry": {  
        "BoundingBox": {  
          "Width": 0.33933889865875244,  
          "Height": 0.022687450051307678,  
          "Left": 0.391388863325119,  
          "Top": 0.28288066387176514  
        },  
        "Polygon": [  
          {  
            "Y": 0.28288066387176514,  
            "X": 0.391388863325119  
          },  
          {  
            "Y": 0.2826388478279114,  
            "X": 0.6376373171806335  
          },  
          {  
            "Y": 0.30532628297805786,  
            "X": 0.637677013874054  
          },  
          {  
            "Y": 0.305568128824234,  
            "X": 0.39142853021621704  
          }  
        ]  
      },  
      "Confidence": 94.35709381103516,  
      "DetectedText": "ESTD 1882",  
      "Type": "LINE",  
      "Id": 1  
    }  
  ]  
}
```

```
        "Top": 0.32603850960731506,  
        "Left": 0.34534579515457153,  
        "Height": 0.07126858830451965  
    },  
    "Polygon": [  
        {  
            "Y": 0.32603850960731506,  
            "X": 0.34534579515457153  
        },  
        {  
            "Y": 0.32633158564567566,  
            "X": 0.684684693813324  
        },  
        {  
            "Y": 0.3976001739501953,  
            "X": 0.684575080871582  
        },  
        {  
            "Y": 0.3973070979118347,  
            "X": 0.345236212015152  
        }  
    ]  
},  
"Confidence": 99.95779418945312,  
"DetectedText": "BRAINS",  
"Type": "LINE",  
"Id": 1  
},  
{  
    "Confidence": 97.22098541259766,  
    "Geometry": {  
        "BoundingBox": {  
            "Width": 0.061079490929841995,  
            "Top": 0.2843210697174072,  
            "Left": 0.391391396522522,  
            "Height": 0.021029088646173477  
        },  
        "Polygon": [  
            {  
                "Y": 0.2843210697174072,  
                "X": 0.391391396522522  
            },  
            {  
                "Y": 0.2828207015991211,
```



```
        "X": 0.4524524509906769
      },
      {
        "Y": 0.3038259446620941,
        "X": 0.4534534513950348
      },
      {
        "Y": 0.30532634258270264,
        "X": 0.3923923969268799
      }
    ]
  },
  "DetectedText": "ESTD",
  "ParentId": 0,
  "Type": "WORD",
  "Id": 2
},
{
  "Confidence": 91.49320983886719,
  "Geometry": {
    "BoundingBox": {
      "Width": 0.07007007300853729,
      "Top": 0.2828207015991211,
      "Left": 0.5675675868988037,
      "Height": 0.02250562608242035
    },
    "Polygon": [
      {
        "Y": 0.2828207015991211,
        "X": 0.5675675868988037
      },
      {
        "Y": 0.2828207015991211,
        "X": 0.6376376152038574
      },
      {
        "Y": 0.30532634258270264,
        "X": 0.6376376152038574
      },
      {
        "Y": 0.30532634258270264,
        "X": 0.5675675868988037
      }
    ]
  }
}
```

```
    },
    "DetectedText": "1882",
    "ParentId": 0,
    "Type": "WORD",
    "Id": 3
  },
  {
    "Confidence": 99.95779418945312,
    "Geometry": {
      "BoundingBox": {
        "Width": 0.33933934569358826,
        "Top": 0.32633158564567566,
        "Left": 0.3453453481197357,
        "Height": 0.07127484679222107
      },
      "Polygon": [
        {
          "Y": 0.32633158564567566,
          "X": 0.3453453481197357
        },
        {
          "Y": 0.32633158564567566,
          "X": 0.684684693813324
        },
        {
          "Y": 0.39759939908981323,
          "X": 0.6836836934089661
        },
        {
          "Y": 0.39684921503067017,
          "X": 0.3453453481197357
        }
      ]
    },
    "DetectedText": "BRAINS",
    "ParentId": 1,
    "Type": "WORD",
    "Id": 4
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectText](#)」の「」を参照してください。

disassociate-faces

次の例は、disassociate-faces を使用する方法を説明しています。

AWS CLI

```
aws rekognition disassociate-faces --face-ids list-of-face-ids
--user-id user-id --collection-id collection-name --region region-name
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateFaces](#)」の「」を参照してください。

get-celebrity-info

次のコード例は、get-celebrity-info を使用する方法を示しています。

AWS CLI

有名人に関する情報を取得するには

次の get-celebrity-info コマンドは、指定された有名人に関する詳細情報を表示します。id パラメータは以前に recognize-celebrities を呼び出したときのものです。

```
aws rekognition get-celebrity-info --id nnnnnnnn
```

出力:

```
{
  "Name": "Celeb A",
  "Urls": [
    "www.imdb.com/name/aaaaaaaa"
  ]
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[有名人に関する情報の取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCelebrityInfo](#)」の「」を参照してください。

get-celebrity-recognition

次の例は、get-celebrity-recognition を使用する方法を説明しています。

AWS CLI

有名人認識オペレーションの結果を取得するには

次のget-celebrity-recognitionコマンドは、 を呼び出して以前に開始した有名人認識オペレーションの結果を表示しますstart-celebrity-recognition。

```
aws rekognition get-celebrity-recognition \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

出力:

```
{
  "NextToken": "3D01Clx1CiT31VsRDkA03IybLb/h5AtDWSGuhYi
+N1FIJwwPtAkuKzDhL2rV3GcwmNt77+12",
  "Celebrities": [
    {
      "Timestamp": 0,
      "Celebrity": {
        "Confidence": 96.0,
        "Face": {
          "BoundingBox": {
            "Width": 0.70333331823349,
            "Top": 0.16750000417232513,
            "Left": 0.19555555284023285,
            "Height": 0.3956249952316284
          },
          "Landmarks": [
            {
              "Y": 0.31031012535095215,
              "X": 0.441436767578125,
              "Type": "eyeLeft"
            },
            {
              "Y": 0.3081788718700409,
```

```
        "X": 0.6437258720397949,
        "Type": "eyeRight"
    },
    {
        "Y": 0.39542075991630554,
        "X": 0.5572493076324463,
        "Type": "nose"
    },
    {
        "Y": 0.4597957134246826,
        "X": 0.4579732120037079,
        "Type": "mouthLeft"
    },
    {
        "Y": 0.45688048005104065,
        "X": 0.6349081993103027,
        "Type": "mouthRight"
    }
],
"Pose": {
    "Yaw": 8.943398475646973,
    "Roll": -2.0309247970581055,
    "Pitch": -0.5674862861633301
},
"Quality": {
    "Sharpness": 99.40211486816406,
    "Brightness": 89.47132110595703
},
"Confidence": 99.99861145019531
},
"Name": "CelebrityA",
"Urls": [
    "www.imdb.com/name/1111111111"
],
"Id": "nnnnnn"
}
},
{
    "Timestamp": 467,
    "Celebrity": {
        "Confidence": 99.0,
        "Face": {
            "BoundingBox": {
                "Width": 0.6877777576446533,
```

```
        "Top": 0.18437500298023224,  
        "Left": 0.20555555820465088,  
        "Height": 0.3868750035762787  
    },  
    "Landmarks": [  
        {  
            "Y": 0.31895750761032104,  
            "X": 0.4411413371562958,  
            "Type": "eyeLeft"  
        },  
        {  
            "Y": 0.3140959143638611,  
            "X": 0.6523157954216003,  
            "Type": "eyeRight"  
        },  
        {  
            "Y": 0.4016456604003906,  
            "X": 0.5682755708694458,  
            "Type": "nose"  
        },  
        {  
            "Y": 0.46894142031669617,  
            "X": 0.4597797095775604,  
            "Type": "mouthLeft"  
        },  
        {  
            "Y": 0.46971091628074646,  
            "X": 0.6286435127258301,  
            "Type": "mouthRight"  
        }  
    ],  
    "Pose": {  
        "Yaw": 10.433465957641602,  
        "Roll": -3.347442388534546,  
        "Pitch": 1.3709543943405151  
    },  
    "Quality": {  
        "Sharpness": 99.5531005859375,  
        "Brightness": 88.5764389038086  
    },  
    "Confidence": 99.99148559570312  
},  
"Name": "Jane Celebrity",  
"Urls": [  
    ]
```

```

        "www.imdb.com/name/1111111111"
      ],
      "Id": "nnnnnn"
    }
  ]
  "JobStatus": "SUCCEEDED",
  "VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.978118896484375,
    "Codec": "h264",
    "DurationMillis": 4570,
    "FrameHeight": 1920,
    "FrameWidth": 1080
  }
}

```

詳細については、[「Amazon Rekognition デベロッパーガイド」の「ストアドビデオの有名人の認識」](#)を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCelebrityRecognition](#)」の「」を参照してください。

get-content-moderation

次のコード例は、get-content-moderation を使用する方法を示しています。

AWS CLI

安全でないコンテンツオペレーションの結果を取得するには

次のget-content-moderationコマンドは、 を呼び出して以前に開始した安全でないコンテンツオペレーションの結果を表示しますstart-content-moderation。

```

aws rekognition get-content-moderation \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef

```

出力:

```

{
  "NextToken": "d1hcKMHMzpCBGFukz6I03JMcWiJAamCVhXHt3r6b4b5Tfbyw3q7o+Jeezt
+Zpgf0nW9FCCgQ",

```

```
"ModerationLabels": [
  {
    "Timestamp": 0,
    "ModerationLabel": {
      "Confidence": 97.39583587646484,
      "ParentName": "",
      "Name": "Violence"
    }
  },
  {
    "Timestamp": 0,
    "ModerationLabel": {
      "Confidence": 97.39583587646484,
      "ParentName": "Violence",
      "Name": "Weapon Violence"
    }
  }
],
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
  "Format": "QuickTime / MOV",
  "FrameRate": 29.97515869140625,
  "Codec": "h264",
  "DurationMillis": 6039,
  "FrameHeight": 1920,
  "FrameWidth": 1080
}
}
```

詳細については、[Amazon Rekognition デベロッパーガイド](#)の「[安全でないストアビデオの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetContentModeration](#)」の「」を参照してください。

get-face-detection

次の例は、get-face-detection を使用する方法を説明しています。

AWS CLI

顔検出オペレーションの結果を取得するには

次のget-face-detectionコマンドは、 を呼び出して以前に開始した顔検出オペレーションの結果を表示しますstart-face-detection。

```
aws rekognition get-face-detection \  
--job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

出力:

```
{  
  "Faces": [  
    {  
      "Timestamp": 467,  
      "Face": {  
        "BoundingBox": {  
          "Width": 0.1560753583908081,  
          "Top": 0.13555361330509186,  
          "Left": -0.0952017530798912,  
          "Height": 0.6934483051300049  
        },  
        "Landmarks": [  
          {  
            "Y": 0.4013825058937073,  
            "X": -0.041750285774469376,  
            "Type": "eyeLeft"  
          },  
          {  
            "Y": 0.41695496439933777,  
            "X": 0.027979329228401184,  
            "Type": "eyeRight"  
          },  
          {  
            "Y": 0.6375303268432617,  
            "X": -0.04034662991762161,  
            "Type": "mouthLeft"  
          },  
          {  
            "Y": 0.6497718691825867,  
            "X": 0.013960429467260838,  
            "Type": "mouthRight"  
          },  
          {  
            "Y": 0.5238034129142761,  
            "X": 0.008022055961191654,  
            "Type": "mouthLeft"  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
        "Type": "nose"
      }
    ],
    "Pose": {
      "Yaw": -58.07863998413086,
      "Roll": 1.9384294748306274,
      "Pitch": -24.66305160522461
    },
    "Quality": {
      "Sharpness": 83.14741516113281,
      "Brightness": 25.75942611694336
    },
    "Confidence": 87.7622299194336
  }
},
{
  "Timestamp": 967,
  "Face": {
    "BoundingBox": {
      "Width": 0.28559377789497375,
      "Top": 0.19436298310756683,
      "Left": 0.024553587660193443,
      "Height": 0.7216082215309143
    },
    "Landmarks": [
      {
        "Y": 0.4650231599807739,
        "X": 0.16269078850746155,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.4843238294124603,
        "X": 0.2782580852508545,
        "Type": "eyeRight"
      },
      {
        "Y": 0.71530681848526,
        "X": 0.1741468608379364,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.7310671210289001,
        "X": 0.26857468485832214,
        "Type": "mouthRight"
      }
    ]
  }
}
```

```
    },
    {
      "Y": 0.582602322101593,
      "X": 0.2566150426864624,
      "Type": "nose"
    }
  ],
  "Pose": {
    "Yaw": 11.487052917480469,
    "Roll": 5.074230670928955,
    "Pitch": 15.396159172058105
  },
  "Quality": {
    "Sharpness": 73.32209777832031,
    "Brightness": 54.96497344970703
  },
  "Confidence": 99.99998474121094
}
]
}
"NextToken":
"0zL223pDKy91160/02KXRqFIEAwxy4PkgYcm3hSo0rdysbXg5Ex0eFgTGEj0ADEac6S037U",
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
  "Format": "QuickTime / MOV",
  "FrameRate": 29.970617294311523,
  "Codec": "h264",
  "DurationMillis": 6806,
  "FrameHeight": 1080,
  "FrameWidth": 1920
}
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」](#)の「ストアドビデオ内の顔の検出」を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFaceDetection](#)」の「」を参照してください。

get-face-search

次の例は、get-face-search を使用する方法を説明しています。

AWS CLI

顔検索オペレーションの結果を取得するには

次のget-face-searchコマンドは、 を呼び出して以前に開始した顔検索オペレーションの結果を表示しますstart-face-search。

```
aws rekognition get-face-search \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

出力:

```
{
  "Persons": [
    {
      "Timestamp": 467,
      "FaceMatches": [],
      "Person": {
        "Index": 0,
        "Face": {
          "BoundingBox": {
            "Width": 0.1560753583908081,
            "Top": 0.13555361330509186,
            "Left": -0.0952017530798912,
            "Height": 0.6934483051300049
          },
          "Landmarks": [
            {
              "Y": 0.4013825058937073,
              "X": -0.041750285774469376,
              "Type": "eyeLeft"
            },
            {
              "Y": 0.41695496439933777,
              "X": 0.027979329228401184,
              "Type": "eyeRight"
            },
            {
              "Y": 0.6375303268432617,
              "X": -0.04034662991762161,
              "Type": "mouthLeft"
            },
            {
```

```
        "Y": 0.6497718691825867,
        "X": 0.013960429467260838,
        "Type": "mouthRight"
      },
      {
        "Y": 0.5238034129142761,
        "X": 0.008022055961191654,
        "Type": "nose"
      }
    ],
    "Pose": {
      "Yaw": -58.07863998413086,
      "Roll": 1.9384294748306274,
      "Pitch": -24.66305160522461
    },
    "Quality": {
      "Sharpness": 83.14741516113281,
      "Brightness": 25.75942611694336
    },
    "Confidence": 87.7622299194336
  }
},
{
  "Timestamp": 967,
  "FaceMatches": [
    {
      "Face": {
        "BoundingBox": {
          "Width": 0.12368900328874588,
          "Top": 0.16007399559020996,
          "Left": 0.5901259779930115,
          "Height": 0.2514039874076843
        },
        "FaceId": "056a95fa-2060-4159-9cab-7ed4daa030fa",
        "ExternalImageId": "image3.jpg",
        "Confidence": 100.0,
        "ImageId": "08f8a078-8929-37fd-8e8f-aadf690e8232"
      },
      "Similarity": 98.44476318359375
    }
  ],
  "Person": {
    "Index": 1,
```

```
"Face": {
  "BoundingBox": {
    "Width": 0.28559377789497375,
    "Top": 0.19436298310756683,
    "Left": 0.024553587660193443,
    "Height": 0.7216082215309143
  },
  "Landmarks": [
    {
      "Y": 0.4650231599807739,
      "X": 0.16269078850746155,
      "Type": "eyeLeft"
    },
    {
      "Y": 0.4843238294124603,
      "X": 0.2782580852508545,
      "Type": "eyeRight"
    },
    {
      "Y": 0.71530681848526,
      "X": 0.1741468608379364,
      "Type": "mouthLeft"
    },
    {
      "Y": 0.7310671210289001,
      "X": 0.26857468485832214,
      "Type": "mouthRight"
    },
    {
      "Y": 0.582602322101593,
      "X": 0.2566150426864624,
      "Type": "nose"
    }
  ],
  "Pose": {
    "Yaw": 11.487052917480469,
    "Roll": 5.074230670928955,
    "Pitch": 15.396159172058105
  },
  "Quality": {
    "Sharpness": 73.32209777832031,
    "Brightness": 54.96497344970703
  },
  "Confidence": 99.99998474121094
}
```

```
    }
  }
},
"NextToken": "5bkgcezyuaqhtWk3C80TW6cjRghrwV9XDMivm5B3MXm+Lv6G+L+GejyFHPhoNa/ldXIC4c/d",
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
  "Format": "QuickTime / MOV",
  "FrameRate": 29.970617294311523,
  "Codec": "h264",
  "DurationMillis": 6806,
  "FrameHeight": 1080,
  "FrameWidth": 1920
}
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」](#)の「[顔の保存済み動画の検索](#)」を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「[コマンドリファレンスGetFaceSearch](#)」の「」を参照してください。

get-label-detection

次の例は、get-label-detection を使用する方法を説明しています。

AWS CLI

オブジェクトとシーンの検出オペレーションの結果を取得するには

次のget-label-detectionコマンドは、 を呼び出して以前に開始したオブジェクトとシーンの検出オペレーションの結果を表示しますstart-label-detection。

```
aws rekognition get-label-detection \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

出力:

```
{
  "Labels": [
    {
      "Timestamp": 0,
```

```
    "Label": {
      "Instances": [],
      "Confidence": 50.19071578979492,
      "Parents": [
        {
          "Name": "Person"
        },
        {
          "Name": "Crowd"
        }
      ],
      "Name": "Audience"
    }
  },
  {
    "Timestamp": 0,
    "Label": {
      "Instances": [],
      "Confidence": 55.74115753173828,
      "Parents": [
        {
          "Name": "Room"
        },
        {
          "Name": "Indoors"
        },
        {
          "Name": "School"
        }
      ],
      "Name": "Classroom"
    }
  }
],
"JobStatus": "SUCCEEDED",
"LabelModelVersion": "2.0",
"VideoMetadata": {
  "Format": "QuickTime / MOV",
  "FrameRate": 29.970617294311523,
  "Codec": "h264",
  "DurationMillis": 6806,
  "FrameHeight": 1080,
  "FrameWidth": 1920
},
```



```
"NextToken": "BMugzAi4L72IERzQdbpyMQuEFBsjl05W0Yx3mfG+sR9mm98E1/
Cp0benspRfs/5FBQFs4X7G"
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」の「動画内のラベルの検出」](#)を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[GetLabelDetection](#)」の「」を参照してください。

get-person-tracking

次のコード例は、get-person-tracking を使用する方法を示しています。

AWS CLI

人物パスオペレーションの結果を取得するには

次のget-person-trackingコマンドは、 を呼び出して以前に開始したユーザーパス操作の結果を表示しますstart-person-tracking。

```
aws rekognition get-person-tracking \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

出力:

```
{
  "Persons": [
    {
      "Timestamp": 500,
      "Person": {
        "BoundingBox": {
          "Width": 0.4151041805744171,
          "Top": 0.07870370149612427,
          "Left": 0.0,
          "Height": 0.9212962985038757
        },
        "Index": 0
      }
    },
    {
      "Timestamp": 567,
```

```
    "Person": {
      "BoundingBox": {
        "Width": 0.4755208194255829,
        "Top": 0.07777778059244156,
        "Left": 0.0,
        "Height": 0.9194444417953491
      },
      "Index": 0
    }
  ],
  "NextToken": "D/vRIYNyhG79ugdta3f+8cRg9oSRo
+HigG0uxRiYpTn0ExnqTi1CJektVAc4HrAXDv25eHYk",
  "JobStatus": "SUCCEEDED",
  "VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.970617294311523,
    "Codec": "h264",
    "DurationMillis": 6806,
    "FrameHeight": 1080,
    "FrameWidth": 1920
  }
}
```

詳細については、Amazon Rekognition デベロッパーガイド」の「[People Pathing](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPersonTracking](#)」の「」を参照してください。

index-faces

次のコード例は、index-faces を使用する方法を示しています。

詳細については、「[コレクションに顔を追加する](#)」を参照してください。

AWS CLI

コレクションに顔を追加するには

次の index-faces コマンドは、画像内の顔を指定されたコレクションに追加します。

```
aws rekognition index-faces \
```

```
--image '{"S3Object":{"Bucket":"MyVideoS3Bucket","Name":"MyPicture.jpg"}}' \  
--collection-id MyCollection \  
--max-faces 1 \  
--quality-filter "AUTO" \  
--detection-attributes "ALL" \  
--external-image-id "MyPicture.jpg"
```

出力:

```
{  
  "FaceRecords": [  
    {  
      "FaceDetail": {  
        "Confidence": 99.993408203125,  
        "Eyeglasses": {  
          "Confidence": 99.11750030517578,  
          "Value": false  
        },  
        "Sunglasses": {  
          "Confidence": 99.98249053955078,  
          "Value": false  
        },  
        "Gender": {  
          "Confidence": 99.92769622802734,  
          "Value": "Male"  
        },  
        "Landmarks": [  
          {  
            "Y": 0.26750367879867554,  
            "X": 0.6202793717384338,  
            "Type": "eyeLeft"  
          },  
          {  
            "Y": 0.26642778515815735,  
            "X": 0.6787431836128235,  
            "Type": "eyeRight"  
          },  
          {  
            "Y": 0.31361380219459534,  
            "X": 0.6421601176261902,  
            "Type": "nose"  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
    "Y": 0.3495299220085144,  
    "X": 0.6216195225715637,  
    "Type": "mouthLeft"  
  },  
  {  
    "Y": 0.35194727778434753,  
    "X": 0.669899046421051,  
    "Type": "mouthRight"  
  },  
  {  
    "Y": 0.26844894886016846,  
    "X": 0.6210268139839172,  
    "Type": "leftPupil"  
  },  
  {  
    "Y": 0.26707562804222107,  
    "X": 0.6817160844802856,  
    "Type": "rightPupil"  
  },  
  {  
    "Y": 0.24834522604942322,  
    "X": 0.6018546223640442,  
    "Type": "leftEyeBrowLeft"  
  },  
  {  
    "Y": 0.24397172033786774,  
    "X": 0.6172008514404297,  
    "Type": "leftEyeBrowUp"  
  },  
  {  
    "Y": 0.24677404761314392,  
    "X": 0.6339119076728821,  
    "Type": "leftEyeBrowRight"  
  },  
  {  
    "Y": 0.24582654237747192,  
    "X": 0.6619398593902588,  
    "Type": "rightEyeBrowLeft"  
  },  
  {  
    "Y": 0.23973053693771362,  
    "X": 0.6804757118225098,  
    "Type": "rightEyeBrowUp"  
  },  
}
```

```
{
  "Y": 0.24441994726657867,
  "X": 0.6978968977928162,
  "Type": "rightEyeBrowRight"
},
{
  "Y": 0.2695908546447754,
  "X": 0.6085202693939209,
  "Type": "leftEyeLeft"
},
{
  "Y": 0.26716896891593933,
  "X": 0.6315826177597046,
  "Type": "leftEyeRight"
},
{
  "Y": 0.26289820671081543,
  "X": 0.6202316880226135,
  "Type": "leftEyeUp"
},
{
  "Y": 0.27123287320137024,
  "X": 0.6205548048019409,
  "Type": "leftEyeDown"
},
{
  "Y": 0.2668408751487732,
  "X": 0.6663622260093689,
  "Type": "rightEyeLeft"
},
{
  "Y": 0.26741549372673035,
  "X": 0.6910083889961243,
  "Type": "rightEyeRight"
},
{
  "Y": 0.2614026665687561,
  "X": 0.6785826086997986,
  "Type": "rightEyeUp"
},
{
  "Y": 0.27075251936912537,
  "X": 0.6789616942405701,
  "Type": "rightEyeDown"
}
```

```
    },
    {
      "Y": 0.3211299479007721,
      "X": 0.6324167847633362,
      "Type": "noseLeft"
    },
    {
      "Y": 0.32276326417922974,
      "X": 0.6558475494384766,
      "Type": "noseRight"
    },
    {
      "Y": 0.34385165572166443,
      "X": 0.6444970965385437,
      "Type": "mouthUp"
    },
    {
      "Y": 0.3671635091304779,
      "X": 0.6459195017814636,
      "Type": "mouthDown"
    }
  ],
  "Pose": {
    "Yaw": -9.54541015625,
    "Roll": -0.5709401965141296,
    "Pitch": 0.6045494675636292
  },
  "Emotions": [
    {
      "Confidence": 39.90074157714844,
      "Type": "HAPPY"
    },
    {
      "Confidence": 23.38753890991211,
      "Type": "CALM"
    },
    {
      "Confidence": 5.840933322906494,
      "Type": "CONFUSED"
    }
  ],
  "AgeRange": {
    "High": 63,
    "Low": 45
  }
}
```

```
    },
    "EyesOpen": {
      "Confidence": 99.80887603759766,
      "Value": true
    },
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618015021085739,
      "Left": 0.5575000047683716,
      "Height": 0.24770642817020416
    },
    "Smile": {
      "Confidence": 99.69740295410156,
      "Value": false
    },
    "MouthOpen": {
      "Confidence": 99.97393798828125,
      "Value": false
    },
    "Quality": {
      "Sharpness": 95.54405975341797,
      "Brightness": 63.867706298828125
    },
    "Mustache": {
      "Confidence": 97.05007934570312,
      "Value": false
    },
    "Beard": {
      "Confidence": 87.34505462646484,
      "Value": false
    }
  },
  "Face": {
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618015021085739,
      "Left": 0.5575000047683716,
      "Height": 0.24770642817020416
    },
    "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
    "ExternalImageId": "example-image.jpg",
    "Confidence": 99.993408203125,
    "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
  }
}
```

```
    }
  ],
  "UnindexedFaces": [],
  "FaceModelVersion": "3.0",
  "OrientationCorrection": "ROTATE_0"
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[コレクションへの顔の追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[IndexFaces](#)」の「」を参照してください。

list-collections

次の例は、list-collections を使用する方法を説明しています。

コレクションの詳細については、「[コレクションを一覧表示する](#)」を参照してください。

AWS CLI

使用可能なコレクションを一覧表示するには

次のlist-collectionsコマンドは、AWS アカウントで使用可能なコレクションを一覧表示します。

```
aws rekognition list-collections
```

出力:

```
{
  "FaceModelVersions": [
    "2.0",
    "3.0",
    "3.0",
    "3.0",
    "4.0",
    "1.0",
    "3.0",
    "4.0",
    "4.0",
    "4.0"
  ],
```



```
"CollectionIds": [  
  "MyCollection1",  
  "MyCollection2",  
  "MyCollection3",  
  "MyCollection4",  
  "MyCollection5",  
  "MyCollection6",  
  "MyCollection7",  
  "MyCollection8",  
  "MyCollection9",  
  "MyCollection10"  
]  
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[コレクションの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCollections](#)」の「」を参照してください。

list-faces

次の例は、list-faces を使用する方法を説明しています。

詳細については、「[コレクションに顔を保存する](#)」を参照してください。

AWS CLI

コレクション内の顔を一覧表示するには

次の list-faces コマンドは、指定されたコレクション内の顔を一覧表示します。

```
aws rekognition list-faces \  
  --collection-id MyCollection
```

出力:

```
{  
  "FaceModelVersion": "3.0",  
  "Faces": [  
    {  
      "BoundingBox": {  
        "Width": 0.5216310024261475,
```

```
        "Top": 0.3256250023841858,
        "Left": 0.13394300639629364,
        "Height": 0.3918749988079071
    },
    "FaceId": "0040279c-0178-436e-b70a-e61b074e96b0",
    "ExternalImageId": "image1.jpg",
    "Confidence": 100.0,
    "ImageId": "f976e487-3719-5e2d-be8b-ea2724c26991"
},
{
    "BoundingBox": {
        "Width": 0.5074880123138428,
        "Top": 0.3774999976158142,
        "Left": 0.18302799761295319,
        "Height": 0.3812499940395355
    },
    "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
    "ExternalImageId": "image2.jpg",
    "Confidence": 99.99930572509766,
    "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
},
{
    "BoundingBox": {
        "Width": 0.5574039816856384,
        "Top": 0.37187498807907104,
        "Left": 0.14559100568294525,
        "Height": 0.4181250035762787
    },
    "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
    "ExternalImageId": "image3.jpg",
    "Confidence": 99.99960327148438,
    "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
},
{
    "BoundingBox": {
        "Width": 0.18562500178813934,
        "Top": 0.1618019938468933,
        "Left": 0.5575000047683716,
        "Height": 0.24770599603652954
    },
    "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
    "ExternalImageId": "image4.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
```

```
  },
  {
    "BoundingBox": {
      "Width": 0.5307819843292236,
      "Top": 0.2862499952316284,
      "Left": 0.1564060002565384,
      "Height": 0.3987500071525574
    },
    "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
    "ExternalImageId": "image5.jpg",
    "Confidence": 99.99970245361328,
    "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
  },
  {
    "BoundingBox": {
      "Width": 0.5773710012435913,
      "Top": 0.34437501430511475,
      "Left": 0.12396000325679779,
      "Height": 0.4337500035762787
    },
    "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
    "ExternalImageId": "image6.jpg",
    "Confidence": 100.0,
    "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
  },
  {
    "BoundingBox": {
      "Width": 0.5349419713020325,
      "Top": 0.29124999046325684,
      "Left": 0.16389399766921997,
      "Height": 0.40187498927116394
    },
    "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
    "ExternalImageId": "image7.jpg",
    "Confidence": 99.99979400634766,
    "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
  },
  {
    "BoundingBox": {
      "Width": 0.41499999165534973,
      "Top": 0.09187500178813934,
      "Left": 0.28083300590515137,
      "Height": 0.3112500011920929
    },
  },
```

```
    "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
    "ExternalImageId": "image8.jpg",
    "Confidence": 99.99769592285156,
    "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
  },
  {
    "BoundingBox": {
      "Width": 0.48166701197624207,
      "Top": 0.20999999344348907,
      "Left": 0.21250000596046448,
      "Height": 0.36125001311302185
    },
    "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
    "ExternalImageId": "image9.jpg",
    "Confidence": 99.99949645996094,
    "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
  },
  {
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618019938468933,
      "Left": 0.5575000047683716,
      "Height": 0.24770599603652954
    },
    "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
    "ExternalImageId": "image10.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
  }
]
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[コレクション内の顔と関連するユーザーを一覧表示します。](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFaces](#)」の「」を参照してください。

list-stream-processors

次のコード例は、list-stream-processors を使用する方法を示しています。

AWS CLI

アカウントのストリームプロセッサを一覧表示するには

次のlist-stream-processorsコマンドは、アカウントのストリームプロセッサと各の状態を一覧表示します。

```
aws rekognition list-stream-processors
```

出力:

```
{
  "StreamProcessors": [
    {
      "Status": "STOPPED",
      "Name": "my-stream-processor"
    }
  ]
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」](#)の「[ストリーミングビデオの使用](#)」を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「[コマンドリファレンスListStreamProcessors](#)」の「」を参照してください。

recognize-celebrities

次のコード例は、recognize-celebrities を使用する方法を示しています。

詳細については、「[イメージ内で有名人を認識する](#)」を参照してください。

AWS CLI

画像内の有名人を認識するには

次の recognize-celebrities コマンドは、Amazon S3 バケットに保存されている指定された画像の有名人を識別します。

```
aws rekognition recognize-celebrities \
  --image "S3Object={Bucket=MyImageS3Bucket,Name=moviestars.jpg}"
```

出力:

```
{
  "UnrecognizedFaces": [
    {
      "BoundingBox": {
        "Width": 0.14416666328907013,
        "Top": 0.077777778059244156,
        "Left": 0.625,
        "Height": 0.2746031880378723
      },
      "Confidence": 99.9990234375,
      "Pose": {
        "Yaw": 10.80408763885498,
        "Roll": -12.761146545410156,
        "Pitch": 10.96889877319336
      },
      "Quality": {
        "Sharpness": 94.1185531616211,
        "Brightness": 79.18367004394531
      },
      "Landmarks": [
        {
          "Y": 0.18220913410186768,
          "X": 0.6702951788902283,
          "Type": "eyeLeft"
        },
        {
          "Y": 0.16337193548679352,
          "X": 0.7188183665275574,
          "Type": "eyeRight"
        },
        {
          "Y": 0.20739148557186127,
          "X": 0.7055801749229431,
          "Type": "nose"
        },
        {
          "Y": 0.2889308035373688,
          "X": 0.687512218952179,
          "Type": "mouthLeft"
        },
        {
          "Y": 0.2706988751888275,
```

```
        "X": 0.7250053286552429,
        "Type": "mouthRight"
      }
    ]
  },
  "CelebrityFaces": [
    {
      "MatchConfidence": 100.0,
      "Face": {
        "BoundingBox": {
          "Width": 0.14000000059604645,
          "Top": 0.1190476194024086,
          "Left": 0.82833331823349,
          "Height": 0.2666666805744171
        },
        "Confidence": 99.99359130859375,
        "Pose": {
          "Yaw": -10.509642601013184,
          "Roll": -14.51749324798584,
          "Pitch": 13.799399375915527
        },
        "Quality": {
          "Sharpness": 78.74752044677734,
          "Brightness": 42.201324462890625
        },
        "Landmarks": [
          {
            "Y": 0.2290833294391632,
            "X": 0.8709492087364197,
            "Type": "eyeLeft"
          },
          {
            "Y": 0.20639978349208832,
            "X": 0.9153988361358643,
            "Type": "eyeRight"
          },
          {
            "Y": 0.25417643785476685,
            "X": 0.8907724022865295,
            "Type": "nose"
          },
          {
            "Y": 0.32729196548461914,
```

```
        "X": 0.8876466155052185,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.3115464746952057,
        "X": 0.9238573312759399,
        "Type": "mouthRight"
      }
    ]
  },
  "Name": "Celeb A",
  "Urls": [
    "www.imdb.com/name/aaaaaaaaa"
  ],
  "Id": "1111111"
},
{
  "MatchConfidence": 97.0,
  "Face": {
    "BoundingBox": {
      "Width": 0.13333334028720856,
      "Top": 0.24920634925365448,
      "Left": 0.4449999928474426,
      "Height": 0.2539682686328888
    },
    "Confidence": 99.99979400634766,
    "Pose": {
      "Yaw": 6.557040691375732,
      "Roll": -7.316643714904785,
      "Pitch": 9.272967338562012
    },
    "Quality": {
      "Sharpness": 83.23492431640625,
      "Brightness": 78.83267974853516
    },
    "Landmarks": [
      {
        "Y": 0.3625510632991791,
        "X": 0.48898839950561523,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.35366007685661316,
        "X": 0.5313721299171448,
```



```
        "Type": "eyeRight"
      },
      {
        "Y": 0.3894785940647125,
        "X": 0.5173314809799194,
        "Type": "nose"
      },
      {
        "Y": 0.44889405369758606,
        "X": 0.5020005702972412,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.4408611059188843,
        "X": 0.5351271629333496,
        "Type": "mouthRight"
      }
    ]
  },
  "Name": "Celeb B",
  "Urls": [
    "www.imdb.com/name/bbbbbbbbbb"
  ],
  "Id": "2222222"
},
{
  "MatchConfidence": 100.0,
  "Face": {
    "BoundingBox": {
      "Width": 0.12416666746139526,
      "Top": 0.2968254089355469,
      "Left": 0.2150000035762787,
      "Height": 0.23650793731212616
    },
    "Confidence": 99.99958801269531,
    "Pose": {
      "Yaw": 7.801797866821289,
      "Roll": -8.326810836791992,
      "Pitch": 7.844768047332764
    },
    "Quality": {
      "Sharpness": 86.93206024169922,
      "Brightness": 79.81291198730469
    }
  },
}
```

```
    "Landmarks": [
      {
        "Y": 0.4027804136276245,
        "X": 0.2575301229953766,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.3934555947780609,
        "X": 0.2956969439983368,
        "Type": "eyeRight"
      },
      {
        "Y": 0.4309830069541931,
        "X": 0.2837020754814148,
        "Type": "nose"
      },
      {
        "Y": 0.48186683654785156,
        "X": 0.26812544465065,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.47338807582855225,
        "X": 0.29905644059181213,
        "Type": "mouthRight"
      }
    ]
  },
  "Name": "Celeb C",
  "Urls": [
    "www.imdb.com/name/ccccccccc"
  ],
  "Id": "3333333"
},
{
  "MatchConfidence": 97.0,
  "Face": {
    "BoundingBox": {
      "Width": 0.11916666477918625,
      "Top": 0.3698412775993347,
      "Left": 0.008333333767950535,
      "Height": 0.22698412835597992
    },
    "Confidence": 99.99999237060547,
```

```
    "Pose": {
      "Yaw": 16.38478660583496,
      "Roll": -1.0260354280471802,
      "Pitch": 5.975185394287109
    },
    "Quality": {
      "Sharpness": 83.23492431640625,
      "Brightness": 61.408443450927734
    },
    "Landmarks": [
      {
        "Y": 0.4632347822189331,
        "X": 0.049406956881284714,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.46388113498687744,
        "X": 0.08722897619009018,
        "Type": "eyeRight"
      },
      {
        "Y": 0.5020678639411926,
        "X": 0.0758260041475296,
        "Type": "nose"
      },
      {
        "Y": 0.544157862663269,
        "X": 0.054029736667871475,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.5463630557060242,
        "X": 0.08464983850717545,
        "Type": "mouthRight"
      }
    ]
  },
  "Name": "Celeb D",
  "Urls": [
    "www.imdb.com/name/ddddddddd"
  ],
  "Id": "4444444"
}
```

```
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[イメージ内の有名人の認識](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RecognizeCelebrities](#)」の「」を参照してください。

search-faces-by-image

次のコード例は、search-faces-by-image を使用する方法を示しています。

詳細については、「[顔を検索する \(イメージ\)](#)」を参照してください。

AWS CLI

画像内の最大の顔と一致するコレクション内の顔を検索するには

次の search-faces-by-image コマンドは、指定された画像内の最大の顔と一致するコレクション内の顔を検索します。

```
aws rekognition search-faces-by-image \
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePerson.jpg"}}' \
  --collection-id MyFaceImageCollection

{
  "SearchedFaceBoundingBox": {
    "Width": 0.18562500178813934,
    "Top": 0.1618015021085739,
    "Left": 0.5575000047683716,
    "Height": 0.24770642817020416
  },
  "SearchedFaceConfidence": 99.993408203125,
  "FaceMatches": [
    {
      "Face": {
        "BoundingBox": {
          "Width": 0.18562500178813934,
          "Top": 0.1618019938468933,
          "Left": 0.5575000047683716,
          "Height": 0.24770599603652954
        },
        "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
```

```
    "ExternalImageId": "example-image.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
  },
  "Similarity": 99.97913360595703
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618019938468933,
      "Left": 0.5575000047683716,
      "Height": 0.24770599603652954
    },
    "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
    "ExternalImageId": "image3.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
  },
  "Similarity": 99.97913360595703
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.41499999165534973,
      "Top": 0.09187500178813934,
      "Left": 0.28083300590515137,
      "Height": 0.3112500011920929
    },
    "FaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",
    "ExternalImageId": "image2.jpg",
    "Confidence": 99.99769592285156,
    "ImageId": "a294da46-2cb1-5cc4-9045-61d7ca567662"
  },
  "Similarity": 99.18069458007812
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.48166701197624207,
      "Top": 0.20999999344348907,
      "Left": 0.21250000596046448,
      "Height": 0.36125001311302185
    },
```

```
    "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",
    "ExternalImageId": "image1.jpg",
    "Confidence": 99.99949645996094,
    "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"
  },
  "Similarity": 98.66607666015625
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5349419713020325,
      "Top": 0.29124999046325684,
      "Left": 0.16389399766921997,
      "Height": 0.40187498927116394
    },
    "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
    "ExternalImageId": "image9.jpg",
    "Confidence": 99.99979400634766,
    "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
  },
  "Similarity": 98.24278259277344
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5307819843292236,
      "Top": 0.2862499952316284,
      "Left": 0.1564060002565384,
      "Height": 0.3987500071525574
    },
    "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
    "ExternalImageId": "image10.jpg",
    "Confidence": 99.99970245361328,
    "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
  },
  "Similarity": 98.10665893554688
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5074880123138428,
      "Top": 0.3774999976158142,
      "Left": 0.18302799761295319,
      "Height": 0.3812499940395355
```

```
    },
    "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
    "ExternalImageId": "image6.jpg",
    "Confidence": 99.99930572509766,
    "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
  },
  "Similarity": 98.10526275634766
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5574039816856384,
      "Top": 0.37187498807907104,
      "Left": 0.14559100568294525,
      "Height": 0.4181250035762787
    },
    "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
    "ExternalImageId": "image5.jpg",
    "Confidence": 99.99960327148438,
    "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
  },
  "Similarity": 97.94659423828125
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5773710012435913,
      "Top": 0.34437501430511475,
      "Left": 0.12396000325679779,
      "Height": 0.4337500035762787
    },
    "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
    "ExternalImageId": "image8.jpg",
    "Confidence": 100.0,
    "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
  },
  "Similarity": 97.93476867675781
}
],
"FaceModelVersion": "3.0"
}
```

詳細については、「Amazon Rekognition 開発者ガイド」の「[画像付きの顔を検索する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchFacesByImage](#)」の「」を参照してください。

search-faces

次のコード例は、search-faces を使用する方法を示しています。

詳細については、[顔 \(フェイス ID\) を検索する](#) を参照してください。

AWS CLI

顔 ID に一致するコレクション内の顔を検索するには

次の search-faces コマンドは、指定され顔 ID に一致するコレクション内の顔を検索します。

```
aws rekognition search-faces \  
  --face-id 8d3cfc70-4ba8-4b36-9644-90fba29c2dac \  
  --collection-id MyCollection
```

出力:

```
{  
  "SearchedFaceId": "8d3cfc70-4ba8-4b36-9644-90fba29c2dac",  
  "FaceModelVersion": "3.0",  
  "FaceMatches": [  
    {  
      "Face": {  
        "BoundingBox": {  
          "Width": 0.48166701197624207,  
          "Top": 0.20999999344348907,  
          "Left": 0.21250000596046448,  
          "Height": 0.36125001311302185  
        },  
        "FaceId": "bd4ceb4d-9acc-4ab7-8ef8-1c2d2ba0a66a",  
        "ExternalImageId": "image1.jpg",  
        "Confidence": 99.99949645996094,  
        "ImageId": "5e1a7588-e5a0-5ee3-bd00-c642518dfe3a"  
      },  
      "Similarity": 99.30997467041016  
    },  
  ],  
}
```



```
{
  "Face": {
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618019938468933,
      "Left": 0.5575000047683716,
      "Height": 0.24770599603652954
    },
    "FaceId": "ce7ed422-2132-4a11-ab14-06c5c410f29f",
    "ExternalImageId": "example-image.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8d67061e-90d2-598f-9fbd-29c8497039c0"
  },
  "Similarity": 99.24862670898438
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.18562500178813934,
      "Top": 0.1618019938468933,
      "Left": 0.5575000047683716,
      "Height": 0.24770599603652954
    },
    "FaceId": "13692fe4-990a-4679-b14a-5ac23d135eab",
    "ExternalImageId": "image3.jpg",
    "Confidence": 99.99340057373047,
    "ImageId": "8df18239-9ad1-5acd-a46a-6581ff98f51b"
  },
  "Similarity": 99.24862670898438
},
{
  "Face": {
    "BoundingBox": {
      "Width": 0.5349419713020325,
      "Top": 0.29124999046325684,
      "Left": 0.16389399766921997,
      "Height": 0.40187498927116394
    },
    "FaceId": "745f7509-b1fa-44e0-8b95-367b1359638a",
    "ExternalImageId": "image9.jpg",
    "Confidence": 99.99979400634766,
    "ImageId": "67a34327-48d1-5179-b042-01e52ccfeada"
  },
  "Similarity": 96.73158264160156
}
```

```
  },
  {
    "Face": {
      "BoundingBox": {
        "Width": 0.5307819843292236,
        "Top": 0.2862499952316284,
        "Left": 0.1564060002565384,
        "Height": 0.3987500071525574
      },
      "FaceId": "2eb5f3fd-e2a9-4b1c-a89f-afa0a518fe06",
      "ExternalImageId": "image10.jpg",
      "Confidence": 99.99970245361328,
      "ImageId": "3c314792-197d-528d-bbb6-798ed012c150"
    },
    "Similarity": 96.48291015625
  },
  {
    "Face": {
      "BoundingBox": {
        "Width": 0.5074880123138428,
        "Top": 0.3774999976158142,
        "Left": 0.18302799761295319,
        "Height": 0.3812499940395355
      },
      "FaceId": "086261e8-6deb-4bc0-ac73-ab22323cc38d",
      "ExternalImageId": "image6.jpg",
      "Confidence": 99.99930572509766,
      "ImageId": "ae1593b0-a8f6-5e24-a306-abf529e276fa"
    },
    "Similarity": 96.43287658691406
  },
  {
    "Face": {
      "BoundingBox": {
        "Width": 0.5574039816856384,
        "Top": 0.37187498807907104,
        "Left": 0.14559100568294525,
        "Height": 0.4181250035762787
      },
      "FaceId": "11c4bd3c-19c5-4eb8-aecc-24feb93a26e1",
      "ExternalImageId": "image5.jpg",
      "Confidence": 99.99960327148438,
      "ImageId": "80739b4d-883f-5b78-97cf-5124038e26b9"
    },
  },
```

```
    "Similarity": 95.25305938720703
  },
  {
    "Face": {
      "BoundingBox": {
        "Width": 0.5773710012435913,
        "Top": 0.34437501430511475,
        "Left": 0.12396000325679779,
        "Height": 0.4337500035762787
      },
      "FaceId": "57189455-42b0-4839-a86c-abda48b13174",
      "ExternalImageId": "image8.jpg",
      "Confidence": 100.0,
      "ImageId": "0aff2f37-e7a2-5dbc-a3a3-4ef6ec18eaa0"
    },
    "Similarity": 95.22837829589844
  }
]
}
```

詳細については、「Amazon Rekognition デイベロッパーガイド」の「[Face ID で顔を検索する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchFaces](#)」の「」を参照してください。

start-celebrity-recognition

次のコード例は、start-celebrity-recognition を使用する方法を示しています。

AWS CLI

保存された動画で有名人の認識を開始するには

次のstart-celebrity-recognitionコマンドは、Amazon S3 バケットに保存されている指定されたビデオファイル内の有名人を検索するジョブを開始します。

```
aws rekognition start-celebrity-recognition \
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

出力:

```
{
```

```
"JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」](#)の「[ストアドビデオの有名人の認識](#)」を参照してください。 Amazon Rekognition

- API 詳細については、AWS CLI 「[コマンドリファレンスStartCelebrityRecognition](#)」の「」を参照してください。

start-content-moderation

次の例は、start-content-moderation を使用する方法を説明しています。

AWS CLI

保存された動画内の安全でないコンテンツの認識を開始するには

次のstart-content-moderationコマンドは、Amazon S3 バケットに保存されている指定されたビデオファイル内の安全でないコンテンツを検出するジョブを開始します。

```
aws rekognition start-content-moderation \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

出力:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

詳細については、[Amazon Rekognition デベロッパーガイド](#)の「[安全でないストアドビデオの検出](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスStartContentModeration](#)」の「」を参照してください。

start-face-detection

次のコード例は、start-face-detection を使用する方法を示しています。

AWS CLI

ビデオ内の顔を検出するには

次のstart-face-detectionコマンドは、Amazon S3 バケットに保存されている指定されたビデオファイルの顔を検出するジョブを開始します。

```
aws rekognition start-face-detection
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

出力:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」の「ストアドビデオ内の顔の検出」](#)を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[StartFaceDetection](#)」の「」を参照してください。

start-face-search

次のコード例は、start-face-search を使用する方法を示しています。

AWS CLI

ビデオで検出された顔と一致するコレクション内の顔を検索するには

次のstart-face-searchコマンドは、Amazon S3 バケット内の指定されたビデオファイルで検出された顔に一致するコレクション内の顔を検索するジョブを開始します。

```
aws rekognition start-face-search \
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}" \
  --collection-id collection
```

出力:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」の「顔の保存済み動画の検索」](#)を参照してください。Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[StartFaceSearch](#)」の「」を参照してください。

start-label-detection

次のコード例は、start-label-detection を使用する方法を示しています。

AWS CLI

ビデオ内のオブジェクトとシーンを検出するには

次のstart-label-detectionコマンドは、Amazon S3 バケットに保存されている指定されたビデオファイル内のオブジェクトとシーンを検出するジョブを開始します。

```
aws rekognition start-label-detection \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

出力:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

詳細については、[「Amazon Rekognition デベロッパーガイド」の「動画内のラベルの検出」](#)を参照してください。 Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[StartLabelDetection](#)」の「」を参照してください。

start-person-tracking

次の例は、start-person-tracking を使用する方法を説明しています。

AWS CLI

保存された動画内のユーザーのパスを開始するには

次のstart-person-trackingコマンドは、Amazon S3 バケットに保存されている指定されたビデオファイルでユーザーが取るパスを追跡するジョブを開始します。

```
aws rekognition start-person-tracking \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

出力:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

詳細については、「Amazon Rekognition デベロッパーガイド」の「[People Pathing](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartPersonTracking](#)」の「」を参照してください。

start-stream-processor

次の例は、start-stream-processor を使用する方法を説明しています。

AWS CLI

ストリームプロセッサを起動するには

次のstart-stream-processorコマンドは、指定されたビデオストリームプロセッサを起動します。

```
aws rekognition start-stream-processor \  
  --name my-stream-processor
```

このコマンドでは何も出力されません。

詳細については、「[Amazon Rekognition デベロッパーガイド](#)」の「[ストリーミングビデオの使用](#)」を参照してください。 Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[StartStreamProcessor](#)」の「」を参照してください。

stop-stream-processor

次の例は、stop-stream-processor を使用する方法を説明しています。

AWS CLI

実行中のストリームプロセッサを停止するには

次の`stop-stream-processor`コマンドは、指定された実行中のストリームプロセッサを停止します。

```
aws rekognition stop-stream-processor \  
  --name my-stream-processor
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Rekognition デベロッパーガイド」の「ストリーミングビデオの使用」](#)を参照してください。 Amazon Rekognition

- API 詳細については、AWS CLI 「コマンドリファレンス[StopStreamProcessor](#)」の「」を参照してください。

AWS RAM を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS RAM。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-resource-share-invitation

次のコード例は、`accept-resource-share-invitation` を使用する方法を示しています。

AWS CLI

リソース共有の招待を受け入れるには

次の`accept-resource-share-invitation`例では、指定されたリソース共有の招待を受け入れます。招待されたアカウントのプリンシパルは、共有内のリソースの使用をすぐに開始できます。

```
aws ram accept-resource-share-invitation \
  --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-
  share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

出力:

```
{
  "resourceShareInvitation": {
    "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111111111111:resource-
  share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE",
    "resourceShareName": "MyLicenseShare",
    "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-
  share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
    "senderAccountId": "111111111111",
    "receiverAccountId": "222222222222",
    "invitationTimestamp": "2021-09-22T15:07:35.620000-07:00",
    "status": "ACCEPTED"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptResourceShareInvitation](#)」の「」を参照してください。

`associate-resource-share-permission`

次のコード例は、`associate-resource-share-permission` を使用する方法を示しています。

AWS CLI

RAM マネージドアクセス許可をリソース共有に関連付けるには

次の`associate-resource-share-permission`例では、関連するリソースタイプの既存の管理アクセス許可を、指定された管理アクセス許可に置き換えます。関連するリソースタイプのすべてのリソースへのアクセスは、新しいアクセス許可によって管理されます。

```
aws ram associate-resource-share-permission \  
  --permission-arn arn:aws:ram::aws:permission/  
AWSRAMPermissionGlueDatabaseReadWrite \  
  --replace \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-  
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

出力:

```
{  
  "returnValue": true  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateResourceSharePermission](#)」の「」を参照してください。

associate-resource-share

次のコード例は、associate-resource-share を使用する方法を示しています。

AWS CLI

例 1: リソースをリソース共有に関連付けるには

次のassociate-resource-share例では、指定されたリソース共有にライセンス設定を追加します。

```
aws ram associate-resource-share \  
  --resource-share arn:aws:ram:us-west-2:123456789012:resource-  
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \  
  --resource-arns arn:aws:license-manager:us-west-2:123456789012:license-  
configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE
```

出力:

```
{  
  "resourceShareAssociations": [  
    {  
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE",
```

```

        "associatedEntity": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
        "associationType": "RESOURCE",
        "status": "ASSOCIATING",
        "external": false
    }
]
}

```

例 2: プリンシパルをリソース共有に関連付けるには

次の `associate-resource-share` 例では、指定された組織単位内のすべてのアカウントに、指定されたリソース共有へのアクセスを許可します。

```

aws ram associate-resource-share \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
  --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-
rEXAMPLE

```

出力:

```

{
  "resourceShareAssociations": [
    {
      "status": "ASSOCIATING",
      "associationType": "PRINCIPAL",
      "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
      "external": false,
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE"
    }
  ]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateResourceShare](#)」の「」を参照してください。

create-resource-share

次のコード例は、`create-resource-share` を使用する方法を示しています。

AWS CLI

例 1: リソース共有を作成するには

次のcreate-resource-share例では、指定された名前で空のリソース共有を作成します。リソース、プリンシパル、およびアクセス許可を共有に個別に追加する必要があります。

```
aws ram create-resource-share \  
  --name MyNewResourceShare
```

出力:

```
{  
  "resourceShare": {  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/4476c27d-8feb-4b21-afe9-7de23EXAMPLE",  
    "name": "MyNewResourceShare",  
    "owningAccountId": "123456789012",  
    "allowExternalPrincipals": true,  
    "status": "ACTIVE",  
    "creationTime": 1634586271.302,  
    "lastUpdatedTime": 1634586271.302  
  }  
}
```

例 2: AWS アカウントをプリンシパルとしてリソース共有を作成するには

次のcreate-resource-share例では、リソース共有を作成し、指定された AWS アカウント (222222222222) へのアクセスを許可します。指定されたプリンシパルが同じ AWS Organization の一部でない場合、招待が送信され、アクセスが許可される前に承諾される必要があります。

```
aws ram create-resource-share \  
  --name MyNewResourceShare \  
  --principals 222222222222
```

例 3: AWS 組織に制限されたリソース共有を作成するには

次のcreate-resource-share例では、アカウントがメンバーである AWS Organization 内のアカウントに制限されたリソース共有を作成し、指定された OU をプリンシパルとして追加します。その OU 内のすべてのアカウントは、リソース共有のリソースを使用できます。

```
aws ram create-resource-share \  
  --name MyNewResourceShare \  
  --no-allow-external-principals \  
  --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-  
rEXAMPLE
```

出力:

```
{  
  "resourceShare": {  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",  
    "name": "MyNewResourceShare",  
    "owningAccountId": "123456789012",  
    "allowExternalPrincipals": false,  
    "status": "ACTIVE",  
    "creationTime": 1634587042.49,  
    "lastUpdatedTime": 1634587042.49  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResourceShare](#)」の「」を参照してください。

delete-resource-share

次の例は、delete-resource-share を使用する方法を説明しています。

AWS CLI

リソース共有を削除するには

次のdelete-resource-share例では、指定されたリソース共有を削除します。

```
aws ram delete-resource-share \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

次の出力は成功を示しています。

```
{
```

```
"returnValue": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResourceShare](#)」の「」を参照してください。

disassociate-resource-share-permission

次のコード例は、disassociate-resource-share-permission を使用する方法を示しています。

AWS CLI

リソース共有からリソースタイプのRAM管理アクセス許可を削除するには

次のdisassociate-resource-share-permission例では、指定されたリソース共有からGlue データベースのRAMマネージドアクセス許可を削除します。

```
aws ram disassociate-resource-share-permission \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE \
  --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite
```

出力:

```
{
  "returnValue": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateResourceSharePermission](#)」の「」を参照してください。

disassociate-resource-share

次の例は、disassociate-resource-share を使用する方法を説明しています。

AWS CLI

リソース共有からリソースを削除するには

次のdisassociate-resource-share例では、指定されたリソース、この場合はVPCサブネットを指定されたリソース共有から削除します。リソース共有にアクセスできるプリンシパルは、そのリソースに対してオペレーションを実行できなくなります。

```
aws ram disassociate-resource-share \  
  --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0250c25a1fEXAMPLE \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

出力:

```
{  
  "resourceShareAssociations": [  
    {  
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",  
      "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0250c25a1fEXAMPLE",  
      "associationType": "RESOURCE",  
      "status": "DISASSOCIATING",  
      "external": false  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateResourceShare](#)」の「」を参照してください。

enable-sharing-with-aws-organization

次の例は、enable-sharing-with-aws-organization を使用する方法を説明しています。

AWS CLI

AWS Organizations 間でリソース共有を有効にするには

次のenable-sharing-with-aws-organization例では、組織および組織単位間でリソース共有を有効にします。

```
aws ram enable-sharing-with-aws-organization
```

次の出力は、正しく処理されたことを示しています。

```
{
  "returnValue": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [EnableSharingWithAwsOrganization](#)」の「」を参照してください。

get-permission

次のコード例は、get-permission を使用する方法を示しています。

AWS CLI

RAM マネージドアクセス許可の詳細を取得するには

次の get-permission 例では、指定された RAM マネージドアクセス許可のデフォルトバージョンの詳細を表示します。

```
aws ram get-permission \
  --permission-arn arn:aws:ram::aws:permission/  
AWSRAMPermissionGlueTableReadWriteForDatabase
```

出力:

```
{
  "permission": {
    "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
    "version": "2",
    "defaultVersion": true,
    "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
    "resourceType": "glue:Database",
    "permission": [{"Effect": "Allow", "Action": ["glue:GetTable
\", \"glue:UpdateTable\", \"glue>DeleteTable\", \"glue:BatchDeleteTable\",
\", \"glue:BatchDeleteTableVersion\", \"glue:GetTableVersion\", \"glue:GetTableVersions
\", \"glue:GetPartition\", \"glue:GetPartitions\", \"glue:BatchGetPartition\",
\", \"glue:BatchCreatePartition\", \"glue>CreatePartition\", \"glue:UpdatePartition
\", \"glue:BatchDeletePartition\", \"glue>DeletePartition\", \"glue:GetTables\",
\", \"glue:SearchTables\" ]}],
    "creationTime": 1624912434.431,
    "lastUpdatedTime": 1624912434.431,
  }
}
```



```
    "isResourceTypeDefault": false
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPermission](#)」の「」を参照してください。

get-resource-policies

次の例は、get-resource-policies を使用する方法を説明しています。

AWS CLI

リソースのポリシーを取得するには

次のget-resource-policies例は、リソース共有に関連付けられた指定されたリソースのリソースベースのアクセス許可ポリシーを示しています。

```
aws ram get-resource-policies \
  --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE
```

出力:

```
{
  "policies": [
    [{"Version": "2008-10-17", "Statement": [{"Sid": "RamStatement1",
      "Effect": "Allow", "Principal": {"AWS": []}, "Action": ["ec2:RunInstances",
      "ec2:CreateNetworkInterface", "ec2:DescribeSubnets"], "Resource":
      "arn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1fEXAMPLE"}]}]
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourcePolicies](#)」の「」を参照してください。

get-resource-share-associations

次の例は、get-resource-share-associations を使用する方法を説明しています。

AWS CLI

例 1: すべてのリソースタイプのすべてのリソース関連付けを一覧表示するには

次のget-resource-share-associations例では、すべてのリソース共有のすべてのリソースタイプのリソース関連付けを一覧表示します。

```
aws ram get-resource-share-associations \
  --association-type RESOURCE
```

出力:

```
{
  "resourceShareAssociations": [
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
      "associatedEntity": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE",
      "resourceShareName": "MySubnetShare",
      "associationType": "RESOURCE",
      "status": "ASSOCIATED",
      "creationTime": 1565303590.973,
      "lastUpdatedTime": 1565303591.695,
      "external": false
    },
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/8167bdfe-4480-4a01-8632-315e0EXAMPLE",
      "associatedEntity": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-36be0485f5ae379cc74cf8e92EXAMPLE",
      "resourceShareName": "MyLicenseShare",
      "associationType": "RESOURCE",
      "status": "ASSOCIATED",
      "creationTime": 1632342958.457,
      "lastUpdatedTime": 1632342958.907,
      "external": false
    }
  ]
}
```

例 2: リソース共有のプリンシパル関連付けを一覧表示するには

次のget-resource-share-associations例では、指定されたリソース共有のみのプリンシパル関連付けのみを一覧表示します。

```
aws ram get-resource-share-associations \
  --resource-share-arns arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE \
  --association-type PRINCIPAL
```

出力:

```
{
  "resourceShareAssociations": [
    {
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
      "resourceShareName": "MyNewResourceShare",
      "associatedEntity": "arn:aws:organizations::123456789012:ou/
o-63bEXAMPLE/ou-46xi-rEXAMPLE",
      "associationType": "PRINCIPAL",
      "status": "ASSOCIATED",
      "creationTime": 1634587042.49,
      "lastUpdatedTime": 1634587044.291,
      "external": false
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceShareAssociations](#)」の「」を参照してください。

get-resource-share-invitations

次の例は、get-resource-share-invitations を使用する方法を説明しています。

AWS CLI

リソース共有の招待を一覧表示するには

次のget-resource-share-invitations例では、現在のリソース共有の招待を一覧表示します。

```
aws ram get-resource-share-invitations
```

出力:

```
{
  "resourceShareInvitations": [
    {
      "resourceShareInvitationArn": "arn:aws:ram:us-
west2-1:111111111111:resource-share-invitation/32b639f0-14b8-7e8f-55ea-
e6117EXAMPLE",
      "resourceShareName": "project-resource-share",
      "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/
fcb639f0-1449-4744-35bc-a983fEXAMPLE",
      "senderAccountId": "111111111111",
      "receiverAccountId": "222222222222",
      "invitationTimestamp": 1565312166.258,
      "status": "PENDING"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceShareInvitations](#)」の「」を参照してください。

get-resource-shares

次のコード例は、get-resource-shares を使用する方法を示しています。

AWS CLI

例 1: 所有しているリソース共有を一覧表示し、他のユーザーと共有するには

次のget-resource-shares例では、 が作成し、他のユーザーと共有しているリソース共有を一覧表示します。

```
aws ram get-resource-shares \
  --resource-owner SELF
```

出力:

```
{
```

```
"resourceShares": [  
  {  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/3ab63985-99d9-1cd2-7d24-75e93EXAMPLE",  
    "name": "my-resource-share",  
    "owningAccountId": "123456789012",  
    "allowExternalPrincipals": false,  
    "status": "ACTIVE",  
    "tags": [  
      {  
        "key": "project",  
        "value": "lima"  
      }  
    ]  
    "creationTime": 1565295733.282,  
    "lastUpdatedTime": 1565295733.282  
  },  
  {  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",  
    "name": "my-resource-share",  
    "owningAccountId": "123456789012",  
    "allowExternalPrincipals": true,  
    "status": "ACTIVE",  
    "creationTime": 1565295733.282,  
    "lastUpdatedTime": 1565295733.282  
  }  
]
```

例 2: 他のユーザーが所有し、共有しているリソース共有を一覧表示するには

次の`get-resource-shares`例では、他のユーザーが作成して共有したリソース共有を一覧表示します。この例では、何もありません。

```
aws ram get-resource-shares \  
--resource-owner OTHER-ACCOUNTS
```

出力:

```
{  
  "resourceShares": []
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResourceShares](#)」の「」を参照してください。

list-pending-invitation-resources

次のコード例は、list-pending-invitation-resources を使用する方法を示しています。

AWS CLI

保留中のリソース共有で利用可能なリソースを一覧表示するには

次のlist-pending-invitation-resources例では、指定された招待に関連付けられているリソース共有内のすべてのリソースを一覧表示します。

```
aws ram list-pending-invitation-resources \
  --resource-share-invitation-arn arn:aws:ram:us-west-2:123456789012:resource-
  share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

出力:

```
{
  "resources": [
    {
      "arn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-04a555b0e6EXAMPLE",
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7be8694e-095c-41ca-9ce8-7be4aEXAMPLE",
      "creationTime": 1634676051.269,
      "lastUpdatedTime": 1634676052.07,
      "status": "AVAILABLE",
      "type": "ec2:Subnet"
    },
    {
      "arn": "arn:aws:license-manager:us-west-2:123456789012:license-
configuration/lic-36be0485f5ae379cc74cf8e92EXAMPLE",
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
      "creationTime": 1624912434.431,
      "lastUpdatedTime": 1624912434.431,
    }
  ]
}
```

```
        "status": "AVAILABLE",
        "type": "license-manager:LicenseConfiguration"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPendingInvitationResources](#)」の「」を参照してください。

list-permissions

次のコード例は、list-permissions を使用する方法を示しています。

AWS CLI

使用可能なRAM管理アクセス許可を一覧表示するには

次のlist-permissions例では、AWS Glue データベースリソースタイプでのみ使用可能なすべてのRAM管理アクセス許可を一覧表示します。

```
aws ram list-permissions \
  --resource-type glue:Database
```

出力:

```
{
  "permissions": [
    {
      "arn": "arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionGlueDatabase",
      "version": "1",
      "defaultVersion": true,
      "name": "AWSRAMDefaultPermissionGlueDatabase",
      "resourceType": "glue:Database",
      "creationTime": 1592007820.935,
      "lastUpdatedTime": 1592007820.935,
      "isResourceTypeDefault": true
    },
    {
      "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueAllTablesReadWriteForDatabase",
```

```

    "version": "2",
    "defaultVersion": true,
    "name": "AWSRAMPermissionGlueAllTablesReadWriteForDatabase",
    "resourceType": "glue:Database",
    "creationTime": 1624912413.323,
    "lastUpdatedTime": 1624912413.323,
    "isResourceTypeDefault": false
  },
  {
    "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueDatabaseReadWrite",
    "version": "2",
    "defaultVersion": true,
    "name": "AWSRAMPermissionGlueDatabaseReadWrite",
    "resourceType": "glue:Database",
    "creationTime": 1624912417.4,
    "lastUpdatedTime": 1624912417.4,
    "isResourceTypeDefault": false
  },
  {
    "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
    "version": "2",
    "defaultVersion": true,
    "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
    "resourceType": "glue:Database",
    "creationTime": 1624912434.431,
    "lastUpdatedTime": 1624912434.431,
    "isResourceTypeDefault": false
  }
]
}

```

次のlist-permissions例では、すべてのリソースタイプで使用可能なRAM管理アクセス許可を表示します。

```
aws ram list-permissions
```

出力:

```

{
  "permissions": [
    {

```



```

    "arn": "arn:aws:ram::aws:permission/
AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
    "version": "1",
    "defaultVersion": true,
    "name":
"AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",
    "resourceType": "acm-pca:CertificateAuthority",
    "creationTime": 1623264861.085,
    "lastUpdatedTime": 1623264861.085,
    "isResourceTypeDefault": false
  },
  {
    "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionAppMesh",
    "version": "1",
    "defaultVersion": true,
    "name": "AWSRAMDefaultPermissionAppMesh",
    "resourceType": "appmesh:Mesh",
    "creationTime": 1589307188.584,
    "lastUpdatedTime": 1589307188.584,
    "isResourceTypeDefault": true
  },
  ...TRUNCATED FOR BREVITY...
  {
    "arn": "arn:aws:ram::aws:permission/
AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
    "version": "1",
    "defaultVersion": true,
    "name":
"AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",
    "resourceType": "acm-pca:CertificateAuthority",
    "creationTime": 1623264876.75,
    "lastUpdatedTime": 1623264876.75,
    "isResourceTypeDefault": false
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPermissions](#)」の「」を参照してください。

list-principals

次のコード例は、list-principals を使用する方法を示しています。

AWS CLI

リソースにアクセスできるプリンシパルを一覧表示するには

次のlist-principals例では、任意のリソース共有を通じて指定されたタイプのリソースにアクセスできるプリンシパルのリストを表示します。

```
aws ram list-principals \  
  --resource-type ec2:Subnet
```

出力:

```
{  
  "principals": [  
    {  
      "id": "arn:aws:organizations::123456789012:ou/o-gx7EXAMPLE/ou-29c5-  
zEXAMPLE",  
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",  
      "creationTime": 1565298209.737,  
      "lastUpdatedTime": 1565298211.019,  
      "external": false  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPrincipals](#)」の「」を参照してください。

list-resource-share-permissions

次のコード例は、list-resource-share-permissions を使用する方法を示しています。

AWS CLI

リソース共有に現在アタッチされているすべてのRAM管理アクセス許可を一覧表示するには

次のlist-resource-share-permissions例では、指定されたリソース共有にアタッチされているすべてのRAM管理アクセス許可を一覧表示します。

```
aws ram list-resource-share-permissions \  
  --resource-share-arn
```

```
--resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

出力:

```
{
  "permissions": [
    {
      "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionLicenseConfiguration",
      "version": "1",
      "resourceType": "license-manager:LicenseConfiguration",
      "status": "ASSOCIATED",
      "lastUpdatedTime": 1632342984.234
    },
    {
      "arn": "arn:aws:ram::aws:permission/AWSRAMPermissionGlueDatabaseReadWrite",
      "version": "2",
      "resourceType": "glue:Database",
      "status": "ASSOCIATED",
      "lastUpdatedTime": 1632512462.297
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceSharePermissions](#)」の「」を参照してください。

list-resource-types

次の例は、list-resource-types を使用する方法を説明しています。

AWS CLI

でサポートされているリソースタイプを一覧表示するには AWS RAM

次のlist-resource-types例では、 で現在サポートされているすべてのリソースタイプを一覧表示します AWS RAM。

```
aws ram list-resource-types
```

出力:

```
{
  "resourceTypes": [
    {
      "resourceType": "route53resolver:FirewallRuleGroup",
      "serviceName": "route53resolver"
    },
    {
      "resourceType": "ec2:LocalGatewayRouteTable",
      "serviceName": "ec2"
    },
    ...OUTPUT TRUNCATED FOR BREVITY...
    {
      "resourceType": "ec2:Subnet",
      "serviceName": "ec2"
    },
    {
      "resourceType": "ec2:TransitGatewayMulticastDomain",
      "serviceName": "ec2"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceTypes](#)」の「」を参照してください。

list-resources

次のコード例は、list-resources を使用する方法を示しています。

AWS CLI

リソース共有に関連付けられているリソースを一覧表示するには

次のlist-resources例では、指定されたリソースタイプの、指定されたリソース共有内のすべてのリソースを一覧表示します。

```
aws ram list-resources \
  --resource-type ec2:Subnet \
  --resource-owner SELF \
```

```
--resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

出力:

```
{
  "resources": [
    {
      "arn": "aarn:aws:ec2:us-west-2:123456789012:subnet/subnet-0250c25a1f4e15235",
      "type": "ec2:Subnet",
      "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
      "creationTime": 1565301545.023,
      "lastUpdatedTime": 1565301545.947
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResources](#)」の「」を参照してください。

promote-resource-share-created-from-policy

次の例は、`promote-resource-share-created-from-policy` を使用する方法を説明しています。

AWS CLI

リソースポリシーベースのリソース共有を のフル機能に昇格させるには AWS RAM

次の`promote-resource-share-created-from-policy`例では、リソースベースのポリシーをアタッチして暗黙的に作成したリソース共有を取得し、コンソールとその CLI および API オペレーションで AWS RAM 完全に機能するように変換します。

```
aws ram promote-resource-share-created-from-policy \  
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/91fa8429-2d06-4032-909a-90909EXAMPLE
```

出力:

```
{
  "returnValue": true
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PromoteResourceShareCreatedFromPolicy](#)」の「」を参照してください。

reject-resource-share-invitation

次の例は、`reject-resource-share-invitation` を使用する方法を説明しています。

AWS CLI

リソース共有の招待を拒否するには

次の`reject-resource-share-invitation`例では、指定されたリソース共有の招待を拒否します。

```
aws ram reject-resource-share-invitation \
  --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-
  share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE
```

出力:

```
"resourceShareInvitations": [
  {
    "resourceShareInvitationArn": "arn:aws:ram:us-west2-1:111111111111:resource-
    share-invitation/32b639f0-14b8-7e8f-55ea-e6117EXAMPLE",
    "resourceShareName": "project-resource-share",
    "resourceShareArn": "arn:aws:ram:us-west-2:111111111111:resource-share/
    fcb639f0-1449-4744-35bc-a983fEXAMPLE",
    "senderAccountId": "111111111111",
    "receiverAccountId": "222222222222",
    "invitationTimestamp": 1565319592.463,
    "status": "REJECTED"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス [RejectResourceShareInvitation](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソース共有にタグを追加するには

次のtag-resource例では、指定されたリソース共有limaにタグキーprojectと関連付けられた値を追加します。

```
aws ram tag-resource \  
  --tags key=project,value=Lima \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソース共有からタグを削除するには

次のuntag-resource例では、指定されたリソース共有からprojectタグキーと関連値を削除します。

```
aws ram untag-resource \  
  --tag-keys project \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-resource-share

次のコード例は、update-resource-share を使用する方法を示しています。

AWS CLI

リソース共有を更新するには

次のupdate-resource-share例では、指定されたリソース共有を変更して、AWS 組織内にならない外部プリンシパルを許可します。

```
aws ram update-resource-share \  
  --allow-external-principals \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

出力:

```
{  
  "resourceShare": {  
    "resourceShareArn": "arn:aws:ram:us-west-2:123456789012:resource-  
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",  
    "name": "my-resource-share",  
    "owningAccountId": "123456789012",  
    "allowExternalPrincipals": true,  
    "status": "ACTIVE",  
    "creationTime": 1565295733.282,  
    "lastUpdatedTime": 1565303080.023  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResourceShare](#)」の「」を参照してください。

を使用した Resource Explorer の例 AWS CLI

次のコード例は、Resource Explorer AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-default-view

次のコード例は、`associate-default-view` を使用する方法を示しています。

AWS CLI

Resource Explorer ビューをその AWS リージョンのデフォルトとして設定するには

次の`associate-default-view`例では、で指定されたビューをARN、オペレーションを呼び出す AWS リージョンのデフォルトビューに設定します。

```
aws resource-explorer-2 associate-default-view \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

出力:

```
{  
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
}
```

詳細については、AWS Resource Explorer ユーザーガイドの[AWS 「リージョンでのデフォルトビューの設定」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateDefaultView](#)」の「」を参照してください。

batch-get-view

次のコード例は、batch-get-view を使用する方法を示しています。

AWS CLI

複数の Resource Explorer ビューの詳細を取得するには

次のbatch-get-view例では、で指定された 2 つのビューの詳細を表示しますARNs。スペースを使用して、--view-arn パラメータARNs内の複数のを区切ります。

```
aws resource-explorer-2 batch-get-view \
  --view-arns arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222, \
             arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

出力:

```
{
  "Views": [
    {
      "Filters": {
        "FilterString": "service:ec2"
      },
      "IncludedProperties": [
        {
          "Name": "tags"
        }
      ],
      "LastUpdatedAt": "2022-07-13T21:33:45.249000+00:00",
      "Owner": "123456789012",
      "Scope": "arn:aws:iam::123456789012:root",
      "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    },
    {
      "Filters": {
        "FilterString": ""
      },
      "IncludedProperties": [
        {
          "Name": "tags"
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",
  "Owner": "123456789012",
  "Scope": "arn:aws:iam::123456789012:root",
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-
Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
]
"Errors": []
}
```

ビューの詳細については、[Resource Explorer ユーザーガイドの「Resource Explorer ビューについて」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetView](#)」の「」を参照してください。

create-index

次のコード例は、create-index を使用する方法を示しています。

AWS CLI

インデックスを作成して AWS リージョンで Resource Explorer を有効にするには

次のcreate-index例では、オペレーションが呼び出される AWS リージョンにローカルインデックスを作成します。は AWS CLIランダムなclient-tokenパラメータ値を自動的に生成し、値を指定 AWS しない場合、への呼び出しに含めます。

```
aws resource-explorer-2 create-index \
  --region us-east-1
```

出力:

```
{
  "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222c",
  "CreatedAt": "2022-11-01T20:00:59.149Z",
  "State": "CREATING"
}
```

ローカルインデックスを作成したら、[update-index-type](#) コマンドを実行してアカウントのアグリゲータインデックスに変換できます。

詳細については、「[Resource Explorer ユーザーガイド](#)」の AWS 「[リージョンで Resource Explorer を有効にしてリソースをインデックス化する](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンス>CreateIndex](#)」の「」を参照してください。

create-view

次の例は、create-view を使用する方法を説明しています。

AWS CLI

例 1: AWS リージョン内のインデックスにフィルタリングされていないビューを作成するには

次のcreate-view例では、指定された AWS リージョンにビューを作成し、フィルタリングなしでリージョン内のすべての結果を返します。ビューには、返された結果にオプションのタグフィールドが含まれます。このビューはアグリゲータインデックスを含むリージョンで作成されるため、Resource Explorer インデックスを含むアカウント内のすべてのリージョンの結果を含めることができます。

```
aws resource-explorer-2 create-view \  
  --view-name My-Main-View \  
  --included-properties Name=tags \  
  --region us-east-1
```

出力:

```
{  
  "View": {  
    "Filters": {  
      "FilterString": ""  
    },  
    "IncludedProperties": [  
      {  
        "Name": "tags"  
      }  
    ],  
    "LastUpdatedAt": "2022-07-13T20:34:11.314000+00:00",  
    "Owner": "123456789012",
```

```
    "Scope": "arn:aws:iam::123456789012:root",
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
  }
}
```

例 2: Amazon に関連付けられたリソースのみ返すビューを作成するには EC2

以下では、Amazon EC2サービスに関連付けられている AWS リージョン内のリソースのみを us-east-1 返すビューをリージョンに create-view 作成します。ビューには、返された結果のオプション Tags フィールドが含まれます。このビューはアグリゲータインデックスを含むリージョンで作成されるため、Resource Explorer インデックスを含むアカウント内のすべてのリージョンの結果を含めることができます。

```
aws resource-explorer-2 create-view \
  --view-name My-EC2-Only-View \
  --included-properties Name=tags \
  --filters FilterString="service:ec2" \
  --region us-east-1
```

出力:

```
{
  "View": {
    "Filters": {
      "FilterString": "service:ec2"
    },
    "IncludedProperties": [
      {
        "Name": "tags"
      }
    ],
    "LastUpdatedAt": "2022-07-13T21:35:09.059Z",
    "Owner": "123456789012",
    "Scope": "arn:aws:iam::123456789012:root",
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
  }
}
```

詳細については、AWS Resource Explorer ユーザーガイドの [「検索用のビューの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateView](#)」の「」を参照してください。

delete-index

次のコード例は、delete-index を使用する方法を示しています。

AWS CLI

インデックスを削除して AWS リージョンの Resource Explorer を無効にするには

次のdelete-index例では、リクエストを行う AWS リージョンで指定された Resource Explorer インデックスを削除します。

```
aws resource-explorer-2 delete-index \  
  --arn arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE22222 \  
  --region us-west-2
```

出力:

```
{  
  "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE22222",  
  "State": "DELETING"  
}
```

インデックスの削除の詳細については、[AWS 「Resource Explorer ユーザーガイド」の AWS 「リージョンで Resource Explorer をオフにする」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIndex](#)」の「」を参照してください。

delete-view

次の例は、delete-view を使用する方法を説明しています。

AWS CLI

Resource Explorer ビューを削除するには

次のdelete-view例では、 で指定されたビューを削除しますARN。

```
aws resource-explorer-2 delete-view \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

出力:

```
{  
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
}
```

詳細については、AWS Resource Explorer ユーザーガイドの「[ビューの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteView](#)」の「」を参照してください。

disassociate-default-view

次の例は、disassociate-default-view を使用する方法を説明しています。

AWS CLI

AWS リージョンのデフォルトの Resource Explorer ビューを削除するには

以下は、オペレーションを呼び出す AWS リージョンのデフォルトの Resource Explorer ビューdisassociate-default-viewを削除します。このオペレーションを実行した後、リージョン内のすべての検索オペレーションでビューを明示的に指定する必要があります。指定しない場合、オペレーションは失敗します。

```
aws resource-explorer-2 disassociate-default-view
```

このコマンドでは何も出力されません。

詳細については、AWS Resource Explorer ユーザーガイドの[AWS 「リージョンでのデフォルトビューの設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateDefaultView](#)」の「」を参照してください。

get-default-view

次の例は、get-default-view を使用する方法を説明しています。

AWS CLI

AWS リージョンのデフォルトビューである Resource Explorer ビューを取得するには

次のget-default-view例では、オペレーションを呼び出す AWS リージョンのデフォルトであるビューARNの を取得します。

```
aws resource-explorer-2 get-default-view
```

出力:

```
{
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/default-view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

詳細については、AWS Resource Explorer ユーザーガイドの[AWS 「リージョンでのデフォルトビューの設定」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDefaultView](#)」の「」を参照してください。

get-index

次のコード例は、get-index を使用する方法を示しています。

AWS CLI

例 1: Resource Explorer アグリゲータインデックスの詳細を取得するには

次のget-index例では、指定された AWS リージョンの Resource Explorer インデックスの詳細を表示します。指定されたリージョンにはアカウントのアグリゲータインデックスが含まれているため、出力には、このリージョンのインデックスにデータをレプリケートするリージョンが一覧表示されます。

```
aws resource-explorer-2 get-index \
```



```
--region us-east-1
```

出力:

```
{
  "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE11111",
  "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
  "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
  "ReplicatingFrom": [
    "ap-south-1",
    "us-west-2"
  ],
  "State": "ACTIVE",
  "Tags": {},
  "Type": "AGGREGATOR"
}
```

例 2: Resource Explorer ローカルインデックスの詳細を取得するには

次のget-index例では、指定された AWS リージョンの Resource Explorer インデックスの詳細を表示します。指定されたリージョンにはローカルインデックスが含まれているため、出力には、このリージョンのインデックスからデータをレプリケートするリージョンが一覧表示されません。

```
aws resource-explorer-2 get-index \  
--region us-west-2
```

出力:

```
{
  "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222",
  "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
  "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
  "ReplicatingTo": [
    "us-west-2"
  ],
  "State": "ACTIVE",
  "Tags": {},
  "Type": "LOCAL"
}
```

```
}
```

インデックスの詳細については、[「Resource Explorer ユーザーガイド」の「Resource Explorer が有効になっている AWS リージョンの確認」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIndex](#)」の「」を参照してください。

get-view

次のコード例は、get-view を使用する方法を示しています。

AWS CLI

Resource Explorer ビューの詳細を取得するには

次のget-view例では、 で指定されたビューの詳細を表示しますARN。

```
aws resource-explorer-2 get-view \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

出力:

```
{  
  "Tags" : {},  
  "View" : {  
    "Filters" : {  
      "FilterString" : "service:ec2"  
    },  
    "IncludedProperties" : [  
      {  
        "Name" : "tags"  
      }  
    ],  
    "LastUpdatedAt" : "2022-07-13T21:33:45.249Z",  
    "Owner" : "123456789012",  
    "Scope" : "arn:aws:iam::123456789012:root",  
    "ViewArn" : "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
  }  
}
```

ビューの詳細については、[Resource Explorer ユーザーガイドの「Resource Explorer ビューについて」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetView](#)」の「」を参照してください。

list-indexes

次のコード例は、list-indexes を使用する方法を示しています。

AWS CLI

Resource Explorer にインデックスがある AWS リージョンを一覧表示するには

次のlist-indexes例では、Resource Explorer にインデックスがあるすべてのリージョンのインデックスを一覧表示します。レスポンスは、各インデックスのタイプ、AWS リージョン、および を指定しますARN。

```
aws resource-explorer-2 list-indexes
```

出力:

```
{
  "Indexes": [
    {
      "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
      "Region": "us-west-2",
      "Type": "AGGREGATOR"
    },
    {
      "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
      "Region": "us-east-1",
      "Type": "LOCAL"
    },
    {
      "Arn": "arn:aws:resource-explorer-2:us-east-2:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333",
      "Region": "us-east-2",
      "Type": "LOCAL"
    }
  ],
}
```

```
{
  "Arn": "arn:aws:resource-explorer-2:us-west-1:123456789012:index/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE44444",
  "Region": "us-west-1",
  "Type": "LOCAL"
}
]
```

インデックスの詳細については、[「Resource Explorer ユーザーガイド」の「Resource Explorer が有効になっている AWS リージョンの確認」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListIndexes](#)」の「」を参照してください。

list-supported-resource-types

次の例は、list-supported-resource-types を使用する方法を説明しています。

AWS CLI

Resource Explorer にインデックスがある AWS リージョンを一覧表示するには

次のlist-supported-resource-types例では、現在 &AREXlong; でサポートされているすべてのリソースタイプを一覧表示します。レスポンスの例には、追加の呼び出しで取得できる出力が多いことを示すNextToken値が含まれています。

```
aws resource-explorer-2 list-supported-resource-types \
--max-items 10
```

出力:

```
{
  "ResourceTypes": [
    {
      "ResourceType": "cloudfront:cache-policy",
      "Service": "cloudfront"
    },
    {
      "ResourceType": "cloudfront:distribution",
      "Service": "cloudfront"
    }
  ]
}
```

```
    },
    {
      "ResourceType": "cloudfront:function",
      "Service": "cloudfront"
    },
    {
      "ResourceType": "cloudfront:origin-access-identity",
      "Service": "cloudfront"
    },
    {
      "ResourceType": "cloudfront:origin-request-policy",
      "Service": "cloudfront"
    },
    {
      "ResourceType": "cloudfront:realtime-log-config",
      "Service": "cloudfront"
    },
    {
      "ResourceType": "cloudfront:response-headers-policy",
      "Service": "cloudfront"
    },
    {
      "ResourceType": "cloudwatch:alarm",
      "Service": "cloudwatch"
    },
    {
      "ResourceType": "cloudwatch:dashboard",
      "Service": "cloudwatch"
    },
    {
      "ResourceType": "cloudwatch:insight-rule",
      "Service": "cloudwatch"
    }
  ],
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}
```

出力の次の部分を取得するには、オペレーションを再度呼び出し、前の呼び出しのNextTokenレスポンス値を の値として渡します--starting-token。NextToken がレスポンスにないまで繰り返します。

```
aws resource-explorer-2 list-supported-resource-types \
  --max-items 10 \
```

--starting-**token eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0=**

出力:

```
{
  "ResourceTypes": [
    {
      "ResourceType": "cloudwatch:metric-stream",
      "Service": "cloudwatch"
    },
    {
      "ResourceType": "dynamodb:table",
      "Service": "dynamodb"
    },
    {
      "ResourceType": "ec2:capacity-reservation",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:capacity-reservation-fleet",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:client-vpn-endpoint",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:customer-gateway",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:dedicated-host",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:dhcp-options",
      "Service": "ec2"
    },
    {
      "ResourceType": "ec2:egress-only-internet-gateway",
      "Service": "ec2"
    },
  ],
}
```

```
{
  "ResourceType": "ec2:elastic-gpu",
  "Service": "ec2"
},
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyMH0="
}
```

インデックスの詳細については、[Resource Explorer ユーザーガイドの「Resource Explorer が有効になっている AWS リージョンの確認」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSupportedResourceTypes](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

Resource Explorer ビューまたはインデックスにアタッチされたタグを一覧表示するには

次のlist-tags-for-resource例では、指定された で表示するためにアタッチされたタグキーと値のペアを一覧表示しますARN。リソースを含む AWS リージョンから オペレーションを呼び出す必要があります。

```
aws resource-explorer-2 list-tags-for-resource \
  --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111
```

出力:

```
{
  "Tags": {
    "application": "MainCorpApp",
    "department": "1234"
  }
}
```

ビューのタグ付けの詳細については、AWS Resource Explorer ユーザーガイドの[「アクセスコントロールのビューのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-views

次の例は、list-views を使用する方法を説明しています。

AWS CLI

AWS リージョンで使用できる Resource Explorer ビューを一覧表示するには

次のlist-views例では、オペレーションを呼び出すリージョンで使用可能なすべてのビューを一覧表示します。

```
aws resource-explorer-2 list-views
```

出力:

```
{
  "Views": [
    "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111",
    "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Default-All-Resources-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
    "arn:aws:resource-explorer-2:us-east-1:123456789012:view/Production-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE33333"
  ]
}
```

ビューの詳細については、[「Resource Explorer ユーザーガイド」の「Resource Explorer ビューについて」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListViews](#)」の「」を参照してください。

search

次のコード例は、search を使用する方法を示しています。

AWS CLI

例 1: デフォルトビューを使用して検索するには

次のsearch例では、サービスに関連付けられている、指定された内のすべてのリソースを表示します。検索では、リージョンのデフォルトビューを使用します。レスポンスの例には、追加の呼び出しで取得できる出力が多いことを示すNextToken値が含まれています。

```
aws resource-explorer-2 search \  
  --query-string "service:iam"
```

出力:

```
{  
  "Count": {  
    "Complete": true,  
    "TotalResources": 55  
  },  
  "NextToken":  
  "AG9V0EF1KLEXAMPLE0hJHVwo5chEXAMPLER5XiEpNrgsEXAMPLE...b0Cm0F0ryHEXAMPLE",  
  "Resources": [{  
    "Arn": "arn:aws:iam::123456789012:policy/service-role/Some-Policy-For-A-  
Service-Role",  
    "LastReportedAt": "2022-07-21T12:34:42Z",  
    "OwningAccountId": "123456789012",  
    "Properties": [],  
    "Region": "global",  
    "ResourceType": "iam:policy",  
    "Service": "iam"  
  }, {  
    "Arn": "arn:aws:iam::123456789012:policy/service-role/Another-Policy-For-A-  
Service-Role",  
    "LastReportedAt": "2022-07-21T12:34:42Z",  
    "OwningAccountId": "123456789012",  
    "Properties": [],  
    "Region": "global",  
    "ResourceType": "iam:policy",  
    "Service": "iam"  
  }, {  
    ... TRUNCATED FOR BREVITY ...  
  }],  
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/my-default-  
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
}
```

例 2: 指定されたビューを使用して検索するには

次の検索search例では、指定されたビューで表示される、指定された AWS リージョン内のすべてのリソース (「*」) を表示します。結果には、ビューにアタッチされたフィルターEC2が原因で、Amazon に関連付けられたリソースのみが含まれます。

```
aws resource-explorer-2 search \  
  -- query-string "*" \  
  -- view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-view/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

出力:

```
HTTP/1.1 200 OK  
Date: Tue, 01 Nov 2022 20:00:59 GMT  
Content-Type: application/json  
Content-Length: <PayloadSizeBytes>  
  
{  
  "Count": {  
    "Complete": true,  
    "TotalResources": 67  
  },  
  "Resources": [{  
    "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/acl-1a2b3c4d",  
    "LastReportedAt": "2022-07-21T18:52:02Z",  
    "OwningAccountId": "123456789012",  
    "Properties": [{  
      "Data": [{  
        "Key": "Department",  
        "Value": "AppDevelopment"  
      }], {  
        "Key": "Environment",  
        "Value": "Production"  
      }],  
    "LastReportedAt": "2021-11-15T14:48:29Z",  
    "Name": "tags"  
  }],  
  "Region": "us-east-1",  
  "ResourceType": "ec2:network-acl",  
  "Service": "ec2"  
}, {  
  "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/subnet-1a2b3c4d",  
  "LastReportedAt": "2022-07-21T21:22:23Z",  
  "OwningAccountId": "123456789012",
```

```
    "Properties": [{
      "Data": [{
        "Key": "Department",
        "Value": "AppDevelopment"
      }, {
        "Key": "Environment",
        "Value": "Production"
      }],
      "LastReportedAt": "2021-07-29T19:02:39Z",
      "Name": "tags"
    }],
    "Region": "us-east-1",
    "ResourceType": "ec2:subnet",
    "Service": "ec2"
  }, {
    "Arn": "arn:aws:ec2:us-east-1:123456789012:dhcp-options/dopt-1a2b3c4d",
    "LastReportedAt": "2022-07-21T06:08:53Z",
    "OwningAccountId": "123456789012",
    "Properties": [{
      "Data": [{
        "Key": "Department",
        "Value": "AppDevelopment"
      }, {
        "Key": "Environment",
        "Value": "Production"
      }],
      "LastReportedAt": "2021-11-15T15:11:05Z",
      "Name": "tags"
    }],
    "Region": "us-east-1",
    "ResourceType": "ec2:dhcptions",
    "Service": "ec2"
  }, {
    ... TRUNCATED FOR BREVITY ...
  }],
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
}
```

詳細については、[AWS 「Resource Explorer ユーザーガイド」の「Resource Explorer を使用してリソースを検索する」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/resource-explorer-2/search.html>」の検索を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

Resource Explorer ビューにタグを付けるには

次のtag-resource例では、指定された のビューに、値「本番」のタグキー「環境」を追加しますARN。

```
aws resource-explorer-2 tag-resource \  
  --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \  
  --tags environment=production
```

このコマンドでは何も出力されません。

詳細については、AWS 「Resource Explorer ユーザーガイド」の「[アクセスコントロールのタグ付けビュー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

Resource Explorer ビューからタグを削除するには

次のuntag-resource例では、キー名「環境」のタグを、指定された のビューから削除しますARN。

```
aws resource-explorer-2 untag-resource \  
  --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \  
  --tag-key environment
```

```
--tag-keys environment
```

このコマンドでは何も出力されません。

詳細については、AWS「Resource Explorer ユーザーガイド」の「[アクセスコントロールのビューのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-index-type

次の例は、update-index-type を使用する方法を説明しています。

AWS CLI

Resource Explorer インデックスのタイプを変更するには

次のupdate-index-type例では、指定されたインデックスを型からlocal型に変換aggregatorとして、アカウント内のすべてのAWSリージョンでリソースを検索する機能を有効にします。更新するインデックスを含むAWSリージョンにリクエストを送信する必要があります。

```
aws resource-explorer-2 update-index-type \  
  --arn arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE11111 \  
  --type aggregator \  
  --region us-east-1
```

出力:

```
{  
  "Arn": "arn:aws:resource-explorer-2:us-east-1:123456789012:index/EXAMPLE8-90ab-  
cdef-fedc-EXAMPLE11111",  
  "LastUpdatedAt": "2022-07-13T18:41:58.799Z",  
  "State": "updating",  
  "Type": "aggregator"  
}
```

インデックスのタイプの変更の詳細については、AWS Resource Explorer ユーザーガイドの「[アグリゲータインデックスを作成してクロスリージョン検索を有効にする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateIndexType](#)」の「」を参照してください。

update-view

次のコード例は、update-view を使用する方法を示しています。

AWS CLI

例 1: Resource Explorer ビューの IncludedProperties フィールドを更新するには

次のupdate-view例では、オプションの `tags` に を追加して、指定されたビューを更新します `IncludedProperties`。このオペレーションを実行した後、このビューを使用する検索オペレーションには、結果に表示されるリソースにアタッチされたタグに関する情報が含まれます。

```
aws resource-explorer-2 update-view \  
  --included-properties Name=tags \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

出力:

```
{  
  "View": {  
    "Filters": {  
      "FilterString": ""  
    },  
    "IncludedProperties": [  
      {  
        "Name": "tags"  
      }  
    ],  
    "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",  
    "Owner": "123456789012",  
    "Scope": "arn:aws:iam::123456789012:root",  
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-  
Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
  }  
}
```

例 2: ビューにアタッチされたフィルターを更新するには

次のupdate-view例では、Amazon EC2サービスに関連付けられているリソースタイプにのみ結果を制限するフィルターを使用するように、指定されたビューを更新します。

```
aws resource-explorer-2 update-view \  
  --filters FilterString="service:ec2" \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

出力:

```
{  
  "View": {  
    "Filters": {  
      "FilterString": "service:ec2"  
    },  
    "IncludedProperties": [],  
    "LastUpdatedAt": "2022-07-19T17:41:21.710000+00:00",  
    "Owner": "123456789012",  
    "Scope": "arn:aws:iam::123456789012:root",  
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"  
  }  
}
```

ビューの詳細については、[Resource Explorer ユーザーガイドの「Resource Explorer ビューについて」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateView](#)」の「」を参照してください。

を使用したリソースグループの例 AWS CLI

次のコード例は、Resource Groups AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-group

次の例は、create-group を使用する方法を説明しています。

AWS CLI

例 1: タグベースのリソースグループを作成するには

次のcreate-group例では、現在のリージョンに Amazon EC2インスタンスのタグベースのリソースグループを作成します。これは、キー Nameと値 でタグ付けされたリソースのクエリに基づいていますWebServers。グループ名は ですtbq-WebServer。クエリは、コマンドに渡される別のJSONファイルにあります。

```
aws resource-groups create-group \  
  --name tbq-WebServer \  
  --resource-query file://query.json
```

query.json の内容:

```
{  
  "Type": "TAG_FILTERS_1_0",  
  "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\":  
  [ { \"Key\": \"Name\", \"Values\": [ \"WebServers\" ] } ] }\"  
}
```

出力:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-  
WebServer",  
    "Name": "tbq-WebServer"  
  },  
  "ResourceQuery": {  
    "Type": "TAG_FILTERS_1_0",
```



```

    "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [\"Key\": \"Name\", \"Values\": [\"WebServers\"]]}"
  }
}

```

例 2: CloudFormation スタックベースのリソースグループを作成するには

次の create-group 例では、 という名前の AWS CloudFormation スタックベースのリソースグループを作成します sampleCFNstackgroup。クエリには、Resource Groups でサポートされている指定された CloudFormation スタック内のすべての AWS リソースが含まれます。

```

aws resource-groups create-group \
  --name cbq-CFNstackgroup \
  --resource-query file://query.json

```

query.json の内容:

```

{
  "Type": "CLOUDFORMATION_STACK_1_0",
  "Query": "{\"ResourceTypeFilters\": [\"AWS::AllSupported\"], \"StackIdentifier\": \"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-z39z-11z8-97z5-500z212zz6fz\"}"
}

```

出力:

```

{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-CFNstackgroup",
    "Name": "cbq-CFNstackgroup"
  },
  "ResourceQuery": {
    "Type": "CLOUDFORMATION_STACK_1_0",
    "Query": "{\"ResourceTypeFilters\": [\"AWS::AllSupported\"], \"StackIdentifier\": \"arn:aws:cloudformation:us-east-2:123456789012:stack/MyCFNStack/1415z9z0-z39z-11z8-97z5-500z212zz6fz\"}"
  }
}

```

詳細については、「リソース [グループユーザーガイド](#)」の「[グループの作成](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateGroup](#)」の「」を参照してください。

delete-group

次のコード例は、delete-group を使用する方法を示しています。

AWS CLI

リソースグループの説明を更新するには

次のdelete-group例では、指定されたリソースグループを更新します。

```
aws resource-groups delete-group \  
  --group-name tbq-WebServer
```

出力:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:1234567890:group/tbq-  
WebServer",  
    "Name": "tbq-WebServer"  
  }  
}
```

詳細については、「リソース[グループユーザーガイド](#)」の「[グループの削除](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGroup](#)」の「」を参照してください。

get-group-query

次のコード例は、get-group-query を使用する方法を示しています。

AWS CLI

リソースグループにアタッチされたクエリを取得するには

次のget-group-query例では、指定されたリソースグループにアタッチされたクエリを表示します。

```
aws resource-groups get-group-query \  
  --group-name tbq-WebServer
```

出力:

```
{  
  "GroupQuery": {  
    "GroupName": "tbq-WebServer",  
    "ResourceQuery": {  
      "Type": "TAG_FILTERS_1_0",  
      "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [{\"Key\": \"Name\", \"Values\": [\"WebServers\"]}]}"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroupQuery](#)」の「」を参照してください。

get-group

次のコード例は、get-group を使用する方法を示しています。

AWS CLI

リソースグループに関する情報を取得するには

次のget-group例では、指定されたリソースグループの詳細を表示します。クエリをグループにアタッチするには、`get-group-query`を使用します。

```
aws resource-groups get-group \  
  --group-name tbq-WebServer
```

出力:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",  
    "Name": "tbq-WebServer",  
    "Description": "A tag-based query resource group of WebServers."  
  }  
}
```

```
}  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroup](#)」の「」を参照してください。

get-tags

次のコード例は、get-tags を使用する方法を示しています。

AWS CLI

リソースグループにアタッチされたタグを取得するには

次のget-tags例では、指定されたリソースグループ (メンバーではなくグループ自体) にアタッチされたタグキーと値のペアを表示します。

```
aws resource-groups get-tags \  
  --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer
```

出力:

```
{  
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",  
  "Tags": {  
    "QueryType": "tags",  
    "QueryResources": "ec2-instances"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTags](#)」の「」を参照してください。

list-group-resources

次の例は、list-group-resources を使用する方法を説明しています。

AWS CLI

リソースグループ内のすべてのリソースを一覧表示するには

例 1: 次の `list-resource-groups` 例では、指定されたリソースグループの一部であるすべてのリソースを一覧表示します。

```
aws resource-groups list-group-resources \  
  --group-name tbq-WebServer
```

出力:

```
{  
  "ResourceIdentifiers": [  
    {  
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/  
i-09f77fa38c12345ab",  
      "ResourceType": "AWS::EC2::Instance"  
    }  
  ]  
}
```

例 2: 次の例では、グループ内のすべてのリソースのうち、'AWS::EC2::Instance' の「resource-type」も示しています。

```
aws resource-groups list-group-resources --group-name tbq-WebServer --filters Name=resource-type,Values=AWS : EC2::Instance
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListGroupResources](#)」の「」を参照してください。

list-groups

次のコード例は、`list-groups` を使用する方法を示しています。

AWS CLI

使用可能なリソースグループを一覧表示するには

次の `list-groups` 例では、すべてのリソースグループのリストを表示します。

```
aws resource-groups list-groups
```

出力:

```
{
  "GroupIdentifiers": [
    {
      "GroupName": "tbq-WebServer",
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer3"
    },
    {
      "GroupName": "cbq-CFNStackQuery",
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-CFNStackQuery"
    }
  ],
  "Groups": [
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
      "Name": "tbq-WebServer"
    },
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-CFNStackQuery",
      "Name": "cbq-CFNStackQuery"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGroups](#)」の「」を参照してください。

list-resource-groups

次のコード例は、list-resource-groups を使用する方法を示しています。

AWS CLI

リソースグループ内のすべてのリソースを一覧表示するには

次のlist-resource-groups例では、指定されたリソースグループの一部であるすべてのリソースを一覧表示します。

```
aws resource-groups list-group-resources \
```

```
--group-name tbq-WebServer
```

出力:

```
{
  "ResourceIdentifiers": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
      "ResourceType": "AWS::EC2::Instance"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceGroups](#)」の「」を参照してください。

put-group-configuration

次のコード例は、put-group-configuration を使用する方法を示しています。

AWS CLI

サービス設定をリソースグループにアタッチするには

例 1: 次のput-group-configuration例では、リソースグループに C5または M5ファミリーのインスタンスの Amazon EC2キャパシティ予約のみを含めるように指定しています。

```
aws resource-groups put-group-configuration \
  --group MyTestGroup \
  --configuration file://config.json
```

config.json の内容:

```
[
  {
    "Type": "AWS::EC2::HostManagement",
    "Parameters": [
      {
        "Name": "allowed-host-families",
        "Values": [ "c5", "m5" ]
      }
    ]
  }
]
```

```
    },
    {
      "Name": "any-host-based-license-configuration",
      "Values": [ "true" ]
    }
  ]
},
{
  "Type": "AWS::ResourceGroups::Generic",
  "Parameters": [
    {
      "Name": "allowed-resource-types",
      "Values": [ "AWS::EC2::Host" ]
    },
    {
      "Name": "deletion-protection",
      "Values": [ "UNLESS_EMPTY" ]
    }
  ]
}
]
```

このコマンドが成功した場合、出力は生成されません。

詳細については、「[リソースグループリファレンスガイド](#)」の「[リソースグループのサービス設定API](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスPutGroupConfiguration](#)」の「」を参照してください。

search-resources

次の例は、search-resources を使用する方法を説明しています。

AWS CLI

クエリに一致するリソースを検索するには

次のsearch-resources例では、指定されたクエリに一致するすべての AWS リソースのリストを取得します。

```
aws resource-groups search-resources \
```



```
--resource-query file://query.json
```

query.json の内容:

```
{
  "Type": "TAG_FILTERS_1_0",
  "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [\"Key\": \"Patch Group\", \"Values\": [\"Dev\"]]}\"
}
```

出力:

```
{
  "ResourceIdentifiers": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-01a23bc45d67890ef",
      "ResourceType": "AWS::EC2::Instance"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [SearchResources](#)」の「」を参照してください。

tag

次の例は、tag を使用する方法を説明しています。

AWS CLI

リソースグループにタグをアタッチするには

次のtag例では、指定されたタグキーと値のペアを指定されたリソースグループ (メンバーではなくグループ自体) にアタッチします。

```
aws resource-groups tag \
  --tags QueryType=tags, QueryResources=ec2-instances \
  --arn arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer
```

出力:

```
{
  "Arn": "arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer",
  "Tags": {
    "QueryType": "tags",
    "QueryResources": "ec2-instances"
  }
}
```

詳細については、「Resource Groups ユーザーガイド」の「[タグの管理](#)」を参照してください。
AWS

- API 詳細については、「コマンドリファレンス」の「[タグ](#)」を参照してください。AWS CLI

untag

次の例は、`untag` を使用する方法を説明しています。

AWS CLI

リソースグループからタグを削除するには

次の `untags` 例では、指定されたキーを持つタグを、メンバーではなくリソースグループ自体から削除します。

```
aws resource-groups untag \
  --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer \
  --keys QueryType
```

出力:

```
{
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
  "Keys": [
    "QueryType"
  ]
}
```

詳細については、「Resource Groups ユーザーガイド」の「[タグの管理](#)」を参照してください。
AWS

- API 詳細については、「コマンドリファレンス」の「[タグ解除](#)」を参照してください。AWS CLI

update-group-query

次のコード例は、update-group-query を使用する方法を示しています。

AWS CLI

例 1: タグベースのリソースグループのクエリを更新するには

次のupdate-group-query例では、指定されたタグベースのリソースグループにアタッチされたクエリを更新します。

```
aws resource-groups update-group-query \  
  --group-name tbq-WebServer \  
  --resource-query '{"Type":"TAG_FILTERS_1_0", "Query":{"ResourceTypeFilters":["AWS::EC2::Instance"],"TagFilters":[{"Key":"Name", "Values":["WebServers"]}]}'
```

出力:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-east-2:123456789012:group/tbq-WebServer",  
    "Name": "tbq-WebServer"  
  },  
  "ResourceQuery": {  
    "Type": "TAG_FILTERS_1_0",  
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"  
  }  
}
```

詳細については、「[リソースグループユーザーガイド](#)」の「[グループの更新](#)」を参照してください。AWS

例 2: CloudFormation スタックベースのリソースグループのクエリを更新するには

次のupdate-group-query例では、指定された AWS CloudFormation スタックベースのリソースグループにアタッチされたクエリを更新します。

```
aws resource-groups update-group-query \  
  --group-name cbq-CFNstackgroup \  
  --resource-query '{"Type":"TAG_FILTERS_1_0", "Query":{"ResourceTypeFilters":["AWS::CloudFormation::Stack"],"TagFilters":[{"Key":"Name", "Values":["CFNStackGroup"]}]}'
```

```
--resource-query '{"Type": "CLOUDFORMATION_STACK_1_0", "Query":
{"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier\":
\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}]}'
```

出力:

```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/cbq-
CFNstackgroup",
    "Name": "cbq-CFNstackgroup"
  },
  "ResourceQuery": {
    "Type": "CLOUDFORMATION_STACK_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\": \"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"
  }
}
```

詳細については、「リソース[グループユーザーガイド](#)」の「[グループの更新](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateGroupQuery](#)」の「」を参照してください。

update-group

次のコード例は、update-group を使用する方法を示しています。

AWS CLI

リソースグループの説明を更新するには

次のupdate-group例では、指定されたリソースグループの説明を更新します。

```
aws resource-groups update-group \
  --group-name tbq-WebServer \
  --description "Resource group for all web server resources."
```

出力:

```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Name": "tbq-WebServer"
    "Description": "Resource group for all web server resources."
  }
}
```

詳細については、「リソース[グループユーザーガイド](#)」の「[グループの更新](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスUpdateGroup](#)」の「」を参照してください。

を使用したリソースグループのタグ付けAPIの例 AWS CLI

次のコード例は、Resource Groups Tagging AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますAPI。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

get-resources

次のコード例は、get-resources を使用する方法を示しています。

AWS CLI

タグ付きリソースのリストを取得するには

次のget-resources例では、指定されたキー名と値でタグ付けされたアカウント内のリソースのリストを表示します。

```
aws resourcegroupstaggingapi get-resources \  
  --tag-filters Key=Environment,Values=Production \  
  --tags-per-page 100
```

出力:

```
{  
  "ResourceTagMappingList": [  
    {  
      "ResourceARN": " arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0",  
      "Tags": [  
        {  
          "Key": "Environment",  
          "Value": "Production"  
        }  
      ]  
    }  
  ]  
}
```

詳細については、[GetResources](#) リソースグループのタグ付けAPIリファレンスの「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResources](#)」の「」を参照してください。

get-tag-keys

次の例は、get-tag-keys を使用する方法を説明しています。

AWS CLI

すべてのタグキーのリストを取得するには

次のget-tag-keys例では、アカウントのリソースで使用されるすべてのタグキー名のリストを取得します。

```
aws resourcegroupstaggingapi get-tag-keys
```

出力:

```
{
  "TagKeys": [
    "Environment",
    "CostCenter",
    "Department"
  ]
}
```

詳細については、「リソースグループのタグ付けリファレンス[GetTagKeys](#)」の「」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTagKeys](#)」の「」を参照してください。

get-tag-values

次の例は、get-tag-values を使用する方法を説明しています。

AWS CLI

すべてのタグ値のリストを取得するには

次のget-tag-values例では、内のすべてのリソースの指定されたキーに使用されるすべての値を表示します。

```
aws resourcegroupstaggingapi get-tag-values \
  --key=Environment
```

出力:

```
{
  "TagValues": [
    "Alpha",
    "Gamma",
    "Production"
  ]
}
```

詳細については、リソースグループのタグ付けリファレンス[GetTagValues](#)の「」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTagValues](#)」の「」を参照してください。

tag-resources

次のコード例は、tag-resources を使用する方法を示しています。

AWS CLI

リソースにタグをアタッチするには

次のtag-resources例では、指定されたリソースにキー名と値をタグ付けします。

```
aws resourcegroupstaggingapi tag-resources \  
  --resource-arn-list arn:aws:s3:::MyProductionBucket \  
  --tags Environment=Production, CostCenter=1234
```

出力:

```
{  
  "FailedResourcesMap": {}  
}
```

詳細については、[TagResources](#)リソースグループのタグ付けAPIリファレンスの「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResources](#)」の「」を参照してください。

untag-resources

次の例は、untag-resources を使用する方法を説明しています。

AWS CLI

リソースからタグを削除するには

次のuntag-resources例では、指定されたリソースから指定されたタグキーと関連付けられた値を削除します。


```
aws resourcegroupstaggingapi untag-resources \  
  --resource-arn-list arn:aws:s3:::awsexamplebucket \  
  --tag-keys Environment CostCenter
```

出力:

```
{  
  "FailedResourcesMap": {}  
}
```

詳細については、「リソースグループのタグ付けリファレンス [UntagResources](#)」の「」を参照してください。 API

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResources](#)」の「」を参照してください。

AWS RoboMaker を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS RoboMaker。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-describe-simulation-job

次の例は、batch-describe-simulation-job を使用する方法を説明しています。

AWS CLI

シミュレーションジョブをバッチ記述するには

次のbatch-describe-simulation-job例では、指定された 3 つのシミュレーションジョブの詳細を取得します。

コマンド:

```
aws robomaker batch-describe-simulation-job \  
--job arn:aws:robomaker:us-west-2:111111111111:simulation-job/  
sim-66bbb3gpxm8x arn:aws:robomaker:us-west-2:111111111111:simulation-job/  
sim-p0cpdrrwng2n arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-  
g8h6tg1mblgw
```

出力:

```
{  
  "jobs": [  
    {  
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/  
sim-66bbb3gpxm8x",  
      "status": "Completed",  
      "lastUpdatedAt": 1548959178.0,  
      "failureBehavior": "Continue",  
      "clientRequestToken": "6020408e-b05c-4310-9f13-4ed71c5221ed",  
      "outputLocation": {  
        "s3Bucket": "awsrobomakerobjecttracker-1111111111-  
bundlesbucket-2lk584kiq1oa",  
        "s3Prefix": "output"  
      },  
      "maxJobDurationInSeconds": 3600,  
      "simulationTimeMillis": 0,  
      "iamRole": "arn:aws:iam::111111111111:role/  
AWSRoboMakerObjectTracker-154895-SimulationJobRole-14D5ASA7PQE3A",  
      "simulationApplications": [  
        {  
          "application": "arn:aws:robomaker:us-  
west-2:111111111111:simulation-application/  
AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",  
          "applicationVersion": "$LATEST",  
          "launchConfig": {  
            "packageName": "object_tracker_simulation",  
            "launchFile": "local_training.launch",  
            "environmentVariables": {  
              "MARKOV_PRESET_FILE": "object_tracker.py",  

```

```

        "MODEL_S3_BUCKET": "awsrobomakerobjecttracker-1111111111-
bundlesbucket-2lk584kiq1oa",
        "MODEL_S3_PREFIX": "model-store",
        "ROS_AWS_REGION": "us-west-2"
    }
}
],
"tags": {},
"vpcConfig": {
    "subnets": [
        "subnet-716dd52a",
        "subnet-43c22325",
        "subnet-3f526976"
    ],
    "securityGroups": [
        "sg-3fb40545"
    ],
    "vpcId": "vpc-99895eff",
    "assignPublicIp": true
}
},
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
p0cpdrrwng2n",
    "status": "Completed",
    "lastUpdatedAt": 1548168817.0,
    "failureBehavior": "Continue",
    "clientRequestToken": "e4a23e75-f9a7-411d-835f-21881c82c58b",
    "outputLocation": {
        "s3Bucket": "awsrobomakercloudwatch-111111111111-
bundlesbucket-14e5s9jvwtmv7",
        "s3Prefix": "output"
    },
    "maxJobDurationInSeconds": 3600,
    "simulationTimeMillis": 0,
    "iamRole": "arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G00BWTQ8YBG6",
    "robotApplications": [
        {
            "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerCloudWatch-1547663411642_NZbpqEJ3T/1547663517377",
            "applicationVersion": "$LATEST",
            "launchConfig": {

```

```
        "packageName": "cloudwatch_robot",
        "launchFile": "await_commands.launch",
        "environmentVariables": {
            "LAUNCH_ID": "1548168752173",
            "ROS_AWS_REGION": "us-west-2"
        }
    }
},
"simulationApplications": [
    {
        "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIIt6D1h6/1547663521470",
        "applicationVersion": "$LATEST",
        "launchConfig": {
            "packageName": "cloudwatch_simulation",
            "launchFile": "bookstore_turtlebot_navigation.launch",
            "environmentVariables": {
                "LAUNCH_ID": "1548168752173",
                "ROS_AWS_REGION": "us-west-2",
                "TURTLEBOT3_MODEL": "waffle_pi"
            }
        }
    }
],
"tags": {},
"vpcConfig": {
    "subnets": [
        "subnet-716dd52a",
        "subnet-43c22325",
        "subnet-3f526976"
    ],
    "securityGroups": [
        "sg-3fb40545"
    ],
    "vpcId": "vpc-99895eff",
    "assignPublicIp": true
}
},
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
g8h6tg1mblgw",
    "status": "Canceled",
```

```

    "lastUpdatedAt": 1546543442.0,
    "failureBehavior": "Fail",
    "clientRequestToken": "d796bbb4-2a2c-1abc-f2a9-0d9e547d853f",
    "outputLocation": {
      "s3Bucket": "sample-bucket",
      "s3Prefix": "SimulationLog_115490482698"
    },
    "maxJobDurationInSeconds": 28800,
    "simulationTimeMillis": 0,
    "iamRole": "arn:aws:iam::111111111111:role/RoboMakerSampleTheFirst",
    "robotApplications": [
      {
        "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerHelloWorldRobot/1546541208251",
        "applicationVersion": "$LATEST",
        "launchConfig": {
          "packageName": "hello_world_robot",
          "launchFile": "rotate.launch"
        }
      }
    ],
    "simulationApplications": [
      {
        "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
RoboMakerHelloWorldSimulation/1546541198985",
        "applicationVersion": "$LATEST",
        "launchConfig": {
          "packageName": "hello_world_simulation",
          "launchFile": "empty_world.launch"
        }
      }
    ],
    "tags": {}
  }
],
  "unprocessedJobs": []
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDescribeSimulationJob](#)」の「」を参照してください。

cancel-simulation-job

次の例は、cancel-simulation-job を使用する方法を説明しています。

AWS CLI

シミュレーションジョブをキャンセルするには

次のcancel-simulation-job例では、指定されたシミュレーションジョブをキャンセルします。

```
aws robomaker cancel-simulation-job \  
  --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelSimulationJob](#)」の「」を参照してください。

create-deployment-job

次のコード例は、create-deployment-job を使用する方法を示しています。

AWS CLI

デプロイジョブを作成するには

この例では、フリートのデプロイジョブを作成します MyFleet。これには、ENVIRONMENT 「」という名前の環境変数が含まれます。また、「リージョン」というタグもアタッチします。

コマンド:

```
aws robomaker create-deployment-job --deployment-  
config concurrentDeploymentPercentage=20, failureThresholdPercentage=25  
  --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/  
Trek/1539894765711 --tags Region=West --deployment-application-  
configs application=arn:aws:robomaker:us-west-2:111111111111:robot-application/  
RoboMakerVoiceInteractionRobot/1546537110575, applicationVersion=1, launchConfig={environmentV
```

出力:

```
{  
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/sim-0974h36s4v0t",
```

```
"fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1539894765711",
"status": "Pending",
"deploymentApplicationConfigs": [
  {
    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-application/RoboMakerVoiceInteractionRobot/1546537110575",
    "applicationVersion": "1",
    "launchConfig": {
      "packageName": "voice_interaction_robot",
      "launchFile": "await_commands.launch",
      "environmentVariables": {
        "ENVIRONMENT": "Beta"
      }
    }
  }
],
"createdAt": 1550770236.0,
"deploymentConfig": {
  "concurrentDeploymentPercentage": 20,
  "failureThresholdPercentage": 25
},
"tags": {
  "Region": "West"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDeploymentJob](#)」の「」を参照してください。

create-fleet

次のコード例は、create-fleet を使用する方法を示しています。

AWS CLI

フリートを作成するには

この例ではフリートを作成します。リージョンというタグがアタッチされます。

コマンド:

```
aws robomaker create-fleet --name MyFleet --tags Region=East
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyOtherFleet/1550771394395",
  "name": "MyFleet",
  "createdAt": 1550771394.0,
  "tags": {
    "Region": "East"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFleet](#)」の「」を参照してください。

create-robot-application-version

次のコード例は、create-robot-application-version を使用する方法を示しています。

AWS CLI

ロボットアプリケーションバージョンを作成するには

この例では、ロボットアプリケーションバージョンを作成します。

コマンド:

```
aws robomaker create-robot-application-version --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551201873931
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551201873931",
  "name": "MyRobotApplication",
  "version": "1",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-robot-application.tar.gz",
      "etag": "f8cf5526f1c6e7b3a72c3ed3f79c5493-70",
    }
  ]
}
```



```
        "architecture": "ARMHF"
      }
    ],
    "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
    },
    "lastUpdatedAt": 1551201873.0,
    "revisionId": "9986bb8d-a695-4ab4-8810-9f4a74d1aa00"
  "tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateRobotApplicationVersion](#)」の「」を参照してください。

create-robot-application

次のコード例は、create-robot-application を使用する方法を示しています。

AWS CLI

ロボットアプリケーションを作成するには

この例では、ロボットアプリケーションを作成します。

コマンド:

```
aws robomaker create-robot-application --name MyRobotApplication --
sources s3Bucket=my-bucket,s3Key=my-robot-application.tar.gz,architecture=X86_64 --
robot-software-suite name=ROS,version=Kinetic
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551201873931",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-robot-application.tar.gz",
```

```
        "architecture": "ARMHF"
      }
    ],
    "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
    },
    "lastUpdatedAt": 1551201873.0,
    "revisionId": "1f3cb539-9239-4841-a656-d3efcffa07e1",
    "tags": {}
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateRobotApplication](#)」の「」を参照してください。

create-robot

次のコード例は、create-robot を使用する方法を示しています。

AWS CLI

ロボットを作成するには

この例ではロボットを作成します。ARMHF アーキテクチャを使用します。また、リージョンというタグもアタッチします。

コマンド:

```
aws robomaker create-robot --name MyRobot --architecture ARMHF --greengrass-group-id 0f728a3c-7dbf-4a3e-976d-d16a8360caba --tags Region=East
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
  "name": "MyRobot",
  "createdAt": 1550772325.0,
  "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
  "architecture": "ARMHF",
  "tags": {
    "Region": "East"
  }
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRobot](#)」の「」を参照してください。

create-simulation-application-version

次のコード例は、create-simulation-application-version を使用する方法を示しています。

AWS CLI

シミュレーションアプリケーションバージョンを作成するには

この例では、ロボットアプリケーションバージョンを作成します。

コマンド:

```
aws robomaker create-simulation-application-version --  
application arn:aws:robomaker:us-west-2:111111111111:robot-application/  
MySimulationApplication/1551203427605
```

出力:

```
{  
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/  
MyRobotApplication/1551203427605",  
  "name": "MyRobotApplication",  
  "version": "1",  
  "sources": [  
    {  
      "s3Bucket": "my-bucket",  
      "s3Key": "my-simulation-application.tar.gz",  
      "etag": "00d8a94ff113856688c4fce618ae0f45-94",  
      "architecture": "X86_64"  
    }  
  ],  
  "simulationSoftwareSuite": {  
    "name": "Gazebo",  
    "version": "7"  
  },  
  "robotSoftwareSuite": {
```

```
    "name": "ROS",
    "version": "Kinetic"
  },
  "renderingEngine": {
    "name": "OGRE",
    "version": "1.x"
  },
  "lastUpdatedAt": 1551203853.0,
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
  "tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSimulationApplicationVersion](#)」の「」を参照してください。

create-simulation-application

次の例は、create-simulation-application を使用方法を説明しています。

AWS CLI

シミュレーションアプリケーションを作成するには

この例では、シミュレーションアプリケーションを作成します。

コマンド:

```
aws robomaker create-simulation-application --name MyRobotApplication --
sources s3Bucket=my-bucket,s3Key=my-simulation-application.tar.gz,architecture=ARMHF
--robot-software-suite name=ROS,version=Kinetic --simulation-software-
suite name=Gazebo,version=7 --rendering-engine name=OGRE,version=1.x
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
MyRobotApplication/1551203301792",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-simulation-application.tar.gz",
```

```
        "architecture": "X86_64"
      }
    ],
    "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
    },
    "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
    },
    "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
    },
    "lastUpdatedAt": 1551203301.0,
    "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
    "tags": {}
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSimulationApplication](#)」の「」を参照してください。

create-simulation-job

次の例は、create-simulation-job を使用する方法を説明しています。

AWS CLI

シミュレーションジョブを作成するには

この例では、シミュレーションジョブを作成します。ロボットアプリケーションとシミュレーションアプリケーションを使用します。

コマンド:

```
aws robomaker create-simulation-job --max-job-duration-
in-seconds 3600 --iam-role arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G00BWTQ8YBG6 --robot-
applications application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821,launchConfig={packageName=hello_world_robot,launchFile=rota
--simulation-applications application=arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
```

```
MySimulationApplication/1551203427605, launchConfig={packageName=hello_world_simulation, launchFile=empty_world.launch}, --tags Region=North
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-w7m68wpr05h8",
  "status": "Pending",
  "lastUpdatedAt": 1551213837.0,
  "failureBehavior": "Fail",
  "clientRequestToken": "b283ccce-e468-43ee-8642-be76a9d69f15",
  "maxJobDurationInSeconds": 3600,
  "simulationTimeMillis": 0,
  "iamRole": "arn:aws:iam::111111111111:role/MySimulationRole",
  "robotApplications": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821",
      "applicationVersion": "$LATEST",
      "launchConfig": {
        "packageName": "hello_world_robot",
        "launchFile": "rotate.launch"
      }
    }
  ],
  "simulationApplications": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605",
      "applicationVersion": "$LATEST",
      "launchConfig": {
        "packageName": "hello_world_simulation",
        "launchFile": "empty_world.launch"
      }
    }
  ],
  "tags": {
    "Region": "North"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSimulationJob](#)」の「」を参照してください。

delete-fleet

次のコード例は、delete-fleet を使用する方法を示しています。

AWS CLI

フリートを削除するには

この例では、フリートを削除します。

コマンド:

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFleet](#)」の「」を参照してください。

delete-robot-application

次のコード例は、delete-robot-application を使用する方法を示しています。

AWS CLI

ロボットアプリケーションを削除するには

この例では、ロボットアプリケーションを削除します。

コマンド:

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRobotApplication](#)」の「」を参照してください。

delete-robot

次の例は、delete-robot を使用する方法を説明しています。

AWS CLI

ロボットを削除するには

この例では、ロボットを削除します。

コマンド:

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteRobot](#)」の「」を参照してください。

delete-simulation-application

次の例は、delete-simulation-application を使用する方法を説明しています。

AWS CLI

シミュレーションアプリケーションを削除するには

この例では、シミュレーションアプリケーションを削除します。

コマンド:

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSimulationApplication](#)」の「」を参照してください。

deregister-robot

次の例は、deregister-robot を使用する方法を説明しています。

AWS CLI

フリートからロボットの登録を解除するには

この例では、ロボットをフリートから登録解除します。

コマンド:


```
aws robomaker deregister-robot --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

出力:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterRobot](#)」の「」を参照してください。

describe-deployment-job

次の例は、describe-deployment-job を使用する方法を説明しています。

AWS CLI

デプロイジョブを記述するには

次のdescribe-deployment-job例では、指定されたデプロイジョブの詳細を取得します。

```
aws robomaker describe-deployment-job \
  --job arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-xl8qssl6pbcn
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-xl8qssl6pbcn",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/Trek/1539894765711",
  "status": "InProgress",
  "deploymentConfig": {
    "concurrentDeploymentPercentage": 20,
    "failureThresholdPercentage": 25
  },
}
```

```
"deploymentApplicationConfigs": [
  {
    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerHelloWorldRobot/1546541208251",
    "applicationVersion": "1",
    "launchConfig": {
      "packageName": "hello_world_robot",
      "launchFile": "rotate.launch"
    }
  }
],
"createdAt": 1551218369.0,
"robotDeploymentSummary": [
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540834232469",
    "deploymentStartTime": 1551218376.0,
    "status": "Deploying",
    "progressDetail": {}
  }
],
"tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDeploymentJob](#)」の「」を参照してください。

describe-fleet

次の例は、describe-fleet を使用する方法を説明しています。

AWS CLI

フリートを記述するには

次のdescribe-fleet例では、指定されたフリートの詳細を取得します。

```
aws robomaker describe-fleet \
  --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907
```

出力:

```
{
  "name": "MyFleet",
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1539894765711",
  "robots": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540834232469",
      "createdAt": 1540834232.0
    },
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyOtherRobot/1540829698778",
      "createdAt": 1540829698.0
    }
  ],
  "createdAt": 1539894765.0,
  "lastDeploymentStatus": "Succeeded",
  "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-xl8qssl6pbcn",
  "lastDeploymentTime": 1551218369.0,
  "tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeFleet](#)」の「」を参照してください。

describe-robot-application

次の例は、describe-robot-application を使用する方法を説明しています。

AWS CLI

ロボットアプリケーションを記述するには

この例では、ロボットアプリケーションについて説明します。

コマンド:

```
aws robomaker describe-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-robot-application.tar.gz",
      "architecture": "X86_64"
    }
  ],
  "robotSoftwareSuite": {
    "name": "ROS",
    "version": "Kinetic"
  },
  "revisionId": "e72efe0d-f44f-4333-b604-f6fa5c6bb50b",
  "lastUpdatedAt": 1551203485.0,
  "tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRobotApplication](#)」の「」を参照してください。

describe-robot

次の例は、describe-robot を使用する方法を説明しています。

AWS CLI

ロボットを記述するには

この例では、ロボットについて説明します。

コマンド:

```
aws robomaker describe-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1550772324398
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398",
  "name": "MyRobot",
  "status": "Available",
  "greengrassGroupId": "0f728a3c-7dbf-4a3e-976d-d16a8360caba",
  "createdAt": 1550772325.0,
  "architecture": "ARMHF",
  "tags": {
    "Region": "East"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeRobot](#)」の「」を参照してください。

describe-simulation-application

次のコード例は、describe-simulation-application を使用する方法を示しています。

AWS CLI

シミュレーションアプリケーションを記述するには

この例では、シミュレーションアプリケーションについて説明します。

コマンド:

```
aws robomaker describe-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
  "version": "$LATEST",
  "sources": [
    {
      "s3Bucket": "my-bucket",
      "s3Key": "my-simulation-application.tar.gz",
    }
  ]
}
```

```
        "architecture": "X86_64"
      }
    ],
    "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
    },
    "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
    },
    "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
    },
    "revisionId": "783674ab-b7b8-42d9-b01f-9373907987e5",
    "lastUpdatedAt": 1551203427.0,
    "tags": {}
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSimulationApplication](#)」の「」を参照してください。

describe-simulation-job

次のコード例は、describe-simulation-job を使用する方法を示しています。

AWS CLI

シミュレーションジョブを記述するには

この例では、シミュレーションジョブについて説明します。

コマンド:

```
aws robomaker describe-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6
```

出力:

```
{
```

```
"arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6",
"status": "Running",
"lastUpdatedAt": 1551219349.0,
"failureBehavior": "Continue",
"clientRequestToken": "a19ec4b5-e50d-3591-33da-c2e593c60615",
"outputLocation": {
  "s3Bucket": "my-output-bucket",
  "s3Prefix": "output"
},
"maxJobDurationInSeconds": 3600,
"simulationTimeMillis": 0,
"iamRole": "arn:aws:iam::111111111111:role/MySimulationRole",
"robotApplications": [
  {
    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1551206341136",
    "applicationVersion": "$LATEST",
    "launchConfig": {
      "packageName": "hello_world_robot",
      "launchFile": "rotate.launch"
    }
  }
],
"simulationApplications": [
  {
    "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-
application/MySimulationApplication/1551206347967",
    "applicationVersion": "$LATEST",
    "launchConfig": {
      "packageName": "hello_world_simulation",
      "launchFile": "empty_world.launch"
    }
  }
],
"tags": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSimulationJob](#)」の「」を参照してください。

list-deployment-jobs

次の例は、list-deployment-jobs を使用する方法を説明しています。

AWS CLI

デプロイジョブを一覧表示するには

次のlist-deployment-jobs例では、デプロイジョブのリストを取得します。

```
aws robomaker list-deployment-jobs
```

出力:

```
{
  "deploymentJobs": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/sim-6293szzm56rv",
      "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1539894765711",
      "status": "InProgress",
      "deploymentApplicationConfigs": [
        {
          "application": "arn:aws:robomaker:us-west-2:111111111111:robot-application/HelloWorldRobot/1546537110575",
          "applicationVersion": "1",
          "launchConfig": {
            "packageName": "hello_world_robot",
            "launchFile": "rotate.launch",
            "environmentVariables": {
              "ENVIRONMENT": "Desert"
            }
          }
        }
      ],
      "deploymentConfig": {
        "concurrentDeploymentPercentage": 20,
        "failureThresholdPercentage": 25
      },
      "createdAt": 1550689373.0
    },
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-4w4g69p25zdb",
      "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1539894765711",

```



```
    "status": "Pending",
    "deploymentApplicationConfigs": [
      {
        "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerHelloWorld-1544562726923_YGHM_sh5M/1544562822877",
        "applicationVersion": "1",
        "launchConfig": {
          "packageName": "fail",
          "launchFile": "fail"
        }
      }
    ],
    "deploymentConfig": {
      "concurrentDeploymentPercentage": 20,
      "failureThresholdPercentage": 25
    },
    "failureReason": "",
    "failureCode": "",
    "createdAt": 1544719763.0
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeploymentJobs](#)」の「」を参照してください。

list-fleets

次のコード例は、list-fleets を使用する方法を示しています。

AWS CLI

フリートを一覧表示するには

この例では、フリートを一覧表示します。最大 20 個のフリートが返されます。

コマンド:

```
aws robomaker list-fleets --max-items 20
```

出力:

```
{
```

```
"fleetDetails": [  
  {  
    "name": "Trek",  
    "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/  
MyFleet/1539894765711",  
    "createdAt": 1539894765.0,  
    "lastDeploymentStatus": "Failed",  
    "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-  
job/deployment-4w4g69p25zdb",  
    "lastDeploymentTime": 1544719763.0  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFleets](#)」の「」を参照してください。

list-robot-applications

次のコード例は、list-robot-applications を使用する方法を示しています。

AWS CLI

ロボットアプリケーションを一覧表示するには

この例では、ロボットアプリケーションを一覧表示します。結果は 20 個のロボットアプリケーションに制限されます。

コマンド:

```
aws robomaker list-robot-applications --max-results 20
```

出力:

```
{  
  "robotApplicationSummaries": [  
    {  
      "name": "MyRobot",  
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/  
MyRobot/1546537110575",  
      "version": "$LATEST",  
      "lastUpdatedAt": 1546540372.0  
    }  
  ]  
}
```

```
    },
    {
      "name": "AnotherRobot",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
AnotherRobot/1546541208251",
      "version": "$LATEST",
      "lastUpdatedAt": 1546541208.0
    },
    {
      "name": "MySuperRobot",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/
MySuperRobot/1547663517377",
      "version": "$LATEST",
      "lastUpdatedAt": 1547663517.0
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRobotApplications](#)」の「」を参照してください。

list-robots

次のコード例は、list-robots を使用する方法を示しています。

AWS CLI

ロボットを一覧表示するには

この例では、ロボットを一覧表示します。最大 20 個のロボットが返されます。

コマンド:

```
aws robomaker list-robots --max-results 20
```

出力:

```
{
  "robots": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot100/1544035373264",
```

```
    "name": "Robot100",
    "status": "Available",
    "createdAt": 1544035373.0,
    "architecture": "X86_64"
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/Robot101/1542146976587",
    "name": "Robot101",
    "status": "Available",
    "createdAt": 1542146976.0,
    "architecture": "X86_64"
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/Robot102/1540834232469",
    "name": "Robot102",
    "fleetArn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/Trek/1539894765711",
    "status": "Available",
    "createdAt": 1540834232.0,
    "architecture": "X86_64",
    "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-jb007b75gl5f",
    "lastDeploymentTime": 1550689533.0
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778",
    "name": "MyRobot",
    "status": "Registered",
    "createdAt": 1540829698.0,
    "architecture": "X86_64"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRobots](#)」の「」を参照してください。

list-simulation-applications

次の例は、list-simulation-applications を使用する方法を説明しています。

AWS CLI

シミュレーションアプリケーションを一覧表示するには

この例では、シミュレーションアプリケーションを一覧表示します。最大 20 個のシミュレーションアプリケーションが返されます。

コマンド:

```
aws robomaker list-simulation-applications --max-results 20
```

出力:

```
{
  "simulationApplicationSummaries": [
    {
      "name": "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
      "version": "$LATEST",
      "lastUpdatedAt": 1548959170.0
    },
    {
      "name": "RoboMakerHelloWorldSimulation",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/RoboMakerHelloWorldSimulation/1546541198985",
      "version": "$LATEST",
      "lastUpdatedAt": 1546541198.0
    },
    {
      "name": "RoboMakerObjectTrackerSimulation",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/RoboMakerObjectTrackerSimulation/1545846795615",
      "version": "$LATEST",
      "lastUpdatedAt": 1545847405.0
    },
    {
      "name": "RoboMakerVoiceInteractionSimulation",
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/RoboMakerVoiceInteractionSimulation/1546537100507",
      "version": "$LATEST",
      "lastUpdatedAt": 1546540352.0
    }
  ]
}
```

```
{
  "name": "AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6",
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6/1547663521470",
  "version": "$LATEST",
  "lastUpdatedAt": 1547663521.0
},
{
  "name": "AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-",
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerDeepRacer-1545848257672_1YZCaieQ-/1545848370525",
  "version": "$LATEST",
  "lastUpdatedAt": 1545848370.0
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSimulationApplications](#)」の「」を参照してください。

list-simulation-jobs

次の例は、list-simulation-jobs を使用する方法を説明しています。

AWS CLI

シミュレーションジョブを一覧表示するには

この例では、シミュレーションジョブを一覧表示します。

コマンド:

```
aws robomaker list-simulation-jobs
```

出力:

```
{
  "simulationJobSummaries": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-66bbb3gpxm8x",
      "lastUpdatedAt": 1548959178.0,
      "status": "Completed",
```

```
    "simulationApplicationNames": [
      "AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq"
    ],
    "robotApplicationNames": [
      null
    ]
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
b27c4rkrtzcv",
    "lastUpdatedAt": 1543514088.0,
    "status": "Canceled",
    "simulationApplicationNames": [
      "AWSRoboMakerPersonDetection-1543513948280_T8rHW2_lu"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerPersonDetection-1543513948280_EYaMT0mYb"
    ]
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-51vxjbzy4q8t",
    "lastUpdatedAt": 1543508858.0,
    "status": "Canceled",
    "simulationApplicationNames": [
      "AWSRoboMakerCloudWatch-1543504747391_1FF9ZQyx6"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
    ]
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
kgf1fqxflqbx",
    "lastUpdatedAt": 1543504862.0,
    "status": "Completed",
    "simulationApplicationNames": [
      "AWSRoboMakerCloudWatch-1543504747391_1FF9ZQyx6"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
    ]
  },
  {
```

```
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
vw8lvh061nqt",
    "lastUpdatedAt": 1543441430.0,
    "status": "Completed",
    "simulationApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341__yb_Jg961"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
    ]
  },
  {
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
txy5ypxmh84",
    "lastUpdatedAt": 1543437488.0,
    "status": "Completed",
    "simulationApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341__yb_Jg961"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSimulationJobs](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

この例では、AWS RoboMaker リソースのタグを一覧表示します。

コマンド:

```
aws robomaker list-tags-for-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/Robby_the_Robot/1544035373264"
```


出力:

```
{
  "tags": {
    "Region": "North",
    "Stage": "Initial"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

register-robot

次のコード例は、register-robot を使用する方法を示しています。

AWS CLI

ロボットを登録するには

この例では、ロボットをフリートに登録します。

コマンド:

```
aws robomaker register-robot --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907 --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

出力:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterRobot](#)」の「」を参照してください。

restart-simulation-job

次のコード例は、restart-simulation-job を使用する方法を示しています。

AWS CLI

シミュレーションを再起動するには

この例では、シミュレーションを再起動します。

コマンド:

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RestartSimulationJob](#)」の「」を参照してください。

sync-deployment-job

次のコード例は、sync-deployment-job を使用する方法を示しています。

AWS CLI

デプロイジョブを同期するには

この例では、デプロイジョブを同期します。

コマンド:

```
aws robomaker sync-deployment-job --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/Trek/1539894765711
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-job/
deployment-09ccxs3tlfms",
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1539894765711",
  "status": "Pending",
  "deploymentConfig": {
    "concurrentDeploymentPercentage": 20,
```

```
    "failureThresholdPercentage": 25
  },
  "deploymentApplicationConfigs": [
    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/MyRobotApplication/1546541208251",
      "applicationVersion": "1",
      "launchConfig": {
        "packageName": "hello_world_simulation",
        "launchFile": "empty_world.launch"
      }
    }
  ],
  "createdAt": 1551286954.0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SyncDeploymentJob](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを付けるには

この例では、リソースにタグを付けます。リージョンとステージの 2 つのタグをアタッチします。

コマンド:

```
aws robomaker tag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tags Region=North,Stage=Initial
```

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースのタグを解除するには

この例では、リソースからタグを削除します。リージョンタグを削除します。

コマンド:

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

update-robot-application

次のコード例は、update-robot-application を使用する方法を示しています。

AWS CLI

ロボットアプリケーションを更新するには

この例では、ロボットアプリケーションを更新します。

コマンド:

```
aws robomaker update-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821 --sources s3Bucket=my-bucket,s3Key=my-robot-application.tar.gz,architecture=X86_64 --robot-software-suite name=ROS,version=Kinetic
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821",
  "name": "MyRobotApplication",
  "version": "$LATEST",
  "sources": [
    {
```

```
        "s3Bucket": "my-bucket",
        "s3Key": "my-robot-application.tar.gz",
        "architecture": "X86_64"
    }
],
"robotSoftwareSuite": {
    "name": "ROS",
    "version": "Kinetic"
},
"lastUpdatedAt": 1551287993.0,
"revisionId": "20b5e331-24fd-4504-8b8c-531afe5f4c94"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateRobotApplication](#)」の「」を参照してください。

update-simulation-application

次の例は、update-simulation-application を使用する方法を説明しています。

AWS CLI

シミュレーションアプリケーションを更新するには

この例では、シミュレーションアプリケーションを更新します。

コマンド:

```
aws robomaker update-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605 --sources s3Bucket=my-bucket,s3Key=my-simulation-application.tar.gz,architecture=X86_64 --robot-software-suite name=ROS,version=Kinetic --simulation-software-suite name=Gazebo,version=7 --rendering-engine name=OGRE,version=1.x
```

出力:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605",
  "name": "MySimulationApplication",
}
```

```
"version": "$LATEST",
"sources": [
  {
    "s3Bucket": "my-bucket",
    "s3Key": "my-simulation-application.tar.gz",
    "architecture": "X86_64"
  }
],
"simulationSoftwareSuite": {
  "name": "Gazebo",
  "version": "7"
},
"robotSoftwareSuite": {
  "name": "ROS",
  "version": "Kinetic"
},
"renderingEngine": {
  "name": "OGRE",
  "version": "1.x"
},
"lastUpdatedAt": 1551289361.0,
"revisionId": "4a22cb5d-93c5-4cef-9311-52bdd119b79e"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSimulationApplication](#)」の「」を参照してください。

を使用した Route 53 の例 AWS CLI

次のコード例は、Route 53 AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

change-resource-record-sets

次の例は、change-resource-record-sets を使用する方法を説明しています。

AWS CLI

リソースレコードセットを作成、更新、または削除するには

次のchange-resource-record-setsコマンドは、ファイルの hosted-zone-idZ1R8UBAEXAMPLEおよび JSON形式の設定を使用してリソースレコードセットを作成しますC:\awscli\route53\change-resource-record-sets.json。

```
aws route53 change-resource-record-sets --hosted-zone-id Z1R8UBAEXAMPLE --change-batch file://C:\awscli\route53\change-resource-record-sets.json
```

詳細については、Amazon Route 53APIリファレンスPOST ChangeResourceRecordSets」の「」を参照してください。

JSON ファイルの設定は、作成するリソースレコードセットの種類によって異なります。

BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover エイリアス

基本構文：

```
{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ]
      }
    },
    ...
  ],
}
```

```

    {...}
  ]
}

```

加重構文 :

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Weight": value between 0 and 255,
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ],
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}

```

エイリアス構文 :

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",

```



```

    "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
bucket, Elastic Load Balancing load balancer, or another resource record set in
this hosted zone",
    "EvaluateTargetHealth": true|false
  },
  "HealthCheckId": "optional ID of an Amazon Route 53 health check"
}
},
{...}
]
}

```

加重エイリアス構文 :

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Weight": value between 0 and 255,
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
bucket, Elastic Load Balancing load balancer, or another resource record set in
this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}

```

レイテンシー構文 :

```

{
  "Comment": "optional comment about the changes in this change batch request",

```

```

"Changes": [
  {
    "Action": "CREATE"|"DELETE"|"UPSERT",
    "ResourceRecordSet": {
      "Name": "DNS domain name",
      "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
      "SetIdentifier": "unique description for this resource record set",
      "Region": "Amazon EC2 region name",
      "TTL": time to live in seconds,
      "ResourceRecords": [
        {
          "Value": "applicable value for the record type"
        },
        {...}
      ],
      "HealthCheckId": "optional ID of an Amazon Route 53 health check"
    }
  },
  {...}
]
}

```

レイテンシーエイリアス構文:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Region": "Amazon EC2 region name",
        "AliasTarget": {
          "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
          "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
bucket, Elastic Load Balancing load balancer, or another resource record set in
this hosted zone",
          "EvaluateTargetHealth": true|false
        },
        "HealthCheckId": "optional ID of an Amazon Route 53 health check"
      }
    }
  ]
}

```

```

    }
  },
  {...}
]
}

```

フェイルオーバー構文 :

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "Failover": "PRIMARY" | "SECONDARY",
        "TTL": time to live in seconds,
        "ResourceRecords": [
          {
            "Value": "applicable value for the record type"
          },
          {...}
        ],
        "HealthCheckId": "ID of an Amazon Route 53 health check"
      }
    },
    {...}
  ]
}

```

フェイルオーバーエイリアス構文 :

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",

```

```

    "Failover": "PRIMARY" | "SECONDARY",
    "AliasTarget": {
      "HostedZoneId": "hosted zone ID for your CloudFront distribution, Amazon
S3 bucket, Elastic Load Balancing load balancer, or Amazon Route 53 hosted zone",
      "DNSName": "DNS domain name for your CloudFront distribution, Amazon S3
bucket, Elastic Load Balancing load balancer, or another resource record set in
this hosted zone",
      "EvaluateTargetHealth": true|false
    },
    "HealthCheckId": "optional ID of an Amazon Route 53 health check"
  }
},
{...}
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ChangeResourceRecordSets](#)」の「」を参照してください。

change-tags-for-resource

次の例は、change-tags-for-resource を使用する方法を説明しています。

AWS CLI

次のコマンドは、ID で指定されたヘルスチェックリソースownerに というタグを追加します。

```
aws route53 change-tags-for-resource --resource-type healthcheck --resource-id 6233434j-18c1-34433-ba8e-3443434 --add-tags Key=owner, Value=myboss
```

次のコマンドは、ID で指定されたホストゾーンリソースownerから というタグを削除します。

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-id Z1523434445 --remove-tag-keys owner
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ChangeTagsForResource](#)」の「」を参照してください。

create-health-check

次の例は、create-health-check を使用する方法を説明しています。

AWS CLI

ヘルスチェックを作成するには

次のcreate-health-checkコマンドは、発信者リファレンス2014-04-01-18:47とJSON形式の設定を使用して、ファイルでヘルスチェックを作成しますC:\awscli\route53\create-health-check.json。

```
aws route53 create-health-check --caller-reference 2014-04-01-18:47 --health-check-config file://C:\awscli\route53\create-health-check.json
```

JSON 構文 :

```
{
  "IPAddress": "IP address of the endpoint to check",
  "Port": port on the endpoint to check--required when Type is "TCP",
  "Type": "HTTP"|"HTTPS"|"HTTP_STR_MATCH"|"HTTPS_STR_MATCH"|"TCP",
  "ResourcePath": "path of the file that you want Amazon Route 53 to request--all Types except TCP",
  "FullyQualifiedDomainName": "domain name of the endpoint to check--all Types except TCP",
  "SearchString": "if Type is HTTP_STR_MATCH or HTTPS_STR_MATCH, the string to search for in the response body from the specified resource",
  "RequestInterval": 10 | 30,
  "FailureThreshold": integer between 1 and 10
}
```

Route 53 リソースレコードセットにヘルスチェックを追加するには、change-resource-record-sets コマンドを使用します。

詳細については、「Amazon Route 53 DNS デベロッパーガイド」のAmazon Route 53ヘルスチェックとフェイルオーバー」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateHealthCheck](#)」の「」を参照してください。

create-hosted-zone

次の例は、create-hosted-zone を使用する方法を説明しています。

AWS CLI

ホストゾーンを作成するには

次のcreate-hosted-zoneコマンドは、発信者リファレンス `example.com` を使用して という名前のホストゾーンを追加します2014-04-01-18:47。オプションのコメントにはスペースが含まれているため、引用符で囲む必要があります。

```
aws route53 create-hosted-zone --name example.com --caller-reference 2014-04-01-18:47 --hosted-zone-config Comment="command-line version"
```

詳細については、「Amazon Route 53デベロッパガイド」の「ホストゾーンの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateHostedZone](#)」の「」を参照してください。

delete-health-check

次のコード例は、delete-health-check を使用する方法を示しています。

AWS CLI

ヘルスチェックを削除するには

次のdelete-health-checkコマンドは、health-check-id の を使用してヘルスチェックを削除しますe75b48d9-547a-4c3d-88a5-ae4002397608。

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-ae4002397608
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteHealthCheck](#)」の「」を参照してください。

delete-hosted-zone

次の例は、delete-hosted-zone を使用する方法を説明しています。

AWS CLI

ホストゾーンを削除するには

次のdelete-hosted-zoneコマンドは、idの を持つホストゾーンを削除しますZ36KTIQEXAMPLE。

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[DeleteHostedZone](#)」の「」を参照してください。

get-change

次のコード例は、get-change を使用する方法を示しています。

AWS CLI

リソースレコードセットの変更のステータスを取得するには

次のget-changeコマンドは、Idの を持つchange-resource-record-setsリクエストのステータスとその他の情報を取得します/change/CWPIK4URU2I5S。

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[GetChange](#)」の「」を参照してください。

get-health-check

次のコード例は、get-health-check を使用する方法を示しています。

AWS CLI

ヘルスチェックに関する情報を取得するには

次のget-health-checkコマンドは、health-check-id の を持つヘルスチェックに関する情報を取得します02ec8401-9879-4259-91fa-04e66d094674。

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[GetHealthCheck](#)」の「」を参照してください。

get-hosted-zone

次の例は、get-hosted-zone を使用する方法を説明しています。

AWS CLI

ホストゾーンに関する情報を取得するには

次のget-hosted-zoneコマンドは、idの を使用してホストゾーンに関する情報を取得しますZ1R8UBAEXAMPLE。

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetHostedZone](#)」の「」を参照してください。

list-health-checks

次の例は、list-health-checks を使用する方法を説明しています。

AWS CLI

現在の AWS アカウントに関連付けられているヘルスチェックを一覧表示するには

次のlist-health-checksコマンドは、現在の AWS アカウントに関連付けられている最初の 100 件のヘルスチェックに関する詳細情報を一覧表示します。

```
aws route53 list-health-checks
```

100 を超えるヘルスチェックがある場合、または 100 未満のグループでリストする場合は、--maxitemsパラメータを含めます。例えば、ヘルスチェックを一度に 1 つずつ一覧表示するには、次のコマンドを使用します。

```
aws route53 list-health-checks --max-items 1
```

次のヘルスチェックを表示するには、前のコマンドに対するレスポンスNextTokenからの値を取得し、--starting-tokenパラメータに含めます。例：

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```


- API 詳細については、AWS CLI 「コマンドリファレンス[ListHealthChecks](#)」の「」を参照してください。

list-hosted-zones-by-name

次の例は、list-hosted-zones-by-name を使用する方法を説明しています。

AWS CLI

次のコマンドは、ドメイン名順に最大 100 のホストゾーンを一覧表示します。

```
aws route53 list-hosted-zones-by-name
```

出力:

```
{
  "HostedZones": [
    {
      "ResourceRecordSetCount": 2,
      "CallerReference": "test20150527-2",
      "Config": {
        "Comment": "test2",
        "PrivateZone": false
      },
      "Id": "/hostedzone/Z119WBBTVP5WFX",
      "Name": "2.example.com."
    },
    {
      "ResourceRecordSetCount": 2,
      "CallerReference": "test20150527-1",
      "Config": {
        "Comment": "test",
        "PrivateZone": false
      },
      "Id": "/hostedzone/Z3P5QSUBK4POTI",
      "Name": "www.example.com."
    }
  ],
  "IsTruncated": false,
  "MaxItems": "100"
}
```

次のコマンドは、で始まる名前順にホストゾーンを一覧表示します `www.example.com`。

```
aws route53 list-hosted-zones-by-name --dns-name www.example.com
```

出力:

```
{
  "HostedZones": [
    {
      "ResourceRecordSetCount": 2,
      "CallerReference": "mwunderl20150527-1",
      "Config": {
        "Comment": "test",
        "PrivateZone": false
      },
      "Id": "/hostedzone/Z3P5QSUBK4P0TI",
      "Name": "www.example.com."
    }
  ],
  "DNSName": "www.example.com",
  "IsTruncated": false,
  "MaxItems": "100"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListHostedZonesByName](#)」の「」を参照してください。

list-hosted-zones

次の例は、`list-hosted-zones` を使用する方法を説明しています。

AWS CLI

現在の AWS アカウントに関連付けられているホストゾーンを一覧表示するには

次の `list-hosted-zones` コマンドは、現在の AWS アカウントに関連付けられている最初の 100 のホストゾーンの概要情報を一覧表示します。

```
aws route53 list-hosted-zones
```

ホストゾーンが 100 個を超える場合や、100 個未満のグループにホストゾーンを一覧表示する場合は、`--max-items` パラメータを含めてください。例えば、タイムゾーンを一度に一覧表示するには、次のコマンドを使用します。

```
aws route53 list-hosted-zones --max-items 1
```

次のホストゾーンに関する情報を表示するには、前のコマンドに対する応答から `NextToken` の値を取得し、その値を `--starting-token` パラメーターに含めます。次に例を示します。

```
aws route53 list-hosted-zones --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListHostedZones](#)」の「」を参照してください。

list-query-logging-configs

次の例は、`list-query-logging-configs` を使用する方法を説明しています。

AWS CLI

クエリログ記録設定を一覧表示するには

次の `list-query-logging-configs` 例は、ホストゾーンの AWS アカウント内の最初の 100 件のクエリログ記録設定に関する情報を示しています `Z10X3WQEXAMPLE`。

```
aws route53 list-query-logging-configs \  
  --hosted-zone-id Z10X3WQEXAMPLE
```

出力:

```
{  
  "QueryLoggingConfigs": [  
    {  
      "Id": "964ff34e-ae03-4f06-80a2-9683cexample",  
      "HostedZoneId": "Z10X3WQEXAMPLE",  
      "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-  
group:/aws/route53/example.com:*"  
    }  
  ]  
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「クエリのログ記録」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[ListQueryLoggingConfigs](#)」の「」を参照してください。

list-resource-record-sets

次のコード例は、list-resource-record-sets を使用する方法を示しています。

AWS CLI

ホストゾーンのリソースレコードセットを一覧表示するには

次のlist-resource-record-setsコマンドは、指定されたホストゾーンの最初の 100 個のリソースレコードセットに関する概要情報を一覧表示します。

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

ホストゾーンに 100 を超えるリソースレコードセットが含まれている場合、または 100 未満のグループでリストする場合は、--maxitemsパラメータを含めます。例えば、リソースレコードセットを一度に 1 つずつ一覧表示するには、次のコマンドを使用します。

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
```

ホストゾーン内の次のリソースレコードセットに関する情報を表示するには、前のコマンドに対するレスポンスNextTokenからの値を取得し、--starting-tokenパラメータに含めます。

例：

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1  
--starting-token Z3M3LMPEXAMPLE
```

特定の名前のすべてのリソースレコードセットを表示するには、--queryパラメータを使用してフィルタリングします。例:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --  
query "ResourceRecordSets[?Name == 'example.domain.']"
```

- API 詳細については、AWS CLI「コマンドリファレンス[ListResourceRecordSets](#)」の「」を参照してください。

を使用した Route 53 ドメイン登録の例 AWS CLI

次のコード例は、Route 53 ドメイン登録 AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

check-domain-availability

次の例は、check-domain-availability を使用する方法を説明しています。

AWS CLI

Route 53 にドメイン名を登録できるかどうかを確認するには

次のcheck-domain-availabilityコマンドexample.comは、Route 53 を使用してドメイン名を登録できるかどうかに関する情報を返します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains check-domain-availability \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "Availability": "UNAVAILABLE"  
}
```

Route 53 は、.com や など、多数の最上位ドメイン (TLDs) をサポートしていますが、.jp、利用可能な をすべてサポートしているわけではありません。ドメインの可用性を確認し、Route 53 が をサポートしていない場合、check-domain-availability を返します。

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability operation: <top-level domain> tld is not supported.
```

Route 53 にドメインを登録するときに使用できる のリスト TLDs については、[Amazon Route 53 デベロッパーガイド](#) の「[Amazon Route 53 に登録できるドメイン](#)」を参照してください。Amazon Route 53 Amazon Route 53 でのドメインの登録の詳細については、Amazon Route 53 デベロッパーガイド」の「[新しいドメインの登録](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CheckDomainAvailability](#)」の「」を参照してください。

check-domain-transferability

次の例は、check-domain-transferability を使用する方法を説明しています。

AWS CLI

ドメインを Route 53 に転送できるかどうかを確認するには

次の check-domain-transferability コマンドは、ドメイン名を example.com Route 53 に転送できるかどうかに関する情報を返します。

このコマンドは us-east-1 リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は us-east-1、region パラメータを省略できます。

```
aws route53domains check-domain-transferability \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "Transferability": {  
    "Transferable": "UNTRANSFERABLE"  
  }  
}
```

詳細については、[Amazon Route 53デベロッパーガイド](#)の「[ドメインの登録の Amazon Route 53 への転送](#)」を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[CheckDomainTransferability](#)」の「」を参照してください。

delete-tags-for-domain

次のコード例は、delete-tags-for-domain を使用する方法を示しています。

AWS CLI

ドメインのタグを削除するには

次のdelete-tags-for-domainコマンドは、指定されたドメインから3つのタグを削除します。タグ値ではなく、タグキーのみを指定することに注意してください。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains delete-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --tags-to-delete accounting-key hr-key engineering-key
```

このコマンドでは何も出力されません。

タグが削除されたことを確認するには、[list-tags-for-domain](#) を実行できます。詳細については、[Amazon Route 53デベロッパーガイド](#)の「[Amazon Route 53 リソースのタグ付け](#)」を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTagsForDomain](#)」の「」を参照してください。

disable-domain-auto-renew

次の例は、disable-domain-auto-renew を使用する方法を説明しています。

AWS CLI

ドメインの自動更新を無効にするには

次の`disable-domain-auto-renew`コマンドは、ドメインの登録の有効期限`example.com`が切れる前にドメインを自動的に更新しないように Route 53 を設定します。

このコマンドは `us-east-1` リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は `us-east-1`、 `region` パラメータを省略できます。

```
aws route53domains disable-domain-auto-renew \  
  --region us-east-1 \  
  --domain-name example.com
```

このコマンドでは何も出力されません。

設定が変更されたことを確認するには、 [get-domain-detail](#) を実行します。自動更新が無効になっている場合、 の値は `AutoRenew` です `False`。自動更新の詳細については、Amazon Route 53 デベロッパーガイドの「ドメイン <<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html>> の登録の更新」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisableDomainAutoRenew](#)」の「」を参照してください。

`disable-domain-transfer-lock`

次の例は、`disable-domain-transfer-lock` を使用する方法を説明しています。

AWS CLI

ドメインの転送ロックを無効にするには

次の`disable-domain-transfer-lock`コマンドは、ドメインを別のレジストラに転送`example.com`できるように、ドメインの転送ロックを削除します。このコマンドは `clientTransferProhibited` ステータスを変更します。

このコマンドは `us-east-1` リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は `us-east-1`、 `region` パラメータを省略できます。

```
aws route53domains disable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:


```
{
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

転送ロックが変更されていることを確認するには、[get-domain-detail](#) を実行します。転送ロックが無効になっている場合、 の値には StatusList は含まれません clientTransferProhibited。

転送プロセスの詳細については、[Amazon Route 53 デベロッパーガイド](#) の「[Amazon Route 53 から別のレジストラへのドメインの転送](#) Amazon Route 53」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisableDomainTransferLock](#)」の「」を参照してください。

enable-domain-auto-renew

次の例は、enable-domain-auto-renew を使用する方法を説明しています。

AWS CLI

ドメインの自動更新を有効にするには

次のenable-domain-auto-renewコマンドは、ドメインの登録の有効期限が切れexample.comの前にドメインを自動的に更新するように Route 53 を設定します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains enable-domain-auto-renew \
  --region us-east-1 \
  --domain-name example.com
```

このコマンドでは何も出力されません。設定が変更されたことを確認するには、[get-domain-detail](#) を実行します。自動更新が有効になっている場合、 の値は AutoRenewですTrue。

自動更新の詳細については、Amazon Route 53 デベロッパーガイドの「ドメイン <<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html>> の登録の更新」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [EnableDomainAutoRenew](#)」の「」を参照してください。

enable-domain-transfer-lock

次の例は、enable-domain-transfer-lock を使用する方法を説明しています。

AWS CLI

ドメインで転送ロックを有効にするには

次のenable-domain-transfer-lockコマンドは、指定されたドメインをロックして、別のレジストラに転送できないようにします。このコマンドはclientTransferProhibitedステータスを変更します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains enable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

転送ロックが変更されたことを確認するには、 [get-domain-detail](#) を実行します。転送ロックを有効にすると、 の値には StatusListが含まれますclientTransferProhibited。

転送プロセスの詳細については、 [Amazon Route 53 デベロッパーガイド](#) の「[Amazon Route 53 から別のレジストラへのドメインの転送](#) Amazon Route 53」を参照してください。

- API 詳細については、AWS CLI 「 [コマンドリファレンスEnableDomainTransferLock](#)」の「」を参照してください。

get-contact-reachability-status

次のコード例は、get-contact-reachability-status を使用する方法を示しています。

AWS CLI

登録者の問い合わせが確認メールに回答したかどうかを判断するには

次の`get-contact-reachability-status`コマンドは、指定されたドメインの登録者の連絡先が確認 E メールに応答したかどうかに関する情報を返します。

このコマンドは `us-east-1` リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は `us-east-1`、`region` パラメータを省略できます。

```
aws route53domains get-contact-reachability-status \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "domainName": "example.com",  
  "status": "DONE"  
}
```

詳細については、[「Amazon Route 53 デベロッパーガイド」の「認証メールと確認メールの再送信」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[GetContactReachabilityStatus](#)」の「」を参照してください。

get-domain-detail

次のコード例は、`get-domain-detail` を使用する方法を示しています。

AWS CLI

指定されたドメインに関する詳細情報を取得するには

次の`get-domain-detail`コマンドは、指定されたドメインに関する詳細情報を表示します。

このコマンドは `us-east-1` リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は `us-east-1`、`region` パラメータを省略できます。

```
aws route53domains get-domain-detail \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{
  "DomainName": "example.com",
  "Nameservers": [
    {
      "Name": "ns-2048.awsdns-64.com",
      "GlueIps": []
    },
    {
      "Name": "ns-2049.awsdns-65.net",
      "GlueIps": []
    },
    {
      "Name": "ns-2050.awsdns-66.org",
      "GlueIps": []
    },
    {
      "Name": "ns-2051.awsdns-67.co.uk",
      "GlueIps": []
    }
  ],
  "AutoRenew": true,
  "AdminContact": {
    "FirstName": "Saanvi",
    "LastName": "Sarkar",
    "ContactType": "COMPANY",
    "OrganizationName": "Example",
    "AddressLine1": "123 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "ssarkar@example.com",
    "ExtraParams": []
  },
  "RegistrantContact": {
    "FirstName": "Alejandro",
    "LastName": "Rosalez",
    "ContactType": "COMPANY",
    "OrganizationName": "Example",
    "AddressLine1": "123 Main Street",
    "City": "Anytown",
```

```
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "arosalez@example.com",
    "ExtraParams": []
  },
  "TechContact": {
    "FirstName": "Wang",
    "LastName": "Xiulan",
    "ContactType": "COMPANY",
    "OrganizationName": "Example",
    "AddressLine1": "123 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "wxiulan@example.com",
    "ExtraParams": []
  },
  "AdminPrivacy": true,
  "RegistrantPrivacy": true,
  "TechPrivacy": true,
  "RegistrarName": "Amazon Registrar, Inc.",
  "WhoIsServer": "whois.registrar.amazon.com",
  "RegistrarUrl": "http://registrar.amazon.com",
  "AbuseContactEmail": "abuse@registrar.amazon.com",
  "AbuseContactPhone": "+1.2062661000",
  "CreationDate": 1444934889.601,
  "ExpirationDate": 1602787689.0,
  "StatusList": [
    "clientTransferProhibited"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDomainDetail](#)」の「」を参照してください。

get-domain-suggestions

次の例は、get-domain-suggestions を使用する方法を説明しています。

AWS CLI

推奨ドメイン名のリストを取得するには

次の`get-domain-suggestions`コマンドは、ドメイン名に基づいて推奨されるドメイン名のリストを表示します`example.com`。レスポンスには、使用可能なドメイン名のみが含まれます。このコマンドは `us-east-1` リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は `us-east-1`、`region` パラメータを省略できます。

```
aws route53domains get-domain-suggestions \  
  --region us-east-1 \  
  --domain-name example.com \  
  --suggestion-count 10 \  
  --only-available
```

出力:

```
{  
  "SuggestionsList": [  
    {  
      "DomainName": "egzaampal.com",  
      "Availability": "AVAILABLE"  
    },  
    {  
      "DomainName": "examplelaw.com",  
      "Availability": "AVAILABLE"  
    },  
    {  
      "DomainName": "examplehouse.net",  
      "Availability": "AVAILABLE"  
    },  
    {  
      "DomainName": "homeexample.net",  
      "Availability": "AVAILABLE"  
    },  
    {  
      "DomainName": "examplelist.com",  
      "Availability": "AVAILABLE"  
    },  
    {  
      "DomainName": "examplenews.net",  
      "Availability": "AVAILABLE"  
    },  
  ],  
}
```

```
{
  "DomainName": "officeexample.com",
  "Availability": "AVAILABLE"
},
{
  "DomainName": "exampleworld.com",
  "Availability": "AVAILABLE"
},
{
  "DomainName": "exampleart.com",
  "Availability": "AVAILABLE"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDomainSuggestions](#)」の「」を参照してください。

get-operation-detail

次のコード例は、get-operation-detail を使用する方法を示しています。

AWS CLI

オペレーションの現在のステータスを取得するには

一部のドメイン登録オペレーションは非同期的に動作し、完了する前にレスポンスを返します。これらのオペレーションは、現在のステータスを取得するために使用できるオペレーション ID を返します。次のget-operation-detail コマンドは、指定されたオペレーションのステータスを返します。

このコマンドは us-east-1 リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は us-east-1、region パラメータを省略できます。

```
aws route53domains get-operation-detail \
  --region us-east-1 \
  --operation-id edbd8d63-7fe7-4343-9bc5-54033example
```

出力:

```
{
  "OperationId": "edbd8d63-7fe7-4343-9bc5-54033example",
```

```
"Status": "SUCCESSFUL",
"DomainName": "example.com",
"Type": "DOMAIN_LOCK",
"SubmittedDate": 1573749367.864
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOperationDetail](#)」の「」を参照してください。

list-domains

次の例は、list-domains を使用する方法を説明しています。

AWS CLI

現在の AWS アカウントに登録されているドメインを一覧表示するには

次のlist-domainsコマンドは、現在の AWS アカウントに登録されているドメインに関する概要情報を一覧表示します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains list-domains
  --region us-east-1
```

出力:

```
{
  "Domains": [
    {
      "DomainName": "example.com",
      "AutoRenew": true,
      "TransferLock": true,
      "Expiry": 1602712345.0
    },
    {
      "DomainName": "example.net",
      "AutoRenew": true,
      "TransferLock": true,
      "Expiry": 1602723456.0
    },
  ],
}
```



```
{
  "DomainName": "example.org",
  "AutoRenew": true,
  "TransferLock": true,
  "Expiry": 1602734567.0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDomains](#)」の「」を参照してください。

list-operations

次のコード例は、list-operations を使用する方法を示しています。

AWS CLI

オペレーション ID を返すオペレーションのステータスを一覧表示するには

一部のドメイン登録オペレーションは非同期的に実行され、完了する前にレスポンスを返しません。これらのオペレーションは、現在のステータスを取得するために使用できるオペレーション ID を返します。次のlist-operationsコマンドは、現在のドメイン登録オペレーションに関するステータスを含む概要情報を一覧表示します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains list-operations
  --region us-east-1
```

出力:

```
{
  "Operations": [
    {
      "OperationId": "aab9822f-1da0-4bf3-8a15-fd4e0example",
      "Status": "SUCCESSFUL",
      "Type": "DOMAIN_LOCK",
      "SubmittedDate": 1455321739.986
    },
    {
```

```
    "OperationId": "c24379ed-76be-42f8-bdad-9379bexample",
    "Status": "SUCCESSFUL",
    "Type": "UPDATE_NAMESERVER",
    "SubmittedDate": 1468960475.109
  },
  {
    "OperationId": "f47e1297-ef9e-4c2b-ae1e-a5fcbexample",
    "Status": "SUCCESSFUL",
    "Type": "RENEW_DOMAIN",
    "SubmittedDate": 1473561835.943
  },
  {
    "OperationId": "75584f23-b15f-459e-aed7-dc6f5example",
    "Status": "SUCCESSFUL",
    "Type": "UPDATE_DOMAIN_CONTACT",
    "SubmittedDate": 1547501003.41
  }
]
```

出力には、オペレーション ID を返し、現在の AWS アカウントを使用して登録したすべてのドメインに対して実行したすべてのオペレーションが含まれます。指定した日付以降に送信したオペレーションのみを取得する場合は、`submitted-since` パラメータを含め、Unix 形式と協定世界時 () で日付を指定できます UTC。次のコマンドは、2020 年 1 UTC 月 1 日午前 12 時以降に送信されたすべてのオペレーションのステータスを取得します。

```
aws route53domains list-operations \
  --submitted-since 1577836800
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListOperations](#)」の「」を参照してください。

list-tags-for-domain

次のコード例は、`list-tags-for-domain` を使用する方法を示しています。

AWS CLI

ドメインのタグを一覧表示するには

次の `list-tags-for-domain` コマンドは、指定されたドメインに現在関連付けられているタグを一覧表示します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains list-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "TagList": [  
    {  
      "Key": "key1",  
      "Value": "value1"  
    },  
    {  
      "Key": "key2",  
      "Value": "value2"  
    }  
  ]  
}
```

詳細については、[Amazon Route 53デベロッパーガイド](#)の「[Amazon Route 53 リソースのタグ付け](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForDomain](#)」の「」を参照してください。

register-domain

次の例は、register-domain を使用する方法を説明しています。

AWS CLI

ドメインを登録するには

次のregister-domainコマンドはドメインを登録し、JSONフォーマットされたファイルからすべてのパラメータ値を取得します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains register-domain \  
--region us-east-1 \  
--cli-input-json file://register-domain.json
```

register-domain.json の内容:

```
{  
  "DomainName": "example.com",  
  "DurationInYears": 1,  
  "AutoRenew": true,  
  "AdminContact": {  
    "FirstName": "Martha",  
    "LastName": "Rivera",  
    "ContactType": "PERSON",  
    "OrganizationName": "Example",  
    "AddressLine1": "1 Main Street",  
    "City": "Anytown",  
    "State": "WA",  
    "CountryCode": "US",  
    "ZipCode": "98101",  
    "PhoneNumber": "+1.8005551212",  
    "Email": "mrivera@example.com"  
  },  
  "RegistrantContact": {  
    "FirstName": "Li",  
    "LastName": "Juan",  
    "ContactType": "PERSON",  
    "OrganizationName": "Example",  
    "AddressLine1": "1 Main Street",  
    "City": "Anytown",  
    "State": "WA",  
    "CountryCode": "US",  
    "ZipCode": "98101",  
    "PhoneNumber": "+1.8005551212",  
    "Email": "ljuan@example.com"  
  },  
  "TechContact": {  
    "FirstName": "Mateo",  
    "LastName": "Jackson",  
    "ContactType": "PERSON",  
    "OrganizationName": "Example",  
    "AddressLine1": "1 Main Street",  
    "City": "Anytown",
```

```
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "mjackson@example.com"
  },
  "PrivacyProtectAdminContact": true,
  "PrivacyProtectRegistrantContact": true,
  "PrivacyProtectTechContact": true
}
```

出力:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

オペレーションが成功したことを確認するには、`aws route53 get-operation-detail` を実行できます。詳細については、[get-operation-detail](#)「」を参照してください。

詳細については、[Amazon Route 53デベロッパーガイド](#)の「[新しいドメインの登録](#)」を参照してください。

の最上位ドメイン (TLDs) が の値を必要としExtraParams、有効な値については、Amazon Route 53APIリファレンス[ExtraParam](#)「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterDomain](#)」の「」を参照してください。

renew-domain

次のコード例は、`renew-domain` を使用する方法を示しています。

AWS CLI

ドメインを更新するには

次の`renew-domain`コマンドは、指定されたドメインを5年間更新します。の値を取得するには`current-expiry-year`、`get-domain-detail` コマンドを使用し、Unix形式`ExpirationDate`からの値を変換します。

このコマンドは us-east-1 リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は us-east-1、 region パラメータを省略できます。

```
aws route53domains renew-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --duration-in-years 5 \  
  --current-expiry-year 2020
```

出力:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

オペレーションが成功したことを確認するには、 を実行できます `get-operation-detail`。詳細については、「」を参照してください [get-operation-detail](#)。

.com や .org など、各最上位ドメイン (TLD) のレジストリは、ドメインを更新できる最大年数を制御します。ドメインの最大更新期間を取得するには、Amazon Route 53 デベロッパーガイドの「Amazon Route 53 に登録できるドメイン TLD」の「登録と更新期間」セクションを参照してください。 [Amazon Route 53](#)

詳細については、「[Amazon Route 53 デベロッパーガイド](#)」の「[ドメインの登録の更新](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス [RenewDomain](#)」の「」を参照してください。

resend-contact-reachability-email

次の例は、resend-contact-reachability-email を使用する方法を説明しています。

AWS CLI

登録者の連絡先の現在の E メールアドレスに確認 E メールを再送信するには

次の resend-contact-reachability-email コマンドは、example.com ドメインの登録者の連絡先の現在の E メールアドレスに確認 E メールを再送信します。

このコマンドは us-east-1 リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は us-east-1、 region パラメータを省略できます。

```
aws route53domains resend-contact-reachability-email \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{  
  "domainName": "example.com",  
  "emailAddress": "moliveira@example.com",  
  "isAlreadyVerified": true  
}
```

の値が `isAlreadyVerified` `true`、この例のように、登録者の連絡先は、指定された E メールアドレスに到達可能であることを既に確認しています。

詳細については、[「Amazon Route 53 デベロッパーガイド」の「認証メールと確認メールの再送信」](#)を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス [ResendContactReachabilityEmail](#)」の「」を参照してください。

retrieve-domain-auth-code

次の例は、`retrieve-domain-auth-code` を使用する方法を説明しています。

AWS CLI

ドメインの承認コードを取得して、ドメインを別のレジストラに転送できるようにするには

次の `retrieve-domain-auth-code` コマンドは、`example.com` ドメインの現在の認証コードを取得します。この値を別のドメインレジストラに渡すのは、ドメインをそのレジストラに転送する場合です。

このコマンドは `us-east-1` リージョンでのみ実行されます。デフォルトのリージョンが `us-east-1` に設定されている場合は `us-east-1`、`region` パラメータを省略できます。

```
aws route53domains retrieve-domain-auth-code \  
  --region us-east-1 \  
  --domain-name example.com
```

出力:

```
{
  "AuthCode": ")o!v3dJeXampLe"
}
```

詳細については、[Amazon Route 53 デベロッパーガイド](#)の「[Amazon Route 53 から別のレジストラへのドメインの転送](#) Amazon Route 53」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RetrieveDomainAuthCode](#)」の「」を参照してください。

transfer-domain

次のコード例は、transfer-domain を使用する方法を示しています。

AWS CLI

ドメインを Amazon Route 53 に転送するには

次のtransfer-domainコマンドは、JSON形式のファイルによって提供されるパラメータを使用して、ドメインを Route 53 に転送しますC:\temp\transfer-domain.json。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンがに設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains transfer-domain \
  --region us-east-1 \
  --cli-input-json file://C:\temp\transfer-domain.json
```

transfer-domain.json の内容:

```
{
  "DomainName": "example.com",
  "DurationInYears": 1,
  "Nameservers": [
    {
      "Name": "ns-2048.awsdns-64.com"
    },
    {
      "Name": "ns-2049.awsdns-65.net"
    },
    {
      "Name": "ns-2050.awsdns-66.org"
    }
  ]
}
```



```
    },
    {
      "Name": "ns-2051.awsdns-67.co.uk"
    }
  ],
  "AuthCode": ")o!v3dJeXampLe",
  "AutoRenew": true,
  "AdminContact": {
    "FirstName": "Martha",
    "LastName": "Rivera",
    "ContactType": "PERSON",
    "OrganizationName": "Example",
    "AddressLine1": "1 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "mrivera@example.com"
  },
  "RegistrantContact": {
    "FirstName": "Li",
    "LastName": "Juan",
    "ContactType": "PERSON",
    "OrganizationName": "Example",
    "AddressLine1": "1 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
    "Email": "ljuan@example.com"
  },
  "TechContact": {
    "FirstName": "Mateo",
    "LastName": "Jackson",
    "ContactType": "PERSON",
    "OrganizationName": "Example",
    "AddressLine1": "1 Main Street",
    "City": "Anytown",
    "State": "WA",
    "CountryCode": "US",
    "ZipCode": "98101",
    "PhoneNumber": "+1.8005551212",
```

```
    "Email": "mjackson@example.com"
  },
  "PrivacyProtectAdminContact": true,
  "PrivacyProtectRegistrantContact": true,
  "PrivacyProtectTechContact": true
}
```

出力:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

オペレーションが成功したことを確認するには、`aws route53domains get-operation-detail` を実行できます。詳細については、[get-operation-detail](#) 「」を参照してください。

詳細については、[Amazon Route 53デベロッパーガイド](#)の「[ドメインの登録を Amazon Route 53 に転送する](#)」を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「[コマンドリファレンスTransferDomain](#)」の「」を参照してください。

update-domain-contact-privacy

次の例は、`aws route53domains update-domain-contact-privacy` を使用する方法を説明しています。

AWS CLI

ドメインの連絡先のプライバシー設定を更新するには

次の`aws route53domains update-domain-contact-privacy` コマンドは、`example.com` ドメインの管理連絡先のプライバシー保護をオフにします。このコマンドは `us-east-1` リージョンでのみ実行されます。

デフォルトのリージョンがに設定されている場合は`us-east-1`、`region`パラメータを省略できます。

```
aws route53domains update-domain-contact-privacy \
  --region us-east-1 \
  --domain-name example.com \
  --no-admin-privacy
```

出力:

```
{
  "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

オペレーションが成功したことを確認するには、`aws route53domains get-operation-detail` を実行できます。詳細については、[get-operation-detail](#)「」を参照してください。

詳細については、[「Amazon Route 53 デベロッパーガイド」の「ドメインの連絡先情報のプライバシー保護の有効化または無効化」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateDomainContactPrivacy](#)」の「」を参照してください。

update-domain-contact

次のコード例は、`update-domain-contact` を使用する方法を示しています。

AWS CLI

ドメインの連絡先情報を更新するには

次の`update-domain-contact` コマンドは、JSON形式のファイルからパラメータを取得して、ドメインの連絡先情報を更新します。C:\temp\update-domain-contact.json。

このコマンドは us-east-1 リージョンでのみ実行されます。デフォルトのリージョンがに設定されている場合は us-east-1、`region` パラメータを省略できます。

```
aws route53domains update-domain-contact \
  --region us-east-1 \
  --cli-input-json file://C:\temp\update-domain-contact.json
```

update-domain-contact.json の内容:

```
{
  "AdminContact": {
    "AddressLine1": "101 Main Street",
    "AddressLine2": "Suite 1a",
    "City": "Seattle",
    "ContactType": "COMPANY",
```

```
"CountryCode": "US",
"Email": "w.xiulan@example.com",
"FirstName": "Wang",
"LastName": "Xiulan",
"OrganizationName": "Example",
"PhoneNumber": "+1.8005551212",
"State": "WA",
"ZipCode": "98101"
},
"DomainName": "example.com",
"RegistrantContact": {
  "AddressLine1": "101 Main Street",
  "AddressLine2": "Suite 1a",
  "City": "Seattle",
  "ContactType": "COMPANY",
  "CountryCode": "US",
  "Email": "w.xiulan@example.com",
  "FirstName": "Wang",
  "LastName": "Xiulan",
  "OrganizationName": "Example",
  "PhoneNumber": "+1.8005551212",
  "State": "WA",
  "ZipCode": "98101"
},
"TechContact": {
  "AddressLine1": "101 Main Street",
  "AddressLine2": "Suite 1a",
  "City": "Seattle",
  "ContactType": "COMPANY",
  "CountryCode": "US",
  "Email": "w.xiulan@example.com",
  "FirstName": "Wang",
  "LastName": "Xiulan",
  "OrganizationName": "Example",
  "PhoneNumber": "+1.8005551212",
  "State": "WA",
  "ZipCode": "98101"
}
}
```

出力:

```
{
```

```
"OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

オペレーションが成功したことを確認するには、[get-domain-detail](#) を実行できます。詳細については、Amazon Route 53デベロッパーガイドの「[ドメインの連絡先情報の更新](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDomainContact](#)」の「」を参照してください。

update-domain-nameservers

次のコード例は、update-domain-nameservers を使用する方法を示しています。

AWS CLI

ドメインのネームサーバーを更新するには

次のupdate-domain-nameserversコマンドは、ドメインのネームサーバーを更新します。

このコマンドは us-east-1リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合はus-east-1、 regionパラメータを省略できます。

```
aws route53domains update-domain-nameservers \
  --region us-east-1 \
  --domain-name example.com \
  --
nameservers Name=ns-1.awsdns-01.org Name=ns-2.awsdns-02.co.uk Name=ns-3.awsdns-03.net Name=ns-4.awsdns-04.com
```

出力:

```
{
  "OperationId": "f1691ec4-0e7a-489e-82e0-b19d3example"
}
```

オペレーションが成功したことを確認するには、[get-domain-detail](#) を実行できます。

詳細については、Amazon Route 53 デベロッパーガイドの「[ドメインのネームサーバーと Glue レコードの追加または変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDomainNameservers](#)」の「」を参照してください。

update-tags-for-domain

次の例は、update-tags-for-domain を使用する方法を説明しています。

AWS CLI

ドメインのタグを追加または更新するには

次のupdate-tags-for-domainコマンドは、example.com ドメインの 2 つのキーと対応する値を追加または更新します。キーの値を更新するには、キーと新しい値を含めるだけです。タグを追加または更新できるのは、一度に 1 つのドメインのみです。

このコマンドは us-east-1 リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は us-east-1、region パラメータを省略できます。

```
aws route53domains update-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --tags-to-update "Key=key1,Value=value1" "Key=key2,Value=value2"
```

このコマンドでは何も出力されません。タグが追加または更新されたことを確認するには、[list-tags-for-domain](#) を実行します。

詳細については、[Amazon Route 53 デベロッパーガイド](#) の「[Amazon Route 53 リソースのタグ付け](#)」を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「[コマンドリファレンス UpdateTagsForDomain](#)」の「」を参照してください。

view-billing

次の例は、view-billing を使用する方法を説明しています。

AWS CLI

現在の AWS アカウントのドメイン登録料金の請求情報を取得するには

次のview-billingコマンドは、Unix 時間で 2018 年 1 月 1 日 (1514764800) から 2019 年 12 月 31 日深夜 (1577836800) の間、現在のアカウントのドメイン関連の請求レコードをすべて返します。

このコマンドは us-east-1 リージョンでのみ実行されます。デフォルトのリージョンが に設定されている場合は us-east-1、 region パラメータを省略できます。

```
aws route53domains view-billing \  
  --region us-east-1 \  
  --start-time 1514764800 \  
  --end-time 1577836800
```

出力:

```
{  
  "BillingRecords": [  
    {  
      "DomainName": "example.com",  
      "Operation": "RENEW_DOMAIN",  
      "InvoiceId": "149962827",  
      "BillDate": 1536618063.181,  
      "Price": 12.0  
    },  
    {  
      "DomainName": "example.com",  
      "Operation": "RENEW_DOMAIN",  
      "InvoiceId": "290913289",  
      "BillDate": 1568162630.884,  
      "Price": 12.0  
    }  
  ]  
}
```

詳細については、Amazon Route 53 API リファレンス [ViewBilling](#) の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ViewBilling](#)」の「」を参照してください。

を使用した Route 53 プロファイルの例 AWS CLI

次のコード例は、Route 53 プロファイル AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-profile

次のコード例は、associate-profile を使用する方法を示しています。

AWS CLI

プロファイルを関連付けるには

次のassociate-profile例では、プロファイルを に関連付けますVPC。

```
aws route53profiles associate-profile \  
  --name test-association \  
  --profile-id rp-4987774726example \  
  --resource-id vpc-0af3b96b3example
```

出力:

```
{  
  "ProfileAssociation": {  
    "CreationTime": 1710851336.527,  
    "Id": "rpassoc-489ce212fexample",  
    "ModificationTime": 1710851336.527,  
    "Name": "test-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceId": "vpc-0af3b96b3example",  
    "Status": "CREATING",  
    "StatusMessage": "Creating Profile Association"  
  }  
}
```



```
}
```

詳細については、[「Amazon Route 53 デベロッパーガイド」の「プロファイルの使用」](#)を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateProfile](#)」の「」を参照してください。

associate-resource-to-profile

次のコード例は、`associate-resource-to-profile` を使用する方法を示しています。

AWS CLI

リソースをプロファイルに関連付けるには

次の`associate-resource-to-profile`例では、DNSファイアウォールルールグループを 102 の優先度でプロファイルに関連付けます。

```
aws route53profiles associate-resource-to-profile \  
  --name test-resource-association \  
  --profile-id rp-4987774726example \  
  --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-  
group/rslvr-frg-cfe7f72example \  
  --resource-properties '{"priority": 102}'
```

出力:

```
{  
  "ProfileResourceAssociation": {  
    "CreationTime": 1710851216.613,  
    "Id": "rpr-001913120a7example",  
    "ModificationTime": 1710851216.613,  
    "Name": "test-resource-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-  
rule-group/rslvr-frg-cfe7f72example",  
    "ResourceProperties": '{"priority":102}',  
    "ResourceType": "FIREWALL_RULE_GROUP",  
    "Status": "UPDATING",
```

```
    "StatusMessage": "Updating the Profile to DNS Firewall rule group
association"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateResourceToProfile](#)」の「」を参照してください。

create-profile

次のコード例は、create-profile を使用する方法を示しています。

AWS CLI

プロファイルを作成するには

次のcreate-profile例では、プロファイルを作成します。

```
aws route53profiles create-profile \
  --name test
```

出力:

```
{
  "Profile": {
    "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
    "ClientToken": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE111111",
    "CreationTime": 1710850903.578,
    "Id": "rp-6ffe47d5example",
    "ModificationTime": 1710850903.578,
    "Name": "test",
    "OwnerId": "123456789012",
    "ShareStatus": "NOT_SHARED",
    "Status": "COMPLETE",
    "StatusMessage": "Created Profile"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProfile](#)」の「」を参照してください。

delete-profile

次の例は、delete-profile を使用する方法を説明しています。

AWS CLI

プロファイルを削除するには

次のdelete-profile例では、プロファイルを削除します。

```
aws route53profiles delete-profile \  
  --profile-id rp-6ffe47d5example
```

出力:

```
{  
  "Profile": {  
    "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/  
rp-6ffe47d5example",  
    "ClientToken": "0a15fec0-05d9-4f78-bec0-EXAMPLE11111",  
    "CreationTime": 1710850903.578,  
    "Id": "rp-6ffe47d5example",  
    "ModificationTime": 1710850903.578,  
    "Name": "test",  
    "OwnerId": "123456789012",  
    "ShareStatus": "NOT_SHARED",  
    "Status": "DELETED",  
    "StatusMessage": "Deleted Profile"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProfile](#)」の「」を参照してください。

disassociate-profile

次のコード例は、disassociate-profile を使用する方法を示しています。

AWS CLI

プロファイルの関連付けを解除するには

次のdisassociate-profile例では、プロファイルと の関連付けを解除しますVPC。

```
aws route53profiles disassociate-profile \  
  --profile-id rp-4987774726example \  
  --resource-id vpc-0af3b96b3example
```

出力:

```
{  
  "ProfileAssociation": {  
    "CreationTime": 1710851336.527,  
    "Id": "rpassoc-489ce212fexample",  
    "ModificationTime": 1710851401.362,  
    "Name": "test-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceId": "vpc-0af3b96b3example",  
    "Status": "DELETING",  
    "StatusMessage": "Deleting Profile Association"  
  }  
}
```

- API 詳細については、AWS CLI 「 コマンドリファレンス[DisassociateProfile](#)」の「」を参照してください。

disassociate-resource-from-profile

次のコード例は、disassociate-resource-from-profile を使用する方法を示しています。

AWS CLI

プロファイルからリソースの関連付けを解除するには

次のdisassociate-resource-from-profile例では、プロファイルからDNSファイアウォールルールグループの関連付けを解除します。

```
aws route53profiles disassociate-resource-from-profile \  
  --profile-id rp-4987774726example \  
  --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example
```

出力:

```
{
  "ProfileResourceAssociation": {
    "CreationTime": 1710851216.613,
    "Id": "rpr-001913120a7example",
    "ModificationTime": 1710852624.36,
    "Name": "test-resource-association",
    "OwnerId": "123456789012",
    "ProfileId": "rp-4987774726example",
    "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
    "ResourceProperties": "{\"priority\":105}",
    "ResourceType": "FIREWALL_RULE_GROUP",
    "Status": "DELETING",
    "StatusMessage": "Deleting the Profile to DNS Firewall rule group
association"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateResourceFromProfile](#)」の「」を参照してください。

get-profile-association

次の例は、get-profile-association を使用する方法を説明しています。

AWS CLI

プロファイルの関連付けに関する情報を取得するには

以下は、指定されたプロファイルの関連付けに関する情報get-profile-associationを返します。

```
aws route53profiles get-profile-association \
  --profile-association-id rpassoc-489ce212fexample
```

出力:

```
{
  "ProfileAssociation": {
```

```
    "CreationTime": 1709338817.148,  
    "Id": "rrpassoc-489ce212fexample",  
    "ModificationTime": 1709338974.772,  
    "Name": "test-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceId": "vpc-0af3b96b3example",  
    "Status": "COMPLETE",  
    "StatusMessage": "Created Profile Association"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetProfileAssociation](#)」の「」を参照してください。

get-profile-resource-association

次の例は、get-profile-resource-association を使用する方法を説明しています。

AWS CLI

プロファイルに関連付けられたリソースに関する情報を取得するには

以下は、指定されたリソースの関連付けに関する情報をプロファイルにget-profile-resource-association返します。

```
aws route53profiles get-profile-resource-association \  
  --profile-resource-association-id rpr-001913120a7example
```

出力:

```
{  
  "ProfileResourceAssociation": {  
    "CreationTime": 1710851216.613,  
    "Id": "rpr-001913120a7example",  
    "ModificationTime": 1710852303.798,  
    "Name": "test-resource-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-  
rule-group/rslvr-frg-cfe7f72example",
```

```
    "ResourceProperties": "{\"priority\":105}",
    "ResourceType": "FIREWALL_RULE_GROUP",
    "Status": "COMPLETE",
    "StatusMessage": "Completed creation of Profile to DNS Firewall rule group
association"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetProfileResourceAssociation](#)」の「」を参照してください。

get-profile

次のコード例は、get-profile を使用する方法を示しています。

AWS CLI

プロファイルに関する情報を取得するには

以下は、指定されたプロファイルに関する情報get-profileを返します。

```
aws route53profiles get-profile \
  --profile-id rp-4987774726example
```

出力:

```
{
  "Profile": {
    "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
    "ClientToken": "0cbc5ae7-4921-4204-bea9-EXAMPLE11111",
    "CreationTime": 1710851044.288,
    "Id": "rp-4987774726example",
    "ModificationTime": 1710851044.288,
    "Name": "test",
    "OwnerId": "123456789012",
    "ShareStatus": "NOT_SHARED",
    "Status": "COMPLETE",
    "StatusMessage": "Created Profile"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetProfile](#)」の「」を参照してください。

list-profile-associations

次の例は、list-profile-associations を使用する方法を説明しています。

AWS CLI

プロファイルの関連付けを一覧表示するには

AWS アカウントのプロファイルの関連付けをlist-profile-associations以下に示します。

```
aws route53profiles list-profile-associations
```

出力:

```
{
  "ProfileAssociations": [
    {
      "CreationTime": 1709338817.148,
      "Id": "rpassoc-489ce212fexample",
      "ModificationTime": 1709338974.772,
      "Name": "test-association",
      "OwnerId": "123456789012",
      "ProfileId": "rp-4987774726example",
      "ResourceId": "vpc-0af3b96b3example",
      "Status": "COMPLETE",
      "StatusMessage": "Created Profile Association"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProfileAssociations](#)」の「」を参照してください。

list-profile-resource-associations

次の例は、list-profile-resource-associations を使用する方法を説明しています。

AWS CLI

プロファイルリソースの関連付けを一覧表示するには

以下に `list-profile-resource-associations`、指定されたプロファイルのプロファイルリソースの関連付けを示します。

```
aws route53profiles list-profile-resource-associations \  
  --profile-id rp-4987774726example
```

出力:

```
{  
  "ProfileResourceAssociations": [  
    {  
      "CreationTime": 1710851216.613,  
      "Id": "rpr-001913120a7example",  
      "ModificationTime": 1710851216.613,  
      "Name": "test-resource-association",  
      "OwnerId": "123456789012",  
      "ProfileId": "rp-4987774726example",  
      "ResourceArn": "arn:aws:route53resolver:us-  
east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example",  
      "ResourceProperties": "{\"priority\":102}",  
      "ResourceType": "FIREWALL_RULE_GROUP",  
      "Status": "COMPLETE",  
      "StatusMessage": "Completed creation of Profile to DNS Firewall rule  
group association"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListProfileResourceAssociations](#)」の「」を参照してください。

list-profiles

次の例は、`list-profiles` を使用する方法を説明しています。

AWS CLI

プロファイルを一覧表示するには

以下は、AWS アカウントのプロファイルを一覧表示し、それらに関する追加情報を表示します。

```
aws route53profiles list-profiles
```

出力:

```
{
  "ProfileSummaries": [
    {
      "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
      "Id": "rp-4987774726example",
      "Name": "test",
      "ShareStatus": "NOT_SHARED"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProfiles](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースのタグを一覧表示するには

以下に、指定されたリソースのタグlist-tags-for-resourceを示します。

```
aws route53profiles list-tags-for-resource \
  --resource-arn arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example
```

出力:

```
{
  "Tags": {
```

```
    "my-key-2": "my-value-2",  
    "my-key-1": "my-value-1"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

update-profile-resource-association

次のコード例は、update-profile-resource-association を使用する方法を示しています。

AWS CLI

プロファイルに関連付けられたリソースを更新するには

次に、プロファイルに関連付けられている DNS Firewall ルールグループの優先度update-profile-resource-associationを更新します。

```
aws route53profiles update-profile-resource-association \  
  --profile-resource-association-id rpr-001913120a7example \  
  --resource-properties "{\"priority\": 105}"
```

出力:

```
{  
  "ProfileResourceAssociation": {  
    "CreationTime": 1710851216.613,  
    "Id": "rpr-001913120a7example",  
    "ModificationTime": 1710852303.798,  
    "Name": "test-resource-association",  
    "OwnerId": "123456789012",  
    "ProfileId": "rp-4987774726example",  
    "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-  
rule-group/rslvr-frg-cfe7f72example",  
    "ResourceProperties": "{\"priority\":105}",  
    "ResourceType": "FIREWALL_RULE_GROUP",  
    "Status": "UPDATING",  
    "StatusMessage": "Updating the Profile to DNS Firewall rule group  
association"  
  }  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateProfileResourceAssociation](#)」の「」を参照してください。

を使用した Route 53 Resolver の例 AWS CLI

次のコード例は、Route 53 Resolver AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-firewall-rule-group

次のコード例は、associate-firewall-rule-group を使用する方法を示しています。

AWS CLI

ファイアウォールルールグループを に関連付けるには VPC

次のassociate-firewall-rule-group例では、DNSFirewall ルールグループを Amazon に関連付けますVPC。

```
aws route53resolver associate-firewall-rule-group \  
  --name test-association \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample \  
  --vpc-id vpc-31e92222 \  
  --priority 101
```

出力:

```
{
  "FirewallRuleGroupAssociation": {
    "Id": "rslvr-frgassoc-57e8873d7example",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "VpcId": "vpc-31e92222",
    "Name": "test-association",
    "Priority": 101,
    "MutationProtection": "DISABLED",
    "Status": "UPDATING",
    "StatusMessage": "Creating Firewall Rule Group Association",
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:47:48.755768Z",
    "ModificationTime": "2021-05-25T21:47:48.755768Z"
  }
}
```

詳細については、Amazon Route [53 デベロッパーガイドの「VPCと Route 53 Resolver DNS Firewall ルールグループ間の関連付けの管理Route 53」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateFirewallRuleGroup](#)」の「」を参照してください。

associate-resolver-endpoint-ip-address

次のコード例は、associate-resolver-endpoint-ip-address を使用する方法を示しています。

AWS CLI

別の IP アドレスを Resolver エンドポイントに関連付けるには

次のassociate-resolver-endpoint-ip-address例では、別の IP アドレスをインバウンド Resolver エンドポイントに関連付けます。サブネット ID のみを指定し、--ip-addressパラメータから IP アドレスを省略すると、Resolver は指定したサブネットで使用可能な IP アドレスの中から、IP アドレスを選択します。

```
aws route53resolver associate-resolver-endpoint-ip-address \
  --resolver-endpoint-id rslvr-in-497098ad5example \
  --ip-address="SubnetId=subnet-12d8exam,Ip=192.0.2.118"
```

出力:

```
{
  "ResolverEndpoint": {
    "Id": "rslvr-in-497098ad5example",
    "CreatorRequestId": "AWSConsole.25.0123456789",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-497098ad5example",
    "Name": "my-inbound-endpoint",
    "SecurityGroupIds": [
      "sg-05cd7b25d6example"
    ],
    "Direction": "INBOUND",
    "IpAddressCount": 3,
    "HostVPCId": "vpc-304bexam",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Resolver Endpoint",
    "CreationTime": "2020-01-02T23:25:45.538Z",
    "ModificationTime": "2020-01-02T23:25:45.538Z"
  }
}
```

詳細については、Amazon Route 53 デベロッパーガイドの「[インバウンドエンドポイントを作成または編集するときに指定する値](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateResolverEndpointIpAddress](#)」の「」を参照してください。

associate-resolver-rule

次の例は、associate-resolver-rule を使用する方法を説明しています。

AWS CLI

Resolver ルールを に関連付けるには VPC

次のassociate-resolver-rule例では、Resolver ルールを Amazon に関連付けますVPC。コマンドを実行すると、Resolver は転送されるDNSクエリのドメイン名など、ルールの設定に基づいてネットワークへのクエリの転送を開始します。

```
aws route53resolver associate-resolver-rule \
  --name my-resolver-rule-association \
```

```
--resolver-rule-id rslvr-rr-42b60677c0example \  
--vpc-id vpc-304bexam
```

出力:

```
{  
  "ResolverRuleAssociation": {  
    "Id": "rslvr-rrassoc-d61cbb2c8bexample",  
    "ResolverRuleId": "rslvr-rr-42b60677c0example",  
    "Name": "my-resolver-rule-association",  
    "VPCId": "vpc-304bexam",  
    "Status": "CREATING",  
    "StatusMessage": "[Trace id: 1-5dc5a8fa-ec2cc480d2ef07617example] Creating  
the association."  
  }  
}
```

詳細については、[「Amazon Route 53 DNS デベロッパーガイド」の「ネットワークへのアウトバウンドクエリの転送」](#)を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateResolverRule](#)」の「」を参照してください。

create-firewall-domain-list

次のコード例は、create-firewall-domain-list を使用する方法を示しています。

AWS CLI

Route 53 Resolver DNS Firewall ドメインリストを作成するには

次のcreate-firewall-domain-list例では、AWS アカウントに test という名前の Route 53 Resolver DNS Firewall ドメインリストを作成します。

```
aws route53resolver create-firewall-domain-list \  
--creator-request-id my-request-id \  
--name test
```

出力:

```
{
```

```
"FirewallDomainList": {
  "Id": "rslvr-fdl-d61cbb2cbexample",
  "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/
rslvr-fdl-d61cbb2cbexample",
  "Name": "test",
  "DomainCount": 0,
  "Status": "COMPLETE",
  "StatusMessage": "Created Firewall Domain List",
  "CreatorRequestId": "my-request-id",
  "CreationTime": "2021-05-25T15:55:51.115365Z",
  "ModificationTime": "2021-05-25T15:55:51.115365Z"
}
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の「[独自のドメインリストの管理](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFirewallDomainList](#)」の「」を参照してください。

create-firewall-rule-group

次のコード例は、create-firewall-rule-group を使用する方法を示しています。

AWS CLI

Firewall ルールグループを作成するには

次のcreate-firewall-rule-group例では、DNSファイアウォールルールグループを作成します。

```
aws route53resolver create-firewall-rule-group \
  --creator-request-id my-request-id \
  --name test
```

出力:

```
{
  "FirewallRuleGroup": {
    "Id": "rslvr-frg-47f93271fexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
```



```
    "Name": "test",
    "RuleCount": 0,
    "Status": "COMPLETE",
    "StatusMessage": "Created Firewall Rule Group",
    "OwnerId": "123456789012",
    "CreatorRequestId": "my-request-id",
    "ShareStatus": "NOT_SHARED",
    "CreationTime": "2021-05-25T18:59:26.490017Z",
    "ModificationTime": "2021-05-25T18:59:26.490017Z"
  }
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「ファイアウォールでのルールグループとルールの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[CreateFirewallRuleGroup](#)」の「」を参照してください。

create-firewall-rule

次の例は、create-firewall-rule を使用する方法を説明しています。

AWS CLI

ファイアウォールルールを作成するには

次のcreate-firewall-rule例では、DNSファイアウォールドメインリストにリストされているドメインのDNSファイアウォールルールにファイアウォールルールを作成します。

```
aws route53resolver create-firewall-rule \
  --name allow-rule \
  --firewall-rule-group-id rslvr-frg-47f93271fexample \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
  --priority 101 \
  --action ALLOW
```

出力:

```
{
  "FirewallRule": {
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
```

```

    "Name": "allow-rule",
    "Priority": 101,
    "Action": "ALLOW",
    "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:44:00.346093Z",
    "ModificationTime": "2021-05-25T21:44:00.346093Z"
  }
}

```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「ファイアウォールでのルールグループとルールの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[CreateFirewallRule](#)」の「」を参照してください。

create-resolver-endpoint

次の例は、create-resolver-endpoint を使用する方法を説明しています。

AWS CLI

インバウンドリゾルバーエンドポイントを作成するには

次のcreate-resolver-endpoint例では、インバウンド Resolver エンドポイントを作成します。同じコマンドを使用して、インバウンドエンドポイントとアウトバウンドエンドポイントの両方を作成できます。

```
aws route53resolver create-resolver-endpoint --name my-inbound-endpoint --creator-request-id 2020-01-01-18:47 --security-group-ids 「sg-f62bexam」 --direction INBOUND --ip-addresses SubnetId=subnet-ba47exam,Ip=192.0.2.255 SubnetId=subnet-12d8exam,Ip=192.0.2.254
```

出力:

```

{
  "ResolverEndpoint": {
    "Id": "rslvr-in-f9ab8a03f1example",
    "CreatorRequestId": "2020-01-01-18:47",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/rslvr-in-f9ab8a03f1example",
    "Name": "my-inbound-endpoint",
    "SecurityGroupIds": [
      "sg-f62bexam"
    ]
  }
}

```

```
    ],
    "Direction": "INBOUND",
    "IpAddressCount": 2,
    "HostVPCId": "vpc-304examp",
    "Status": "CREATING",
    "StatusMessage": "[Trace id: 1-5dc1ff84-f3477826e4a190025example] Creating
the Resolver Endpoint",
    "CreationTime": "2020-01-01T23:02:29.583Z",
    "ModificationTime": "2020-01-01T23:02:29.583Z"
  }
}
```

アウトバウンド Resolver エンドポイントを作成するには

次のcreate-resolver-endpoint例では、JSON形式のドキュメントの値を使用してアウトバウンドリゾルバーエンドポイントを作成しますcreate-outbound-resolver-endpoint.json。

```
aws route53resolver create-resolver-endpoint \
  --cli-input-json file:///c:\temp\create-outbound-resolver-endpoint.json
```

create-outbound-resolver-endpoint.json の内容:

```
{
  "CreatorRequestId": "2020-01-01-18:47",
  "Direction": "OUTBOUND",
  "IpAddresses": [
    {
      "Ip": "192.0.2.255",
      "SubnetId": "subnet-ba47exam"
    },
    {
      "Ip": "192.0.2.254",
      "SubnetId": "subnet-12d8exam"
    }
  ],
  "Name": "my-outbound-endpoint",
  "SecurityGroupIds": [ "sg-05cd7b25d6example" ],
  "Tags": [
    {
      "Key": "my-key-name",
      "Value": "my-key-value"
    }
  ]
}
```

```
    }  
  ]  
}
```

詳細については、Amazon Route 53デベロッパーガイド」の[VPCs「とネットワーク間のDNSクエリの解決」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[CreateResolverEndpoint](#)」の「」を参照してください。

create-resolver-rule

次の例は、create-resolver-rule を使用する方法を説明しています。

AWS CLI

Resolver ルールを作成するには

次のcreate-resolver-rule例では、リゾルバー転送ルールを作成します。このルールは、アウトバウンドエンドポイント rslvr-out-d5e5920e37example を使用して、のDNSクエリexample.comを IP アドレス 10.24.8.75 および 10.24.8.156 に転送します。

```
aws route53resolver create-resolver-rule \  
  --creator-request-id 2020-01-02-18:47 \  
  --domain-name example.com \  
  --name my-rule \  
  --resolver-endpoint-id rslvr-out-d5e5920e37example \  
  --rule-type FORWARD \  
  --target-ips "Ip=10.24.8.75" "Ip=10.24.8.156"
```

出力:

```
{  
  "ResolverRule": {  
    "Status": "COMPLETE",  
    "RuleType": "FORWARD",  
    "ResolverEndpointId": "rslvr-out-d5e5920e37example",  
    "Name": "my-rule",  
    "DomainName": "example.com.",  
    "CreationTime": "2022-05-10T21:35:30.923187Z",  
    "TargetIps": [  
      {
```

```

        "Ip": "10.24.8.75",
        "Port": 53
    },
    {
        "Ip": "10.24.8.156",
        "Port": 53
    }
],
"CreatorRequestId": "2022-05-10-16:33",
"ModificationTime": "2022-05-10T21:35:30.923187Z",
"ShareStatus": "NOT_SHARED",
"Arn": "arn:aws:route53resolver:us-east-1:111117012054:resolver-rule/rslvr-rr-b1e0b905e93611111",
"OwnerId": "111111111111",
"Id": "rslvr-rr-rslvr-rr-b1e0b905e93611111",
"StatusMessage": "[Trace id: 1-22222222-3e56afcc71a3724664f22e24]
Successfully created Resolver Rule."
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateResolverRule](#)」の「」を参照してください。

delete-firewall-domain-list

次のコード例は、delete-firewall-domain-list を使用する方法を示しています。

AWS CLI

Route 53 Resolver DNS Firewall ドメインリストを削除するには

次のdelete-firewall-domain-list例では、AWS アカウントの Route 53 Resolver DNS Firewall ドメインリストを test という名前で削除します。

```
aws route53resolver delete-firewall-domain-list \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

出力:

```
{
  "FirewallDomainList": {
    "Id": "rslvr-fdl-9e956e9ffexample",
```

```
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/
rslvr-fdl-9e956e9ffexample",
    "Name": "test",
    "DomainCount": 6,
    "Status": "DELETING",
    "StatusMessage": "Deleting the Firewall Domain List",
    "CreatorRequestId": "my-request-id",
    "CreationTime": "2021-05-25T15:55:51.115365Z",
    "ModificationTime": "2021-05-25T18:58:05.588024Z"
  }
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の「[独自のドメインリストの管理](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFirewallDomainList](#)」の「」を参照してください。

delete-firewall-rule-group

次の例は、delete-firewall-rule-group を使用する方法を説明しています。

AWS CLI

ファイアウォールルールグループを削除するには

次のdelete-firewall-rule-group例では、ファイアウォールルールグループを削除します。

```
aws route53resolver delete-firewall-rule-group \
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

出力:

```
{
  "FirewallRuleGroup": {
    "Id": "rslvr-frg-47f93271fexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
    "Name": "test",
    "RuleCount": 0,
    "Status": "UPDATING",
```

```

    "StatusMessage": "Updating Firewall Rule Group",
    "OwnerId": "123456789012",
    "CreatorRequestId": "my-request-id",
    "ShareStatus": "NOT_SHARED",
    "CreationTime": "2021-05-25T18:59:26.490017Z",
    "ModificationTime": "2021-05-25T21:51:53.028688Z"
  }
}

```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「ファイアウォールでのルールグループとルールの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteFirewallRuleGroup](#)」の「」を参照してください。

delete-firewall-rule

次の例は、delete-firewall-rule を使用する方法を説明しています。

AWS CLI

ファイアウォールルールを削除するには

次のdelete-firewall-rule例では、指定されたファイアウォールルールを削除します。

```

aws route53resolver delete-firewall-rule \
  --firewall-rule-group-id rslvr-frg-47f93271fexample \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample

```

出力:

```

{
  "FirewallRule": {
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
    "Name": "allow-rule",
    "Priority": 102,
    "Action": "ALLOW",
    "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
    "CreationTime": "2021-05-25T21:44:00.346093Z",
    "ModificationTime": "2021-05-25T21:45:59.611600Z"
  }
}

```

```
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「ファイアウォールでのルールグループとルールの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteFirewallRule](#)」の「」を参照してください。

delete-resolver-endpoint

次のコード例は、delete-resolver-endpoint を使用方法を示しています。

AWS CLI

Resolver エンドポイントを削除するには

次のdelete-resolver-endpoint例では、指定されたエンドポイントを削除します。

重要 インバウンドエンドポイントを削除すると、ネットワークからのDNSクエリは、エンドポイントでVPC指定した の Resolver に転送されなくなります。アウトバウンドエンドポイントを削除すると、Resolver は削除されたアウトバウンドエンドポイントを指定するルールの からネットワークVPCへのDNSクエリの転送を停止します。

```
aws route53resolver delete-resolver-endpoint \  
  --resolver-endpoint-id rslvr-in-497098ad59example
```

出力:

```
{  
  "ResolverEndpoint": {  
    "Id": "rslvr-in-497098ad59example",  
    "CreatorRequestId": "AWSConsole.25.157290example",  
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/  
rslvr-in-497098ad59example",  
    "Name": "my-inbound-endpoint",  
    "SecurityGroupIds": [  
      "sg-05cd7b25d6example"  
    ],  
    "Direction": "INBOUND",  
    "IpAddressCount": 5,  
    "HostVPCId": "vpc-304bexam",  
    "Status": "DELETING",
```



```

    "StatusMessage": "[Trace id: 1-5dc5b658-811b5be0922bbc382example] Deleting
    ResolverEndpoint.",
    "CreationTime": "2020-01-01T23:25:45.538Z",
    "ModificationTime": "2020-01-02T23:25:45.538Z"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResolverEndpoint](#)」の「」を参照してください。

delete-resolver-rule

次のコード例は、delete-resolver-rule を使用する方法を示しています。

AWS CLI

リゾルバールールを削除するには

次のdelete-resolver-rule例では、指定されたルールを削除します。

注 ルールが に関連付けられている場合はVPCs、削除VPCsする前に、まず からルールの関連付けを解除する必要があります。

```

aws route53resolver delete-resolver-rule \
  --resolver-rule-id rslvr-rr-5b3809426bexample

```

出力:

```

{
  "ResolverRule": {
    "Id": "rslvr-rr-5b3809426bexample",
    "CreatorRequestId": "2020-01-03-18:47",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-rr-5b3809426bexample",
    "DomainName": "zenith.example.com.",
    "Status": "DELETING",
    "StatusMessage": "[Trace id: 1-5dc5e05b-602e67b052cb74f05example] Deleting
    Resolver Rule.",
    "RuleType": "FORWARD",
    "Name": "my-resolver-rule",
    "TargetIps": [
      {

```

```

        "Ip": "192.0.2.50",
        "Port": 53
    }
],
"ResolverEndpointId": "rslvr-out-d5e5920e3example",
"OwnerId": "111122223333",
"ShareStatus": "NOT_SHARED"
}
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteResolverRule](#)」の「」を参照してください。

disassociate-firewall-rule-group

次の例は、disassociate-firewall-rule-group を使用する方法を説明しています。

AWS CLI

ファイアウォールルールグループの からの関連付けを解除するには VPC

次のdisassociate-firewall-rule-group例では、Amazon から DNS Firewall ルールグループの関連付けを解除しますVPC。

```
aws route53resolver disassociate-firewall-rule-group \
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

出力:

```

{
  "FirewallRuleGroupAssociation": {
    "Id": "rslvr-frgassoc-57e8873d7example",
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
    "VpcId": "vpc-31e92222",
    "Name": "test-association",
    "Priority": 103,
    "MutationProtection": "DISABLED",
    "Status": "DELETING",
    "StatusMessage": "Deleting the Firewall Rule Group Association",
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
  }
}

```

```

    "CreationTime": "2021-05-25T21:47:48.755768Z",
    "ModificationTime": "2021-05-25T21:51:02.377887Z"
  }
}

```

詳細については、[「Amazon Route 53 デベロッパーガイド」の「VPCと Route 53 Resolver DNS Firewall ルールグループ間の関連付けの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateFirewallRuleGroup](#)」の「」を参照してください。

disassociate-resolver-endpoint-ip-address

次のコード例は、disassociate-resolver-endpoint-ip-address を使用する方法を示しています。

AWS CLI

Resolver エンドポイントから IP アドレスの関連付けを解除するには

次のdisassociate-resolver-endpoint-ip-address例では、指定された Resolver インバウンドエンドポイントまたはアウトバウンドエンドポイントから IP アドレスを削除します。

注 エンドポイントには少なくとも 2 つの IP アドレスが必要です。エンドポイントに現在 2 つの IP アドレスしかなく、1 つのアドレスを別のアドレスに置き換える場合は、まず [associate-resolver-endpoint-ip-address](#) を使用して新しい IP アドレスを関連付ける必要があります。その後、元の IP アドレスの 1 つをエンドポイントから関連付け解除できます。

```

aws route53resolver disassociate-resolver-endpoint-ip-address \
  --resolver-endpoint-id rslvr-in-f9ab8a03f1example \
  --ip-address="SubnetId=subnet-12d8a459,Ip=172.31.40.121"

```

出力:

```

{
  "ResolverEndpoint": {
    "Id": "rslvr-in-f9ab8a03f1example",
    "CreatorRequestId": "2020-01-01-18:47",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/rslvr-in-f9ab8a03f1example",
    "Name": "my-inbound-endpoint",
  }
}

```

```

    "SecurityGroupIds": [
      "sg-f62bexam"
    ],
    "Direction": "INBOUND",
    "IpAddressCount": 3,
    "HostVPCId": "vpc-304bexam",
    "Status": "UPDATING",
    "StatusMessage": "Updating the Resolver Endpoint",
    "CreationTime": "2020-01-01T23:02:29.583Z",
    "ModificationTime": "2020-01-05T23:02:29.583Z"
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateResolverEndpointIpAddress](#)」の「」を参照してください。

disassociate-resolver-rule

次のコード例は、disassociate-resolver-rule を使用する方法を示しています。

AWS CLI

Amazon から Resolver ルールの関連付けを解除するには VPC

次のdisassociate-resolver-rule例では、指定されたリゾルバールールと指定された の間の関連付けを削除しますVPC。以下の状況VPCでは、 からルールの関連付けを解除できます。

この で発生するDNSクエリの場合VPC、ルールで指定されたドメイン名のネットワークへのクエリの転送を Resolver で停止する必要があります。転送ルールを削除します。ルールが現在 1 つ以上の に関連付けられている場合はVPCs、ルールを削除するVPCs前に、すべての からルールの関連付けを解除する必要があります。

```

aws route53resolver disassociate-resolver-rule \
  --resolver-rule-id rslvr-rr-4955cb98ceexample \
  --vpc-id vpc-304bexam

```

出力:

```

{
  "ResolverRuleAssociation": {
    "Id": "rslvr-rrassoc-322f4e8b9cexample",

```

```
"ResolverRuleId": "rslvr-rr-4955cb98ceexample",
  "Name": "my-resolver-rule-association",
  "VPCId": "vpc-304bexam",
  "Status": "DELETING",
  "StatusMessage": "[Trace id: 1-5dc5ffa2-a26c38004c1f94006example] Deleting
Association"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateResolverRule](#)」の「」を参照してください。

get-firewall-config

次の例は、get-firewall-config を使用する方法を説明しています。

AWS CLI

のファイアウォール設定を取得するには VPC

次のget-firewall-config例では、指定された のDNSファイアウォール動作を取得します VPC。

```
aws route53resolver get-firewall-config \
  --resource-id vpc-31e9222
```

出力:

```
{
  "FirewallConfig": {
    "Id": "rslvr-fc-86016850cexample",
    "ResourceId": "vpc-31e9222",
    "OwnerId": "123456789012",
    "FirewallFailOpen": "DISABLED"
  }
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「ファイアウォールVPCの設定」](#)を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFirewallConfig](#)」の「」を参照してください。

get-firewall-domain-list

次の例は、get-firewall-domain-list を使用する方法を説明しています。

AWS CLI

Route 53 Resolver DNS Firewall ドメインリストを取得するには

次のget-firewall-domain-list例では、指定した ID を持つドメインリストを取得します。

```
aws route53resolver get-firewall-domain-list \  
  --firewall-domain-list-id rslvr-fdl-42b60677cexample
```

出力:

```
{  
  "FirewallDomainList": {  
    "Id": "rslvr-fdl-9e956e9ffexample",  
    "Arn": "arn:aws:route53resolver:us-west-2:123457689012:firewall-domain-list/  
rslvr-fdl-42b60677cexample",  
    "Name": "test",  
    "DomainCount": 0,  
    "Status": "COMPLETE",  
    "StatusMessage": "Created Firewall Domain List",  
    "CreatorRequestId": "my-request-id",  
    "CreationTime": "2021-05-25T15:55:51.115365Z",  
    "ModificationTime": "2021-05-25T15:55:51.115365Z"  
  }  
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の「[独自のドメインリストの管理](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFirewallDomainList](#)」の「」を参照してください。

get-firewall-rule-group-association

次の例は、get-firewall-rule-group-association を使用する方法を説明しています。

AWS CLI

ファイアウォールルールグループの関連付けを取得するには

次のget-firewall-rule-group-association例では、ファイアウォールルールグループの関連付けを取得します。

```
aws route53resolver get-firewall-rule-group-association \  
--firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

出力:

```
{  
  "FirewallRuleGroupAssociation": {  
    "Id": "rslvr-frgassoc-57e8873d7example",  
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-  
association/rslvr-frgassoc-57e8873d7example",  
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",  
    "VpcId": "vpc-31e92222",  
    "Name": "test-association",  
    "Priority": 101,  
    "MutationProtection": "DISABLED",  
    "Status": "COMPLETE",  
    "StatusMessage": "Finished rule group association update",  
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",  
    "CreationTime": "2021-05-25T21:47:48.755768Z",  
    "ModificationTime": "2021-05-25T21:47:48.755768Z"  
  }  
}
```

詳細については、[「Amazon Route 53 デベロッパーガイド」の「VPCと Route 53 Resolver DNS Firewall ルールグループ間の関連付けの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFirewallRuleGroupAssociation](#)」の「」を参照してください。

get-firewall-rule-group-policy

次のコード例は、get-firewall-rule-group-policy を使用する方法を示しています。

AWS CLI

ポリシーを取得する AWS IAMには

次のget-firewall-rule-group-policy例では、指定されたルールグループを共有するための AWS Identity and Access Management (AWS IAM) ポリシーを取得します。

```
aws route53resolver get-firewall-rule-group-policy \  
  --arn arn:aws:route53resolver:us-west-2:AWS_ACCOUNT_ID:firewall-rule-group/  
rslvr-frg-47f93271fexample
```

出力:

```
{  
  "FirewallRuleGroupPolicy": "{\"Version\":\"2012-10-17\",  
  \"Statement\": [{\"Sid\":\"test\", \"Effect\":\"Allow\", \"Principal\": {\"AWS\": \"arn:aws:iam::AWS_ACCOUNT_ID:root\"}, \"Action\": [\"route53resolver:GetFirewallRuleGroup\", \"route53resolver:ListFirewallRuleGroups\"], \"Resource\": \"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-group/rslvr-frg-47f93271fexample\"}]}"  
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「ファイアウォールでのルールグループとルールの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[GetFirewallRuleGroupPolicy](#)」の「」を参照してください。

get-firewall-rule-group

次のコード例は、get-firewall-rule-group を使用する方法を示しています。

AWS CLI

Firewall ルールグループを取得するには

次のget-firewall-rule-group例では、指定した ID を持つ DNS Firewall ルールグループに関する情報を取得します。

```
aws route53resolver get-firewall-rule-group \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

出力:

```
{  
  "FirewallRuleGroup": {  
    "Id": "rslvr-frg-47f93271fexample",
```



```
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group/
rslvr-frg-47f93271fexample",
    "Name": "test",
    "RuleCount": 0,
    "Status": "COMPLETE",
    "StatusMessage": "Created Firewall Rule Group",
    "OwnerId": "123456789012",
    "CreatorRequestId": "my-request-id",
    "ShareStatus": "NOT_SHARED",
    "CreationTime": "2021-05-25T18:59:26.490017Z",
    "ModificationTime": "2021-05-25T18:59:26.490017Z"
  }
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の[DNS「ファイアウォールでのルールグループとルールの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI「コマンドリファレンス[GetFirewallRuleGroup](#)」の「」を参照してください。

get-resolver-endpoint

次のコード例は、get-resolver-endpoint を使用する方法を示しています。

AWS CLI

Resolver エンドポイントに関する情報を取得するには

次のget-resolver-endpoint例では、アウトバウンドで指定されたエンドポイントの詳細を表示します。該当するエンドポイント ID を指定することで、インバウンドエンドポイントとアウトバウンドエンドポイントget-resolver-endpointの両方に 使用できます。

```
aws route53resolver get-resolver-endpoint \
  --resolver-endpoint-id rslvr-out-d5e5920e37example
```

出力:

```
{
  "ResolverEndpoint": {
    "Id": "rslvr-out-d5e5920e37example",
    "CreatorRequestId": "2020-01-01-18:47",
```

```
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-out-d5e5920e37example",
    "Name": "my-outbound-endpoint",
    "SecurityGroupIds": [
      "sg-05cd7b25d6example"
    ],
    "Direction": "OUTBOUND",
    "IpAddressCount": 2,
    "HostVPCId": "vpc-304bexam",
    "Status": "OPERATIONAL",
    "StatusMessage": "This Resolver Endpoint is operational.",
    "CreationTime": "2020-01-01T23:50:50.979Z",
    "ModificationTime": "2020-01-02T23:50:50.979Z"
  }
}
```

詳細については、Amazon Route 53 デベロッパーガイドの「[インバウンドエンドポイントを作成または編集するときに指定する値](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResolverEndpoint](#)」の「」を参照してください。

get-resolver-rule-association

次のコード例は、get-resolver-rule-association を使用する方法を示しています。

AWS CLI

Resolver ルールと の関連付けに関する情報を取得するには VPC

次のget-resolver-rule-association例では、指定されたリゾルバールールと の関連付けの詳細を表示しますVPC。リゾルバールールと を関連付けるには、 VPCを使用します[associate-resolver-rule](#)。

```
aws route53resolver get-resolver-rule-association \
  --resolver-rule-association-id rslvr-rrassoc-d61cbb2c8bexample
```

出力:

```
{
  "ResolverRuleAssociation": {
    "Id": "rslvr-rrassoc-d61cbb2c8bexample",
```

```
    "ResolverRuleId": "rslvr-rr-42b60677c0example",
    "Name": "my-resolver-rule-association",
    "VPCId": "vpc-304bexam",
    "Status": "COMPLETE",
    "StatusMessage": ""
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResolverRuleAssociation](#)」の「」を参照してください。

get-resolver-rule

次のコード例は、get-resolver-rule を使用する方法を示しています。

AWS CLI

リゾルバールールに関する情報を取得するには

次のget-resolver-rule例では、ルールがDNSクエリを転送するドメイン名や、ルールが関連付けられているアウトバウンドリゾルバーエンドポイントの ID など、指定されたリゾルバールールの詳細を表示します。

```
aws route53resolver get-resolver-rule \
  --resolver-rule-id rslvr-rr-42b60677c0example
```

出力:

```
{
  "ResolverRule": {
    "Id": "rslvr-rr-42b60677c0example",
    "CreatorRequestId": "2020-01-01-18:47",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-rr-42b60677c0example",
    "DomainName": "example.com.",
    "Status": "COMPLETE",
    "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example] Successfully created Resolver Rule.",
    "RuleType": "FORWARD",
    "Name": "my-rule",
    "TargetIps": [
```

```
{
  "Ip": "192.0.2.45",
  "Port": 53
},
"ResolverEndpointId": "rslvr-out-d5e5920e37example",
"OwnerId": "111122223333",
"ShareStatus": "NOT_SHARED"
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の「[ルールを作成または編集するときに指定する値](#)」を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResolverRule](#)」の「」を参照してください。

import-firewall-domains

次のコード例は、import-firewall-domains を使用する方法を示しています。

AWS CLI

ドメインをドメインリストにインポートするには

次のimport-firewall-domains例では、一連のドメインをファイルから指定した DNS Firewall ドメインリストにインポートします。

```
aws route53resolver import-firewall-domains \
  --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample \
  --operation REPLACE \
  --domain-file-url s3://PATH/TO/YOUR/FILE
```

出力:

```
{
  "Id": "rslvr-fdl-d61cbb2cbexample",
  "Name": "test",
  "Status": "IMPORTING",
  "StatusMessage": "Importing domains from provided file."
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の[「独自のドメインリストの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportFirewallDomains](#)」の「」を参照してください。

list-firewall-configs

次の例は、list-firewall-configs を使用する方法を説明しています。

AWS CLI

ファイアウォール設定を一覧表示するには

次のlist-firewall-configs例では、DNSファイアウォール設定を一覧表示します。

```
aws route53resolver list-firewall-configs
```

出力:

```
{
  "FirewallConfigs": [
    {
      "Id": "rslvr-fc-86016850cexample",
      "ResourceId": "vpc-31e92222",
      "OwnerId": "123456789012",
      "FirewallFailOpen": "DISABLED"
    }
  ]
}
```

詳細については、「Amazon Route 53デベロッパーガイド」の[DNS「ファイアウォールVPCの設定」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFirewallConfigs](#)」の「」を参照してください。

list-firewall-domain-lists

次のコード例は、list-firewall-domain-lists を使用する方法を示しています。

AWS CLI

Route 53 Resolver DNS Firewall のすべてのドメインリストを一覧表示するには

次の`list-firewall-domain-lists`例では、すべてのドメインリストを一覧表示します。

```
aws route53resolver list-firewall-domain-lists
```

出力:

```
{
  "FirewallDomainLists": [
    {
      "Id": "rslvr-fdl-2c46f2ecfexample",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/rslvr-fdl-2c46f2ecfexample",
      "Name": "AWSManagedDomainsMalwareDomainList",
      "CreatorRequestId": "AWSManagedDomainsMalwareDomainList",
      "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
    },
    {
      "Id": "rslvr-fdl-aa970e9e1example",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-domain-list/rslvr-fdl-aa970e9e1example",
      "Name": "AWSManagedDomainsBotnetCommandandControl",
      "CreatorRequestId": "AWSManagedDomainsBotnetCommandandControl",
      "ManagedOwnerName": "Route 53 Resolver DNS Firewall"
    },
    {
      "Id": "rslvr-fdl-42b60677cexample",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789111:firewall-domain-list/rslvr-fdl-42b60677cexample",
      "Name": "test",
      "CreatorRequestId": "my-request-id"
    }
  ]
}
```

詳細については、Amazon [Route 53 デベロッパーガイド](#)の「[Route 53 Resolver DNS Firewall ドメインリスト](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「[コマンドリファレンスListFirewallDomainLists](#)」の「」を参照してください。

list-firewall-domains

次の例は、list-firewall-domains を使用する方法を説明しています。

AWS CLI

ドメインリスト内のドメインを一覧表示するには

次のlist-firewall-domains例では、指定した DNS Firewall ドメインリスト内のドメインを一覧表示します。

```
aws route53resolver list-firewall-domains \  
  --firewall-domain-list-id rs1vr-fdl-d61cbb2cbexample
```

出力:

```
{  
  "Domains": [  
    "test1.com.",  
    "test2.com.",  
    "test3.com."  
  ]  
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の「[独自のドメインリストの管理](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFirewallDomains](#)」の「」を参照してください。

list-firewall-rule-group-associations

次の例は、list-firewall-rule-group-associations を使用する方法を説明しています。

AWS CLI

DNS Firewall ルールグループの関連付けを一覧表示するには

次のlist-firewall-rule-group-associations例では、Amazon との DNS Firewall ルールグループの関連付けを一覧表示しますVPCs。

```
aws route53resolver list-firewall-rule-group-associations
```

出力:

```
{
  "FirewallRuleGroupAssociations": [
    {
      "Id": "rslvr-frgassoc-57e8873d7example",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-association/rslvr-frgassoc-57e8873d7example",
      "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
      "VpcId": "vpc-31e92222",
      "Name": "test-association",
      "Priority": 101,
      "MutationProtection": "DISABLED",
      "Status": "UPDATING",
      "StatusMessage": "Creating Firewall Rule Group Association",
      "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
      "CreationTime": "2021-05-25T21:47:48.755768Z",
      "ModificationTime": "2021-05-25T21:47:48.755768Z"
    }
  ]
}
```

詳細については、[「Amazon Route 53 デベロッパーガイド」の「VPCと Route 53 Resolver DNS Firewall ルールグループ間の関連付けの管理」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFirewallRuleGroupAssociations](#)」の「」を参照してください。

list-firewall-rule-groups

次のコード例は、list-firewall-rule-groups を使用する方法を示しています。

AWS CLI

Firewall ルールグループのリストを取得するには

次のlist-firewall-rule-groups例では、DNSファイアウォールルールグループを一覧表示します。

```
aws route53resolver list-firewall-rule-groups
```

出力:


```
{
  "FirewallRuleGroups": [
    {
      "Id": "rslvr-frg-47f93271fexample",
      "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-
group/rslvr-frg-47f93271fexample",
      "Name": "test",
      "OwnerId": "123456789012",
      "CreatorRequestId": "my-request-id",
      "ShareStatus": "NOT_SHARED"
    }
  ]
}
```

詳細については、Amazon Route 53デベロッパーガイド」の[DNS「Firewallでのルールグループとルールの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListFirewallRuleGroups](#)」の「」を参照してください。

list-firewall-rules

次のコード例は、list-firewall-rules を使用する方法を示しています。

AWS CLI

ファイアウォールルールを一覧表示するには

次のlist-firewall-rules例では、DNSファイアウォールルールグループ内のすべてのファイアウォールルールを一覧表示します。

```
aws route53resolver list-firewall-rules \
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

出力:

```
{
  "FirewallRules": [
    {
      "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",
      "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",
      "Name": "allow-rule",

```

```
        "Priority": 101,
        "Action": "ALLOW",
        "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",
        "CreationTime": "2021-05-25T21:44:00.346093Z",
        "ModificationTime": "2021-05-25T21:44:00.346093Z"
    }
]
}
```

詳細については、Amazon Route 53デベロッパーガイド」の[DNS「Firewallでのルールグループとルールの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[ListFirewallRules](#)」の「」を参照してください。

list-resolver-endpoint-ip-addresses

次の例は、list-resolver-endpoint-ip-addresses を使用する方法を説明しています。

AWS CLI

指定されたインバウンドエンドポイントまたはアウトバウンドエンドポイントの IP アドレスを一覧表示するには

次のlist-resolver-endpoint-ip-addresses例は、インバウンドエンドポイントに関連付けられている IP アドレスに関する情報を示していますrs1vr-in-f9ab8a03f1example。該当するエンドポイント ID を指定することで、アウトバウンドエンドポイントlist-resolver-endpoint-ip-addressesに を使用することもできます。

```
aws route53resolver list-resolver-endpoint-ip-addresses \
  --resolver-endpoint-id rs1vr-in-f9ab8a03f1example
```

出力:

```
{
  "MaxResults": 10,
  "IpAddresses": [
    {
      "IpId": "rni-1de60cdbfeexample",
      "SubnetId": "subnet-ba47exam",
      "Ip": "192.0.2.44",
```

```
    "Status": "ATTACHED",
    "StatusMessage": "This IP address is operational.",
    "CreationTime": "2020-01-03T23:02:29.587Z",
    "ModificationTime": "2020-01-03T23:03:05.555Z"
  },
  {
    "IpId": "rni-aac7085e38example",
    "SubnetId": "subnet-12d8exam",
    "Ip": "192.0.2.45",
    "Status": "ATTACHED",
    "StatusMessage": "This IP address is operational.",
    "CreationTime": "2020-01-03T23:02:29.593Z",
    "ModificationTime": "2020-01-03T23:02:55.060Z"
  }
]
```

出力の値の詳細については、Amazon Route 53 デベロッパーガイドの「[インバウンドエンドポイントを作成または編集するときに指定する値](#)」と「[アウトバウンドエンドポイントを作成または編集するときに指定する値](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResolverEndpointIpAddresses](#)」の「」を参照してください。

list-resolver-endpoints

次の例は、list-resolver-endpoints を使用する方法を説明しています。

AWS CLI

AWS リージョンの Resolver エンドポイントを一覧表示するには

次のlist-resolver-endpoints例では、現在のアカウントに存在するインバウンドおよびアウトバウンドの Resolver エンドポイントを一覧表示します。

```
aws route53resolver list-resolver-endpoints
```

出力:

```
{
  "MaxResults": 10,
  "ResolverEndpoints": [
```

```
{
  "Id": "rslvr-in-497098ad59example",
  "CreatorRequestId": "2020-01-01-18:47",
  "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-in-497098ad59example",
  "Name": "my-inbound-endpoint",
  "SecurityGroupIds": [
    "sg-05cd7b25d6example"
  ],
  "Direction": "INBOUND",
  "IpAddressCount": 2,
  "HostVPCId": "vpc-304bexam",
  "Status": "OPERATIONAL",
  "StatusMessage": "This Resolver Endpoint is operational.",
  "CreationTime": "2020-01-01T23:25:45.538Z",
  "ModificationTime": "2020-01-01T23:25:45.538Z"
},
{
  "Id": "rslvr-out-d5e5920e37example",
  "CreatorRequestId": "2020-01-01-18:48",
  "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-
endpoint/rslvr-out-d5e5920e37example",
  "Name": "my-outbound-endpoint",
  "SecurityGroupIds": [
    "sg-05cd7b25d6example"
  ],
  "Direction": "OUTBOUND",
  "IpAddressCount": 2,
  "HostVPCId": "vpc-304bexam",
  "Status": "OPERATIONAL",
  "StatusMessage": "This Resolver Endpoint is operational.",
  "CreationTime": "2020-01-01T23:50:50.979Z",
  "ModificationTime": "2020-01-01T23:50:50.979Z"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResolverEndpoints](#)」の「」を参照してください。

list-resolver-rule-associations

次のコード例は、list-resolver-rule-associations を使用する方法を示しています。

AWS CLI

Resolver ルールと の関連付けを一覧表示するには VPCs

次のlist-resolver-rule-associations例では、リゾルバールールと現在の AWS アカウントの の関連付けを一覧表示VPCsします。

```
aws route53resolver list-resolver-rule-associations
```

出力:

```
{
  "MaxResults": 30,
  "ResolverRuleAssociations": [
    {
      "Id": "rslvr-autodefined-assoc-vpc-304bexam-internet-resolver",
      "ResolverRuleId": "rslvr-autodefined-rr-internet-resolver",
      "Name": "System Rule Association",
      "VPCId": "vpc-304bexam",
      "Status": "COMPLETE",
      "StatusMessage": ""
    },
    {
      "Id": "rslvr-rrassoc-d61cbb2c8bexample",
      "ResolverRuleId": "rslvr-rr-42b60677c0example",
      "Name": "my-resolver-rule-association",
      "VPCId": "vpc-304bexam",
      "Status": "COMPLETE",
      "StatusMessage": ""
    }
  ]
}
```

詳細については、「Amazon [Route 53 デベロッパーガイド](#)」の「[Route 53 Resolver が からネットワークVPCsにDNSクエリを転送する方法](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「 [コマンドリファレンスListResolverRuleAssociations](#) 」の「」を参照してください。

list-resolver-rules

次のコード例は、list-resolver-rules を使用する方法を示しています。

AWS CLI

リゾルバールールを一覧表示するには

次のlist-resolver-rules例では、現在の AWS アカウントのすべてのリゾルバールールを一覧表示します。

```
aws route53resolver list-resolver-rules
```

出力:

```
{
  "MaxResults": 30,
  "ResolverRules": [
    {
      "Id": "rslvr-autodefined-rr-internet-resolver",
      "CreatorRequestId": "",
      "Arn": "arn:aws:route53resolver:us-west-2::autodefined-rule/rslvr-autodefined-rr-internet-resolver",
      "DomainName": ".",
      "Status": "COMPLETE",
      "RuleType": "RECURSIVE",
      "Name": "Internet Resolver",
      "OwnerId": "Route 53 Resolver",
      "ShareStatus": "NOT_SHARED"
    },
    {
      "Id": "rslvr-rr-42b60677c0example",
      "CreatorRequestId": "2020-01-01-18:47",
      "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-rr-42b60677c0bc4e299",
      "DomainName": "example.com.",
      "Status": "COMPLETE",
      "StatusMessage": "[Trace id: 1-5dc4b177-ff1d9d001a0f80005example] Successfully created Resolver Rule.",
      "RuleType": "FORWARD",
      "Name": "my-rule",
      "TargetIps": [
        {
          "Ip": "192.0.2.45",
          "Port": 53
        }
      ]
    }
  ],
}
```

```
        "ResolverEndpointId": "rslvr-out-d5e5920e37example",
        "OwnerId": "111122223333",
        "ShareStatus": "NOT_SHARED"
    }
]
}
```

詳細については、「Amazon [Route 53 デベロッパーガイド](#)」の「[Route 53 Resolver が からネットワークVPCsにDNSクエリを転送する方法](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResolverRules](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

Resolver リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定されたリゾルバールールに割り当てられたタグを一覧表示します。

```
aws route53resolver list-tags-for-resource \
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
  rslvr-rr-42b60677c0example"
```

出力:

```
{
  "Tags": [
    {
      "Key": "my-key-1",
      "Value": "my-value-1"
    },
    {
      "Key": "my-key-2",
      "Value": "my-value-2"
    }
  ]
}
```

コスト配分にタグを使用する方法については、AWS「[請求とコスト管理ユーザーガイド](#)」の「[コスト配分タグの使用](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスListTagsForResource](#)」の「」を参照してください。

put-firewall-rule-group-policy

次の例は、put-firewall-rule-group-policy を使用する方法を説明しています。

AWS CLI

Firewall ルールグループポリシーを共有するポリシーをアタッチ AWS IAMするには

次のput-firewall-rule-group-policy例では、ルールグループを共有するための AWS Identity and Access Management (AWS IAM) ポリシーをアタッチします。

```
aws route53resolver put-firewall-rule-group-policy \
  --firewall-rule-group-policy "{\"Version\":\"2012-10-17\",
  \"Statement\": [{\"Sid\":\"test\", \"Effect\":\"Allow\", \"Principal\": {\"AWS\": \"arn:aws:iam::AWS_ACCOUNT_ID:root\"}, \"Action\": [\"route53resolver:GetFirewallRuleGroup\", \"route53resolver:ListFirewallRuleGroups\"], \"Resource\": \"arn:aws:route53resolver:us-east-1:AWS_ACCOUNT_ID:firewall-rule-group/rs1vr-frg-47f93271fexample\"}]}"
```

出力:

```
{
  "ReturnValue": true
}
```

詳細については、Amazon Route 53デベロッパーガイド」の[DNS「Firewallでのルールグループとルールの管理」](#)を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスPutFirewallRuleGroupPolicy](#)」の「」を参照してください。

put-resolver-rule-policy

次のコード例は、put-resolver-rule-policy を使用する方法を示しています。

AWS CLI

Resolver ルールを別の AWS アカウントと共有するには

次のput-resolver-rule-policy例では、別の AWS アカウントと共有したい Resolver ルール、ルールを共有する アカウント、および アカウントがルールに対して実行できるルール関連のオペレーションを指定します。

注 このコマンドは、ルールを作成したのと同じアカウントの認証情報を使用して実行する必要があります。

```
aws route53resolver put-resolver-rule-policy \
  --region us-east-1 \
  --arn "arn:aws:route53resolver:us-east-1:111122223333:resolver-rule/rslvr-rr-42b60677c0example" \
  --resolver-rule-policy "{\"Version\": \"2012-10-17\", \
    \"Statement\": [ { \
      \"Effect\" : \"Allow\", \
      \"Principal\" : {\"AWS\" : \"444455556666\" }, \
      \"Action\" : [ \
        \"route53resolver:GetResolverRule\", \
        \"route53resolver:AssociateResolverRule\", \
        \"route53resolver:DisassociateResolverRule\", \
        \"route53resolver:ListResolverRules\", \
        \"route53resolver:ListResolverRuleAssociations\" ], \
      \"Resource\" : [ \"arn:aws:route53resolver:us-east-1:111122223333:resolver-rule/rslvr-rr-42b60677c0example\" ] } ] }"
```

出力:

```
{
  "ReturnValue": true
}
```

を実行したらput-resolver-rule-policy、次の 2 つの Resource Access Manager (RAM) コマンドを実行できます。ルールを共有するアカウントを使用する必要があります。

get-resource-share-invitations は値 を返しますresourceShareInvitationArn。共有ルールを使用するための招待を受け入れるには、この値が必要です。accept-resource-share-invitationは共有ルールを使用するための招待を受け入れます。

詳細については、次のドキュメントを参照してください。

Amazon Route 53 デベロッパーガイドの[get-resource-share-invitationsaccept-resource-share-invitations](#)転送ルールを他の AWS アカウントと共有し、共有ルールを使用する

- API 詳細については、AWS CLI 「コマンドリファレンス[PutResolverRulePolicy](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

タグを Resolver リソースに関連付けるには

次のtag-resource例では、2 つのタグキーと値のペアを指定されたリゾルバールールに関連付けます。

```
aws route53resolver tag-resource \  
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/  
  rslvr-rr-42b60677c0example" \  
  --tags "Key=my-key-1,Value=my-value-1" "Key=my-key-2,Value=my-value-2"
```

このコマンドでは何も出力されません。

コスト配分にタグを使用する方法については、AWS 「請求とコスト管理ユーザーガイド」の「コスト[配分タグの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

Resolver リソースからタグを削除するには

次のuntag-resource例では、指定されたリゾルバールールから 2 つのタグを削除します。

```
aws route53resolver untag-resource \  
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/  
  rslvr-rr-42b60677c0example" \  
  --tags "Key=my-key-1,Value=my-value-1" "Key=my-key-2,Value=my-value-2"
```

```
--tag-keys my-key-1 my-key-2
```

このコマンドでは何も出力されません。タグが削除されたことを確認するには、[を使用できます](#) `list-tags-for-resource`。

コスト配分にタグを使用する方法については、AWS「[請求とコスト管理ユーザーガイド](#)」の「[コスト配分タグの使用](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスUntagResource](#)」の「」を参照してください。

update-firewall-config

次のコード例は、`update-firewall-config` を使用する方法を示しています。

AWS CLI

ファイアウォール設定を更新するには

次の`update-firewall-config`例では、DNSFirewall 設定を更新します。

```
aws route53resolver update-firewall-config \  
  --resource-id vpc-31e92222 \  
  --firewall-fail-open DISABLED
```

出力:

```
{  
  "FirewallConfig": {  
    "Id": "rslvr-fc-86016850cexample",  
    "ResourceId": "vpc-31e92222",  
    "OwnerId": "123456789012",  
    "FirewallFailOpen": "DISABLED"  
  }  
}
```

詳細については、「[Amazon Route 53 デベロッパーガイド](#)」の[DNS「ファイアウォールVPC設定](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI「[コマンドリファレンスUpdateFirewallConfig](#)」の「」を参照してください。

update-firewall-domains

次の例は、update-firewall-domains を使用する方法を説明しています。

AWS CLI

ドメインリストを更新するには

次のupdate-firewall-domains例では、指定した ID を使用してドメインをドメインリストに追加します。

```
aws route53resolver update-firewall-domains \  
  --firewall-domain-list-id rslvr-fdl-42b60677cexampleb \  
  --operation ADD \  
  --domains test1.com test2.com test3.com
```

出力:

```
{  
  "Id": "rslvr-fdl-42b60677cexample",  
  "Name": "test",  
  "Status": "UPDATING",  
  "StatusMessage": "Updating the Firewall Domain List"  
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の「[独自のドメインリストの管理](#)」を参照してください。 Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFirewallDomains](#)」の「」を参照してください。

update-firewall-rule-group-association

次の例は、update-firewall-rule-group-association を使用する方法を説明しています。

AWS CLI

ファイアウォールルールグループの関連付けを更新するには

次のupdate-firewall-rule-group-association例では、ファイアウォールルールグループの関連付けを更新します。

```
aws route53resolver update-firewall-rule-group-association \  
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example \  
  --priority 103
```

出力:

```
{  
  "FirewallRuleGroupAssociation": {  
    "Id": "rslvr-frgassoc-57e8873d7example",  
    "Arn": "arn:aws:route53resolver:us-west-2:123456789012:firewall-rule-group-  
association/rslvr-frgassoc-57e8873d7example",  
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",  
    "VpcId": "vpc-31e92222",  
    "Name": "test-association",  
    "Priority": 103,  
    "MutationProtection": "DISABLED",  
    "Status": "UPDATING",  
    "StatusMessage": "Updating the Firewall Rule Group Association Attributes",  
    "CreatorRequestId": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",  
    "CreationTime": "2021-05-25T21:47:48.755768Z",  
    "ModificationTime": "2021-05-25T21:50:09.272569Z"  
  }  
}
```

詳細については、Amazon Route [53 デベロッパーガイドの「VPCと Route 53 Resolver DNS Firewall ルールグループ間の関連付けの管理Route 53」](#)を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFirewallRuleGroupAssociation](#)」の「」を参照してください。

update-firewall-rule

次のコード例は、update-firewall-rule を使用する方法を示しています。

AWS CLI

ファイアウォールルールを更新するには

次のupdate-firewall-rule例では、指定したパラメータを使用してファイアウォールルールを更新します。

```
aws route53resolver update-firewall-rule \  

```

```
--firewall-rule-group-id rslvr-frg-47f93271fexample \  
--firewall-domain-list-id rslvr-fdl-9e956e9ffexample \  
--priority 102
```

出力:

```
{  
  "FirewallRule": {  
    "FirewallRuleGroupId": "rslvr-frg-47f93271fexample",  
    "FirewallDomainListId": "rslvr-fdl-9e956e9ffexample",  
    "Name": "allow-rule",  
    "Priority": 102,  
    "Action": "ALLOW",  
    "CreatorRequestId": "d81e3fb7-020b-415e-939f-EXAMPLE11111",  
    "CreationTime": "2021-05-25T21:44:00.346093Z",  
    "ModificationTime": "2021-05-25T21:45:59.611600Z"  
  }  
}
```

詳細については、Amazon Route 53デベロッパーガイド」の[DNS「Firewallでのルールグループとルールの管理」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateFirewallRule](#)」の「」を参照してください。

update-resolver-endpoint

次の例は、update-resolver-endpoint を使用する方法を説明しています。

AWS CLI

Resolver エンドポイントの名前を更新するには

次のupdate-resolver-endpoint例では、Resolver エンドポイントの名前を更新します。他の値の更新はサポートされていません。

```
aws route53resolver update-resolver-endpoint \  
--resolver-endpoint-id rslvr-in-b5d45e32bdc445f09 \  
--name my-renamed-inbound-endpoint
```

出力:

```
{
  "ResolverEndpoint": {
    "Id": "rslvr-in-b5d45e32bdexample",
    "CreatorRequestId": "2020-01-02-18:48",
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-endpoint/
rslvr-in-b5d45e32bdexample",
    "Name": "my-renamed-inbound-endpoint",
    "SecurityGroupIds": [
      "sg-f62bexam"
    ],
    "Direction": "INBOUND",
    "IpAddressCount": 2,
    "HostVPCId": "vpc-304bexam",
    "Status": "OPERATIONAL",
    "StatusMessage": "This Resolver Endpoint is operational.",
    "CreationTime": "2020-01-01T18:33:59.265Z",
    "ModificationTime": "2020-01-08T18:33:59.265Z"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResolverEndpoint](#)」の「」を参照してください。

update-resolver-rule

次のコード例は、update-resolver-rule を使用する方法を示しています。

AWS CLI

例 1: 設定を更新するには Resolver エンドポイント

次のupdate-resolver-rule例では、ルールの名前、DNSクエリが転送されるオンプレミスネットワーク上の IP アドレス、およびクエリをネットワークに転送するために使用するアウトバウンド Resolver エンドポイントの ID を更新します。

の既存の値はTargetIps上書きされるため、更新後にルールに含めるすべての IP アドレスを指定する必要があります。

```
aws route53resolver update-resolver-rule \  
  --resolver-rule-id rslvr-rr-1247fa64f3example \  
  --target-ips 10.0.0.0/24 \  
  --resolver-endpoint-id res-rr-123456789012 \  
  --direction INBOUND \  
  --ip-address-count 2 \  
  --host-vpc-id vpc-12345678 \  
  --status OPERATIONAL \  
  --status-message This Resolver Endpoint is operational. \  
  --creation-time 2020-01-01T18:33:59.265Z \  
  --modification-time 2020-01-08T18:33:59.265Z
```

```
--config Name="my-2nd-rule",TargetIps=[{Ip=192.0.2.45,Port=53},  
{Ip=192.0.2.46,Port=53}],ResolverEndpointId=rslvr-out-7b89ed0d25example
```

出力:

```
{  
  "ResolverRule": {  
    "Id": "rslvr-rr-1247fa64f3example",  
    "CreatorRequestId": "2020-01-02-18:47",  
    "Arn": "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-  
rr-1247fa64f3example",  
    "DomainName": "www.example.com.",  
    "Status": "COMPLETE",  
    "StatusMessage": "[Trace id: 1-5dcc90b9-8a8ee860aba1ebd89example]  
Successfully updated Resolver Rule.",  
    "RuleType": "FORWARD",  
    "Name": "my-2nd-rule",  
    "TargetIps": [  
      {  
        "Ip": "192.0.2.45",  
        "Port": 53  
      },  
      {  
        "Ip": "192.0.2.46",  
        "Port": 53  
      }  
    ],  
    "ResolverEndpointId": "rslvr-out-7b89ed0d25example",  
    "OwnerId": "111122223333",  
    "ShareStatus": "NOT_SHARED"  
  }  
}
```

例 2: 「config」設定用のファイルを使用して設定を更新するには

または、config 設定をJSONファイルに追加し、 を呼び出すときにそのファイルを指定することもできますupdate-resolver-rule。

```
aws route53resolver update-resolver-rule \  
--resolver-rule-id rslvr-rr-1247fa64f3example \  
--config file://c:\temp\update-resolver-rule.json
```


update-resolver-rule.json の内容。

```
{
  "Name": "my-2nd-rule",
  "TargetIps": [
    {
      "Ip": "192.0.2.45",
      "Port": 53
    },
    {
      "Ip": "192.0.2.46",
      "Port": 53
    }
  ],
  "ResolverEndpointId": "rslvr-out-7b89ed0d25example"
}
```

詳細については、「Amazon Route 53 デベロッパーガイド」の「[ルールを作成または編集するときに指定する値](#)」を参照してください。Amazon Route 53

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResolverRule](#)」の「」を参照してください。

を使用した Amazon S3 の例 AWS CLI

次のコード例は、Amazon S3 AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

abort-multipart-upload

次の例は、abort-multipart-upload を使用する方法を説明しています。

AWS CLI

指定したマルチパートアップロードを中止するには

次の abort-multipart-upload コマンドは、バケット my-bucket 内のキー multipart/01 のマルチパートアップロードを中止します。

```
aws s3api abort-multipart-upload \  
  --bucket my-bucket \  
  --key multipart/01 \  
  --upload-  
id dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

このコマンドに必要なアップロード ID create-multipart-upload はによって出力され、list-multipart-uploads で取得することもできます。

- API 詳細については、AWS CLI 「コマンドリファレンス [AbortMultipartUpload](#)」の「」を参照してください。

complete-multipart-upload

次の例は、complete-multipart-upload を使用する方法を説明しています。

AWS CLI

次のコマンドは、バケット my-bucket 内のキー multipart/01 のマルチパートアップロードを完了します。

```
aws s3api complete-multipart-upload --multipart-upload file://  
mpustruct --bucket my-bucket --key 'multipart/01' --upload-  
id dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

このコマンドに必要なアップロード ID create-multipart-upload はによって出力され、list-multipart-uploads で取得することもできます。

上記のコマンドのマルチパートアップロードオプションは、ファイル全体に再アSEMBルする必要があるマルチパートアップロードの部分を説明するJSON構造を取ります。この例では、file://プレフィックスを使用して、という名前のローカルフォルダ内のファイルからJSON構造をロードしますmpustruct。

mpustruct:

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

各パートのETag値は、upload-part コマンドを使用してパートをアップロードするたびに出力されます。また、を呼び出して取得list-partsすることも、各パートのMD5チェックサムを取得して計算することもできます。

出力:

```
{
  "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
  "Bucket": "my-bucket",
  "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
  "Key": "multipart/01"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CompleteMultipartUpload](#)」の「」を参照してください。

copy-object

次のコード例は、copy-object を使用する方法を示しています。

AWS CLI

次のコマンドは、bucket-1 から bucket-2 にオブジェクトをコピーします。

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --  
bucket bucket-2
```

出力:

```
{  
  "CopyObjectResult": {  
    "LastModified": "2015-11-10T01:07:25.000Z",  
    "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""  
  },  
  "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifM1kA"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyObject](#)」の「」を参照してください。

cp

次の例は、cp を使用する方法を説明しています。

AWS CLI

例 1: ローカルファイルを S3 にコピーする

次のcpコマンドは、指定されたバケットとキーに 1 つのファイルをコピーします。

```
aws s3 cp test.txt s3://mybucket/test2.txt
```

出力:

```
upload: test.txt to s3://mybucket/test2.txt
```

例 2: ローカルファイルを有効期限付き S3 にコピーする

次のcpコマンドは、指定された 8601 タイムスタンプで期限切れになる指定されたバケットとキーに 1 ISO つのファイルをコピーします。

```
aws s3 cp test.txt s3://mybucket/test2.txt \  
--expires 2014-10-01T20:30:00Z
```

出力:

```
upload: test.txt to s3://mybucket/test2.txt
```

例 3: S3 から S3 にファイルをコピーする

次のcpコマンドは、単一の s3 オブジェクトを指定されたバケットとキーにコピーします。

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt
```

出力:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

例 4: S3 オブジェクトをローカルファイルにコピーする

次のcpコマンドは、1 つのオブジェクトを指定されたファイルにローカルにコピーします。

```
aws s3 cp s3://mybucket/test.txt test2.txt
```

出力:

```
download: s3://mybucket/test.txt to test2.txt
```

例 5: あるバケットから別のバケットに S3 オブジェクトをコピーする

次のcpコマンドは、元の名前を保持しながら、単一のオブジェクトを指定されたバケットにコピーします。

```
aws s3 cp s3://mybucket/test.txt s3://mybucket2/
```

出力:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

例 6: S3 オブジェクトをローカルディレクトリに再帰的にコピーする

パラメータで渡されると `--recursive`、次の `cp` コマンドは、指定されたプレフィックスとバケットのすべてのオブジェクトを指定されたディレクトリに再帰的にコピーします。この例では、バケットにオブジェクト `test1.txt` と `mybucket` があります `test2.txt`。

```
aws s3 cp s3://mybucket . \
  --recursive
```

出力:

```
download: s3://mybucket/test1.txt to test1.txt
download: s3://mybucket/test2.txt to test2.txt
```

例 7: ローカルファイルを S3 に再帰的にコピーする

パラメータで渡されると `--recursive`、次の `cp` コマンドは、`--exclude` パラメータを使用して一部のファイルを除外しながら、指定されたディレクトリ内のすべてのファイルを指定されたバケットとプレフィックスに再帰的にコピーします。この例では、ディレクトリにファイル `test1.txt` と `myDir` があります `test2.jpg`。

```
aws s3 cp myDir s3://mybucket/ \
  --recursive \
  --exclude "*.jpg"
```

出力:

```
upload: myDir/test1.txt to s3://mybucket/test1.txt
```

例 8: S3 オブジェクトを別のバケットに再帰的にコピーする

パラメータで渡されると `--recursive`、次の `cp` コマンドは、`--exclude` パラメータを使用して一部のオブジェクトを除外しながら、指定されたバケット内のすべてのオブジェクトを再帰的に別のバケットにコピーします。この例では、バケット `mybucket` にはオブジェクト `test1.txt` とがあります `another/test1.txt`。

```
aws s3 cp s3://mybucket/ s3://mybucket2/ \
  --recursive \
  --exclude "another/*"
```

出力:

```
copy: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

--exclude と --include オプションを組み合わせて、パターンに一致するオブジェクトのみをコピーできます。ただし、他のすべてのオブジェクトはコピーできません。

```
aws s3 cp s3://mybucket/logs/ s3://mybucket2/logs/ \  
  --recursive \  
  --exclude "*" \  
  --include "*.log"
```

出力:

```
copy: s3://mybucket/logs/test/test.log to s3://mybucket2/logs/test/test.log  
copy: s3://mybucket/logs/test3.log to s3://mybucket2/logs/test3.log
```

例 9: S3 オブジェクトのコピー中にアクセスコントロールリスト (ACL) を設定する

次の cp コマンドは、 を に設定する際に、単一のオブジェクトを指定されたバケットとキーにコピー ACL します public-read-write。

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt \  
  --acl public-read-write
```

出力:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

--acl オプションを使用している場合は、関連する IAM ポリシーに "s3:PutObjectAcl" アクションが含まれていることを確認してください。

```
aws iam get-user-policy \  
  --user-name myuser \  
  --policy-name mypolicy
```

出力:

```
{
```

```

    "UserName": "myuser",
    "PolicyName": "mypolicy",
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl"
          ],
          "Resource": [
            "arn:aws:s3:::mybucket/*"
          ],
          "Effect": "Allow",
          "Sid": "Stmt1234567891234"
        }
      ]
    }
  }
}

```

例 10: S3 オブジェクトに対するアクセス許可の付与

次のcpコマンドは、によって識別されたすべてのユーザーに読み取りアクセスを付与する--grantsオプションの使用URIと、正規 ID によって識別された特定のユーザーに完全なコントロールを付与するオプションの使用を示しています。

```

aws s3 cp file.txt s3://mybucket/ --grants read=uri=http://acs.amazonaws.com/groups/global/AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be

```

出力:

```

upload: file.txt to s3://mybucket/file.txt

```

例 11: ローカルファイルストリームを S3 にアップロードする

PowerShell は、のエンコーディングを変更したり、パイプ入力CRLFにを追加したりすることがあります。

次のcpコマンドは、ローカルファイルストリームを標準入力から指定されたバケットとキーにアップロードします。


```
aws s3 cp - s3://mybucket/stream.txt
```

例 12: 50GB を超えるローカルファイルストリームを S3 にアップロードする

次のcpコマンドは、標準入力から指定されたバケットとキーに 51GB のローカルファイルストリームをアップロードします。--expected-size オプションを指定する必要があります。指定しない場合、デフォルトのパート制限である 10,000 に達するとアップロードが失敗する可能性があります。

```
aws s3 cp - s3://mybucket/stream.txt --expected-size 54760833024
```

例 13: S3 オブジェクトをローカルファイルストリームとしてダウンロードする

PowerShell は、 のエンコーディングを変更したり、パイプ出力またはリダイレクト出力CRLFに追加したりすることがあります。

次のcpコマンドは、標準出力へのストリームとして S3 オブジェクトをローカルにダウンロードします。ストリームとしてのダウンロードは現在、 --recursiveパラメータと互換性がありません。

```
aws s3 cp s3://mybucket/stream.txt -
```

例 14: S3 アクセスポイントへのアップロード

次のcpコマンドは、キー (mydoc.txt) のアクセスポイント () に 1 つのファイル (myaccesspoint) をアップロードしますmykey。

```
aws s3 cp mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

出力:

```
upload: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

例 15: S3 アクセスポイントからのダウンロード

次のcpコマンドは、アクセスポイント (mykey) からローカルファイル () に単一のオブジェクト (myaccesspoint) をダウンロードしますmydoc.txt。

```
aws s3 cp s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/  
mykey mydoc.txt
```

出力:

```
download: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey to  
mydoc.txt
```

- API 詳細については、「コマンドリファレンス」の「[Cp](#)」を参照してください。AWS CLI

create-bucket

次の例は、create-bucket を使用する方法を説明しています。

AWS CLI

例 1: バケットを作成するには

次の create-bucket の例は、my-bucket という名前のバケットを作成します。

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1
```

出力:

```
{  
  "Location": "/my-bucket"  
}
```

詳細については、「Amazon S3 ユーザーガイド」の「[バケットの作成](#)」を参照してください。

例 2: 所有者の強制を使用してバケットを作成するには

次の create-bucket の例は、S3 オブジェクトの所有権のバケット所有者の強制設定を使用して、my-bucket という名前のバケットを作成します。

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1 \  
  --ownership-control s3-bucket-owner-enforced
```

```
--object-ownership BucketOwnerEnforced
```

出力:

```
{  
  "Location": "/my-bucket"  
}
```

詳細については、[「Amazon S3 ユーザーガイド」の「オブジェクトの所有権の制御」と「無効化 ACLs」](#)を参照してください。Amazon S3

例 3: ``us-east-1`` リージョンの外にバケットを作成するには

次の create-bucket の例は、eu-west-1 リージョンに my-bucket という名前のバケットを作成します。us-east-1 の外にある目的のリージョンにバケットを作成するには、適切な LocationConstraint を指定する必要があります。

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region eu-west-1 \  
  --create-bucket-configuration LocationConstraint=eu-west-1
```

出力:

```
{  
  "Location": "http://my-bucket.s3.amazonaws.com/"  
}
```

詳細については、「Amazon S3 ユーザーガイド」の[「バケットの作成」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateBucket](#)」の「」を参照してください。

create-multipart-upload

次のコード例は、create-multipart-upload を使用する方法を示しています。

AWS CLI

次のコマンドは、キー multipart/01 を使用して、バケット my-bucket にマルチパートアップロードを作成します。

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

出力:

```
{
  "Bucket": "my-bucket",
  "UploadId":
  "dfRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3URC
  "Key": "multipart/01"
}
```

完成したファイルは、my-bucket バケット内の multipart というフォルダで 01 という名前が付けられます。upload-part コマンドで使用できるように、アップロード ID、キー、バケット名を保存します。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateMultipartUpload](#)」の「」を参照してください。

delete-bucket-analytics-configuration

次の例は、delete-bucket-analytics-configuration を使用する方法を説明しています。

AWS CLI

バケットの分析設定を削除するには

次の delete-bucket-analytics-configuration の例では、指定されたバケットと ID の分析設定を削除します。

```
aws s3api delete-bucket-analytics-configuration \
  --bucket my-bucket \
  --id 1
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketAnalyticsConfiguration](#)」の「」を参照してください。

delete-bucket-cors

次の例は、delete-bucket-cors を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットから Cross-Origin Resource Sharing 設定を削除します。

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketCors](#)」の「」を参照してください。

delete-bucket-encryption

次のコード例は、delete-bucket-encryption を使用する方法を示しています。

AWS CLI

バケットのサーバー側の暗号化設定を削除するには

次の delete-bucket-encryption の例では、指定したバケットのサーバー側の暗号化設定を削除します。

```
aws s3api delete-bucket-encryption \  
  --bucket my-bucket
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketEncryption](#)」の「」を参照してください。

delete-bucket-intelligent-tiering-configuration

次の例は、delete-bucket-intelligent-tiering-configuration を使用する方法を説明しています。

AWS CLI

バケットの S3 Intelligent-Tiering 設定を削除するには

次のdelete-bucket-intelligent-tiering-configuration例では、 という名前の S3 Intelligent-Tiering 設定をバケット ExampleConfigから削除します。

```
aws s3api delete-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id ExampleConfig
```

このコマンドでは何も出力されません。

詳細については、「Amazon [S3 ユーザーガイド](#)」の「[S3 Intelligent-Tiering の使用](#)」を参照してください。 Amazon S3

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteBucketIntelligentTieringConfiguration](#)」の「」を参照してください。

delete-bucket-inventory-configuration

次のコード例は、delete-bucket-inventory-configuration を使用する方法を示しています。

AWS CLI

バケットのインベントリ設定を削除するには

次の delete-bucket-inventory-configuration の例では、指定したバケットで ID 1 を持つインベントリ設定を削除します。

```
aws s3api delete-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteBucketInventoryConfiguration](#)」の「」を参照してください。

delete-bucket-lifecycle

次の例は、delete-bucket-lifecycle を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットからライフサイクル設定を削除します。

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketLifecycle](#)」の「」を参照してください。

delete-bucket-metrics-configuration

次の例は、delete-bucket-metrics-configuration を使用する方法を説明しています。

AWS CLI

バケットのメトリクス設定を削除するには

次の delete-bucket-metrics-configuration の例では、指定したバケットと ID のメトリクス設定を削除します。

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketMetricsConfiguration](#)」の「」を参照してください。

delete-bucket-ownership-controls

次のコード例は、delete-bucket-ownership-controls を使用する方法を示しています。

AWS CLI

バケットのバケット所有権設定を削除するには

次のdelete-bucket-ownership-controls例では、バケットのバケット所有権設定を削除します。

```
aws s3api delete-bucket-ownership-controls \  
  --bucket DOC-EXAMPLE-BUCKET
```

このコマンドでは何も出力されません。

詳細については、「Amazon S3 ユーザーガイド」の「[既存のバケットでのオブジェクト所有権の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketOwnershipControls](#)」の「」を参照してください。

delete-bucket-policy

次のコード例は、delete-bucket-policy を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットからバケットポリシーを削除します。

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketPolicy](#)」の「」を参照してください。

delete-bucket-replication

次のコード例は、delete-bucket-replication を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットからレプリケーションの設定を削除します。

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketReplication](#)」の「」を参照してください。

delete-bucket-tagging

次の例は、delete-bucket-tagging を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットからタグの設定を削除します。


```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketTagging](#)」の「」を参照してください。

delete-bucket-website

次の例は、delete-bucket-website を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットからウェブサイト設定を削除します。

```
aws s3api delete-bucket-website --bucket my-bucket
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucketWebsite](#)」の「」を参照してください。

delete-bucket

次のコード例は、delete-bucket を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットを削除します。

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteBucket](#)」の「」を参照してください。

delete-object-tagging

次の例は、delete-object-tagging を使用する方法を説明しています。

AWS CLI

オブジェクトのタグセットを削除するには

次の `delete-object-tagging` の例では、指定したキーを持つタグをオブジェクト `doc1.rtf` から削除します。

```
aws s3api delete-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteObjectTagging](#)」の「」を参照してください。

delete-object

次のコード例は、`delete-object` を使用する方法を示しています。

AWS CLI

次のコマンドは、`my-bucket` という名前のバケットから `test.txt` という名前のオブジェクトを削除します。

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

バケットバージョニングが有効になっている場合、出力には削除マーカのバージョン ID が含まれます。

```
{  
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tND1W1Gq",  
  "DeleteMarker": true  
}
```

オブジェクトの削除の詳細については、「Amazon S3 ユーザーガイド」の「オブジェクトの削除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteObject](#)」の「」を参照してください。

delete-objects

次のコード例は、`delete-objects` を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットからオブジェクトを削除します。

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

delete.json は、削除するオブジェクトを指定する現在のディレクトリ内のJSONドキュメントです。

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

出力:

```
{
  "Deleted": [
    {
      "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUP01koHwzU",
      "Key": "test1.txt",
      "DeleteMarker": true
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteObjects](#)」の「」を参照してください。

delete-public-access-block

次の例は、delete-public-access-block を使用する方法を説明しています。

AWS CLI

バケットのブロックパブリックアクセス設定を削除するには

次の `delete-public-access-block` の例では、指定したバケットのブロックパブリックアクセス設定を削除します。

```
aws s3api delete-public-access-block \  
  --bucket my-bucket
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeletePublicAccessBlock](#)」の「」を参照してください。

get-bucket-accelerate-configuration

次の例は、`get-bucket-accelerate-configuration` を使用する方法を説明しています。

AWS CLI

バケットの加速設定を取得するには

次の `get-bucket-accelerate-configuration` の例では、指定したバケットの加速設定を取得します。

```
aws s3api get-bucket-accelerate-configuration \  
  --bucket my-bucket
```

出力:

```
{  
  "Status": "Enabled"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetBucketAccelerateConfiguration](#)」の「」を参照してください。

get-bucket-acl

次のコード例は、`get-bucket-acl` を使用する方法を示しています。

AWS CLI

次のコマンドは、`my-bucket` という名前のバケットのアクセス制御リストを取得します。

```
aws s3api get-bucket-acl --bucket my-bucket
```

出力:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketAcl](#)」の「」を参照してください。

get-bucket-analytics-configuration

次のコード例は、get-bucket-analytics-configuration を使用する方法を示しています。

AWS CLI

特定の ID を持つバケットの分析設定を取得するには

次の get-bucket-analytics-configuration の例では、指定されたバケットと ID の分析設定を表示します。

```
aws s3api get-bucket-analytics-configuration \
  --bucket my-bucket \
  --id 1
```

出力:

```
{
  "AnalyticsConfiguration": {
    "StorageClassAnalysis": {},
    "Id": "1"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketAnalyticsConfiguration](#)」の「」を参照してください。

get-bucket-cors

次のコード例は、get-bucket-cors を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットの Cross-Origin Resource Sharing 設定を取得します。

```
aws s3api get-bucket-cors --bucket my-bucket
```

出力:

```
{
  "CORSRules": [
    {
      "AllowedHeaders": [
        "*"
      ],
      "ExposeHeaders": [
        "x-amz-server-side-encryption"
      ],
      "AllowedMethods": [
        "PUT",
        "POST",
        "DELETE"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedOrigins": [
        "http://www.example.com"
      ]
    }
  ]
}
```

```
    },
    {
      "AllowedHeaders": [
        "Authorization"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedMethods": [
        "GET"
      ],
      "AllowedOrigins": [
        "*"
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketCors](#)」の「」を参照してください。

get-bucket-encryption

次の例は、get-bucket-encryption を使用する方法を説明しています。

AWS CLI

バケットのサーバー側の暗号化設定を取得するには

次の get-bucket-encryption の例では、バケット my-bucket のサーバー側の暗号化設定を取得します。

```
aws s3api get-bucket-encryption \
  --bucket my-bucket
```

出力:

```
{
  "ServerSideEncryptionConfiguration": {
    "Rules": [
      {
        "ApplyServerSideEncryptionByDefault": {
          "SSEAlgorithm": "AES256"
        }
      }
    ]
  }
}
```

```
    }
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketEncryption](#)」の「」を参照してください。

get-bucket-intelligent-tiering-configuration

次のコード例は、`get-bucket-intelligent-tiering-configuration` を使用する方法を示しています。

AWS CLI

バケットで S3 Intelligent-Tiering 設定を取得するには

次の`get-bucket-intelligent-tiering-configuration`例では、`ExampleConfig` という名前の S3 Intelligent-Tiering 設定をバケット `ExampleConfig` で取得します。

```
aws s3api get-bucket-intelligent-tiering-configuration \
  --bucket DOC-EXAMPLE-BUCKET \
  --id ExampleConfig
```

出力:

```
{
  "IntelligentTieringConfiguration": {
    "Id": "ExampleConfig2",
    "Filter": {
      "Prefix": "images"
    },
    "Status": "Enabled",
    "Tierings": [
      {
        "Days": 90,
        "AccessTier": "ARCHIVE_ACCESS"
      },
      {
        "Days": 180,
```



```
        "AccessTier": "DEEP_ARCHIVE_ACCESS"
      }
    ]
  }
}
```

詳細については、「Amazon [S3 ユーザーガイド](#)」の「[S3 Intelligent-Tiering の使用](#)」を参照してください。 Amazon S3

- API 詳細については、AWS CLI 「[コマンドリファレンスGetBucketIntelligentTieringConfiguration](#)」の「」を参照してください。

get-bucket-inventory-configuration

次のコード例は、get-bucket-inventory-configuration を使用する方法を示しています。

AWS CLI

バケットのインベントリ設定を取得するには

次の get-bucket-inventory-configuration の例では、ID 1 を持つ指定したバケットのインベントリ設定を取得します。

```
aws s3api get-bucket-inventory-configuration \
  --bucket my-bucket \
  --id 1
```

出力:

```
{
  "InventoryConfiguration": {
    "IsEnabled": true,
    "Destination": {
      "S3BucketDestination": {
        "Format": "ORC",
        "Bucket": "arn:aws:s3:::my-bucket",
        "AccountId": "123456789012"
      }
    },
    "IncludedObjectVersions": "Current",
    "Id": "1",
    "Schedule": {
```

```
        "Frequency": "Weekly"
      }
    }
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketInventoryConfiguration](#)」の「」を参照してください。

get-bucket-lifecycle-configuration

次のコード例は、get-bucket-lifecycle-configuration を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットのライフサイクル設定を取得します。

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

出力:

```
{
  "Rules": [
    {
      "ID": "Move rotated logs to Glacier",
      "Prefix": "rotated/",
      "Status": "Enabled",
      "Transitions": [
        {
          "Date": "2015-11-10T00:00:00.000Z",
          "StorageClass": "GLACIER"
        }
      ]
    },
    {
      "Status": "Enabled",
      "Prefix": "",
      "NoncurrentVersionTransitions": [
        {
          "NoncurrentDays": 0,
          "StorageClass": "GLACIER"
        }
      ]
    }
  ],
}
```

```
        "ID": "Move old versions to Glacier"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketLifecycleConfiguration](#)」の「」を参照してください。

get-bucket-lifecycle

次のコード例は、get-bucket-lifecycle を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットのライフサイクル設定を取得します。

```
aws s3api get-bucket-lifecycle --bucket my-bucket
```

出力:

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (objects in logs/2015/)",
      "Prefix": "logs/2015/",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    },
    {
      "Expiration": {
        "Date": "2016-01-01T00:00:00.000Z"
      },
      "ID": "Delete 2014 logs in 2016.",
      "Prefix": "logs/2014/",
      "Status": "Enabled"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketLifecycle](#)」の「」を参照してください。

get-bucket-location

次のコード例は、get-bucket-location を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットの場所の制約を取得します (制約が存在する場合)。

```
aws s3api get-bucket-location --bucket my-bucket
```

出力:

```
{
  "LocationConstraint": "us-west-2"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketLocation](#)」の「」を参照してください。

get-bucket-logging

次の例は、get-bucket-logging を使用する方法を説明しています。

AWS CLI

バケットのログ記録ステータスを取得するには

次の get-bucket-logging の例では、指定したバケットのログ記録ステータスを取得します。

```
aws s3api get-bucket-logging \
  --bucket my-bucket
```

出力:

```
{
  "LoggingEnabled": {
```

```
    "TargetPrefix": "",
    "TargetBucket": "my-bucket-logs"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketLogging](#)」の「」を参照してください。

get-bucket-metrics-configuration

次の例は、get-bucket-metrics-configuration を使用する方法を説明しています。

AWS CLI

特定の ID を持つバケットのメトリクス設定を取得するには

次の get-bucket-metrics-configuration の例では、指定したバケットと ID のメトリクス設定を表示します。

```
aws s3api get-bucket-metrics-configuration \
  --bucket my-bucket \
  --id 123
```

出力:

```
{
  "MetricsConfiguration": {
    "Filter": {
      "Prefix": "logs"
    },
    "Id": "123"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketMetricsConfiguration](#)」の「」を参照してください。

get-bucket-notification-configuration

次の例は、get-bucket-notification-configuration を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットの通知設定を取得します。

```
aws s3api get-bucket-notification-configuration --bucket my-bucket
```

出力:

```
{
  "TopicConfigurations": [
    {
      "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWw1",
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
      "Events": [
        "s3:ObjectCreated:*"
      ]
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketNotificationConfiguration](#)」の「」を参照してください。

get-bucket-notification

次のコード例は、get-bucket-notification を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットの通知設定を取得します。

```
aws s3api get-bucket-notification --bucket my-bucket
```

出力:

```
{
  "TopicConfiguration": {
    "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
    "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWw1",
    "Event": "s3:ObjectCreated:*",
  }
}
```

```
    "Events": [  
      "s3:ObjectCreated:*"  
    ]  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketNotification](#)」の「」を参照してください。

get-bucket-ownership-controls

次のコード例は、get-bucket-ownership-controls を使用する方法を示しています。

AWS CLI

バケットのバケット所有権設定を取得するには

次のget-bucket-ownership-controls例では、バケットのバケット所有権設定を取得します。

```
aws s3api get-bucket-ownership-controls \  
  --bucket DOC-EXAMPLE-BUCKET
```

出力:

```
{  
  "OwnershipControls": {  
    "Rules": [  
      {  
        "ObjectOwnership": "BucketOwnerEnforced"  
      }  
    ]  
  }  
}
```

詳細については、「Amazon [S3 ユーザーガイド](#)」の「[S3 バケットのオブジェクト所有権設定の表示](#)」を参照してください。 Amazon S3

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketOwnershipControls](#)」の「」を参照してください。

get-bucket-policy-status

次のコード例は、get-bucket-policy-status を使用する方法を示しています。

AWS CLI

バケットがパブリックかどうかを示すバケットのポリシーステータスを取得するには

次の get-bucket-policy-status の例では、バケット my-bucket のポリシーステータスを取得します。

```
aws s3api get-bucket-policy-status \  
  --bucket my-bucket
```

出力:

```
{  
  "PolicyStatus": {  
    "IsPublic": false  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketPolicyStatus](#)」の「」を参照してください。

get-bucket-policy

次の例は、get-bucket-policy を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットのバケットポリシーを取得します。

```
aws s3api get-bucket-policy --bucket my-bucket
```

出力:

```
{  
  "Policy": "{\n\"Version\": \"2008-10-17\", \"Statement\": [\n{\n\"Sid\": \"\", \"Effect\n\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\":
```



```
\\"arn:aws:s3:::my-bucket/*\"},{\"Sid\\":\\\"\\\",\\\"Effect\\\":\\\"Deny\\\",\\\"Principal\\\":\\\"*\\\",\\\"Action\\\":\\\"s3:GetObject\\\",\\\"Resource\\\":\\\"arn:aws:s3:::my-bucket/secret/*\"}]}"
}
```

policyThe 次の例では、Amazon S3 バケットポリシーをダウンロードし、ファイルに変更を加え、を使用して変更されたバケットポリシーput-bucket-policyを適用する方法を示しています。バケットポリシーをファイルにダウンロードするには、以下を実行します。

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
policy.json
```

その後、必要に応じて policy.json ファイルを変更できます。最後に、次のコマンドを実行して、変更したポリシーを S3 バケットに適用することができます。

必要に応じて policy.json ファイル。最後に、次のコマンドを実行して、変更したポリシーを S3 バケットに適用することができます。

必要に応じてファイル。最後に、次のコマンドを実行して、変更したポリシーを S3 バケットに適用することができます。

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketPolicy](#)」の「」を参照してください。

get-bucket-replication

次の例は、get-bucket-replication を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットのレプリケーション設定を取得します。

```
aws s3api get-bucket-replication --bucket my-bucket
```

出力:

```
{
```

```
"ReplicationConfiguration": {
  "Rules": [
    {
      "Status": "Enabled",
      "Prefix": "",
      "Destination": {
        "Bucket": "arn:aws:s3:::my-bucket-backup",
        "StorageClass": "STANDARD"
      },
      "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
    }
  ],
  "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketReplication](#)」の「」を参照してください。

get-bucket-request-payment

次の例は、get-bucket-request-payment を使用する方法を説明しています。

AWS CLI

バケットのリクエストの支払い設定を取得するには

次の get-bucket-request-payment の例では、指定したバケットのリクエスト支払いの設定を取得します。

```
aws s3api get-bucket-request-payment \
  --bucket my-bucket
```

出力:

```
{
  "Payer": "BucketOwner"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketRequestPayment](#)」の「」を参照してください。

get-bucket-tagging

次の例は、get-bucket-tagging を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットのタグ付け設定を取得します。

```
aws s3api get-bucket-tagging --bucket my-bucket
```

出力:

```
{
  "TagSet": [
    {
      "Value": "marketing",
      "Key": "organization"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketTagging](#)」の「」を参照してください。

get-bucket-versioning

次のコード例は、get-bucket-versioning を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットのバージョンニング設定を取得します。

```
aws s3api get-bucket-versioning --bucket my-bucket
```

出力:

```
{
  "Status": "Enabled"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketVersioning](#)」の「」を参照してください。

get-bucket-website

次のコード例は、get-bucket-website を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットの静的ウェブサイト設定を取得します。

```
aws s3api get-bucket-website --bucket my-bucket
```

出力:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetBucketWebsite](#)」の「」を参照してください。

get-object-acl

次の例は、get-object-acl を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケット内のオブジェクトのアクセス制御リストを取得します。

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

出力:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    },
    {
      "Grantee": {
        "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
      },
      "Permission": "READ"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetObjectAcl](#)」の「」を参照してください。

get-object-attributes

次のコード例は、get-object-attributes を使用する方法を示しています。

AWS CLI

オブジェクト自体を返さずにオブジェクトからメタデータを取得するには

次の get-object-attributes の例は、オブジェクト doc1.rtf からメタデータを取得します。

```
aws s3api get-object-attributes \
  --bucket my-bucket \
  --key doc1.rtf \
  --object-attributes "StorageClass" "ETag" "ObjectSize"
```

出力:

```
{
  "LastModified": "2022-03-15T19:37:31+00:00",
  "VersionId": "IuCPjXTDzHNf1dAuitVBIKJpF2p1fg4P",
  "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",
  "StorageClass": "STANDARD",
  "ObjectSize": 405
}
```

詳細については、Amazon S3APIリファレンス[GetObjectAttributes](#)の「」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetObjectAttributes](#)」の「」を参照してください。

get-object-legal-hold

次のコード例は、get-object-legal-hold を使用する方法を示しています。

AWS CLI

オブジェクトのリーガルホールドステータスを取得する

次の get-object-legal-hold の例では、指定したオブジェクトのリーガルホールドステータスを取得します。

```
aws s3api get-object-legal-hold \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf
```

出力:

```
{
  "LegalHold": {
    "Status": "ON"
  }
}
```

- API 詳細については、AWS CLI「コマンドリファレンス[GetObjectLegalHold](#)」の「」を参照してください。

get-object-lock-configuration

次の例は、`get-object-lock-configuration` を使用する方法を説明しています。

AWS CLI

バケットのオブジェクトロック設定を取得するには

次の `get-object-lock-configuration` 例では、指定されたバケットのオブジェクトロック設定を取得します。

```
aws s3api get-object-lock-configuration \  
  --bucket my-bucket-with-object-lock
```

出力:

```
{  
  "ObjectLockConfiguration": {  
    "ObjectLockEnabled": "Enabled",  
    "Rule": {  
      "DefaultRetention": {  
        "Mode": "COMPLIANCE",  
        "Days": 50  
      }  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetObjectLockConfiguration](#)」の「」を参照してください。

get-object-retention

次の例は、`get-object-retention` を使用する方法を説明しています。

AWS CLI

オブジェクトのオブジェクト保持設定を取得するには

次の `get-object-retention` 例では、指定されたオブジェクトの保持設定を取得します。

```
aws s3api get-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

出力:

```
{  
  "Retention": {  
    "Mode": "GOVERNANCE",  
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetObjectRetention](#)」の「」を参照してください。

get-object-tagging

次の例は、get-object-tagging を使用する方法を説明しています。

AWS CLI

オブジェクトにアタッチされたタグを取得するには

次の get-object-tagging 指定したオブジェクトから指定したキーの値を取得します。

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

出力:

```
{  
  "TagSet": [  
    {  
      "Value": "confidential",  
      "Key": "designation"  
    }  
  ]  
}
```


次の `get-object-tagging` の例では、タグのないオブジェクト `doc2.rtf` のタグセットの取得を試行します。

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc2.rtf
```

出力:

```
{  
  "TagSet": []  
}
```

次の `get-object-tagging` の例では、複数のタグがあるオブジェクト `doc3.rtf` のタグセットの取得を試行します。

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc3.rtf
```

出力:

```
{  
  "TagSet": [  
    {  
      "Value": "confidential",  
      "Key": "designation"  
    },  
    {  
      "Value": "finance",  
      "Key": "department"  
    },  
    {  
      "Value": "payroll",  
      "Key": "team"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetObjectTagging](#)」の「」を参照してください。

get-object-torrent

次の例は、`get-object-torrent` を使用する方法を説明しています。

AWS CLI

次のコマンドは、`my-bucket` という名前のバケット内のオブジェクトの Torrent を作成します。

```
aws s3api get-object-torrent --bucket my-bucket --key large-video-file.mp4 large-video-file.torrent
```

Torrent ファイルは、現在のフォルダにローカルで保存されます。出力ファイル名 (`large-video-file.torrent`) はオプション名なしで指定され、コマンドの最後の引数である必要があります。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetObjectTorrent](#)」の「」を参照してください。

get-object

次の例は、`get-object` を使用する方法を説明しています。

AWS CLI

次の例は `get-object` コマンドを使用して、Amazon S3 からオブジェクトをダウンロードします。

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2 my_images.tar.bz2
```

`outfile` パラメータは、「`--outfile`」などのオプション名なしで指定されていることに注意してください。出力ファイルの名前はコマンドの最後のパラメータでなければなりません。

次の例は `--range` を使用して、オブジェクトから特定のバイト範囲をダウンロードする方法を示しています。バイト範囲には「`bytes=`」というプレフィックスを付ける必要があることに注意してください。

```
aws s3api get-object --bucket text-content --key dir/my_data --range bytes=8888-9999 my_data_range
```

オブジェクトの取得の詳細については、「Amazon S3 ユーザーガイド」の「オブジェクトを取得する」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetObject](#)」の「」を参照してください。

get-public-access-block

次の例は、get-public-access-block を使用する方法を説明しています。

AWS CLI

バケットのブロックパブリックアクセス設定を指定または削除するには

次の get-public-access-block の例では、指定したバケットのブロックパブリックアクセス設定を表示します。

```
aws s3api get-public-access-block \
  --bucket my-bucket
```

出力:

```
{
  "PublicAccessBlockConfiguration": {
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "BlockPublicAcls": true,
    "RestrictPublicBuckets": true
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPublicAccessBlock](#)」の「」を参照してください。

head-bucket

次の例は、head-bucket を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットへのアクセスを確認します。

```
aws s3api head-bucket --bucket my-bucket
```

バケットが存在し、そのバケットにアクセスできる場合、出力は返されません。それ以外の場合、エラーメッセージが表示されます。例:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- API 詳細については、AWS CLI 「コマンドリファレンス[HeadBucket](#)」の「」を参照してください。

head-object

次のコード例は、head-object を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケット内のオブジェクトのメタデータを取得します。

```
aws s3api head-object --bucket my-bucket --key index.html
```

出力:

```
{
  "AcceptRanges": "bytes",
  "ContentType": "text/html",
  "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
  "ContentLength": 77,
  "VersionId": "null",
  "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
  "Metadata": {}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[HeadObject](#)」の「」を参照してください。

list-bucket-analytics-configurations

次の例は、list-bucket-analytics-configurations を使用する方法を説明しています。

AWS CLI

バケットの分析設定のリストを取得するには

次の `list-bucket-analytics-configurations` の例では、指定したバケットの分析設定のリストを取得します。

```
aws s3api list-bucket-analytics-configurations \  
  --bucket my-bucket
```

出力:

```
{  
  "AnalyticsConfigurationList": [  
    {  
      "StorageClassAnalysis": {},  
      "Id": "1"  
    }  
  ],  
  "IsTruncated": false  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListBucketAnalyticsConfigurations](#)」の「」を参照してください。

`list-bucket-intelligent-tiering-configurations`

次の例は、`list-bucket-intelligent-tiering-configurations` を使用する方法を説明しています。

AWS CLI

バケットのすべての S3 Intelligent-Tiering 設定を取得するには

次の `list-bucket-intelligent-tiering-configurations` 例では、バケットのすべての S3 Intelligent-Tiering 設定を取得します。

```
aws s3api list-bucket-intelligent-tiering-configurations \  
  --bucket DOC-EXAMPLE-BUCKET
```

出力:

```
{
  "IsTruncated": false,
  "IntelligentTieringConfigurationList": [
    {
      "Id": "ExampleConfig",
      "Filter": {
        "Prefix": "images"
      },
      "Status": "Enabled",
      "Tierings": [
        {
          "Days": 90,
          "AccessTier": "ARCHIVE_ACCESS"
        },
        {
          "Days": 180,
          "AccessTier": "DEEP_ARCHIVE_ACCESS"
        }
      ]
    },
    {
      "Id": "ExampleConfig2",
      "Status": "Disabled",
      "Tierings": [
        {
          "Days": 730,
          "AccessTier": "ARCHIVE_ACCESS"
        }
      ]
    },
    {
      "Id": "ExampleConfig3",
      "Filter": {
        "Tag": {
          "Key": "documents",
          "Value": "taxes"
        }
      },
      "Status": "Enabled",
      "Tierings": [
        {
          "Days": 90,
```

```

        "AccessTier": "ARCHIVE_ACCESS"
      },
      {
        "Days": 365,
        "AccessTier": "DEEP_ARCHIVE_ACCESS"
      }
    ]
  }
]
}

```

詳細については、「Amazon [S3 ユーザーガイド](#)」の「[S3 Intelligent-Tiering の使用](#)」を参照してください。 Amazon S3

- API 詳細については、AWS CLI 「コマンドリファレンス [ListBucketIntelligentTieringConfigurations](#)」の「」を参照してください。

list-bucket-inventory-configurations

次のコード例は、list-bucket-inventory-configurations を使用する方法を示しています。

AWS CLI

バケットのインベントリ設定のリストを取得するには

次の list-bucket-inventory-configurations の例では、指定したバケットのインベントリ設定をリスト表示します。

```
aws s3api list-bucket-inventory-configurations \
  --bucket my-bucket
```

出力:

```

{
  "InventoryConfigurationList": [
    {
      "IsEnabled": true,
      "Destination": {
        "S3BucketDestination": {
          "Format": "ORC",
          "Bucket": "arn:aws:s3:::my-bucket",
          "AccountId": "123456789012"
        }
      }
    }
  ]
}

```

```
    }
  },
  "IncludedObjectVersions": "Current",
  "Id": "1",
  "Schedule": {
    "Frequency": "Weekly"
  }
},
{
  "IsEnabled": true,
  "Destination": {
    "S3BucketDestination": {
      "Format": "CSV",
      "Bucket": "arn:aws:s3:::my-bucket",
      "AccountId": "123456789012"
    }
  },
  "IncludedObjectVersions": "Current",
  "Id": "2",
  "Schedule": {
    "Frequency": "Daily"
  }
}
],
"IsTruncated": false
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBucketInventoryConfigurations](#)」の「」を参照してください。

list-bucket-metrics-configurations

次の例は、list-bucket-metrics-configurations を使用する方法を説明しています。

AWS CLI

バケットのメトリクス設定のリストを取得するには

次のlist-bucket-metrics-configurations例では、指定されたバケットのメトリクス設定のリストを取得します。

```
aws s3api list-bucket-metrics-configurations \
```



```
--bucket my-bucket
```

出力:

```
{
  "IsTruncated": false,
  "MetricsConfigurationList": [
    {
      "Filter": {
        "Prefix": "logs"
      },
      "Id": "123"
    },
    {
      "Filter": {
        "Prefix": "tmp"
      },
      "Id": "234"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBucketMetricsConfigurations](#)」の「」を参照してください。

list-buckets

次のコード例は、list-buckets を使用する方法を示しています。

AWS CLI

次のコマンドは、list-buckets コマンドを使用してすべての Amazon S3 バケット (すべてのリージョン) の名前を表示します。

```
aws s3api list-buckets --query "Buckets[].Name"
```

クエリオプションで list-buckets の出力をバケット名だけにフィルタリングします。

バケットの詳細については、「Amazon S3 ユーザーガイド」の「バケットの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListBuckets](#)」の「」を参照してください。

list-multipart-uploads

次のコード例は、list-multipart-uploads を使用する方法を示しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットのすべてのアクティブなマルチパートアップロードを一覧表示します。

```
aws s3api list-multipart-uploads --bucket my-bucket
```

出力:

```
{
  "Uploads": [
    {
      "Initiator": {
        "DisplayName": "username",
        "ID": "arn:aws:iam::0123456789012:user/username"
      },
      "Initiated": "2015-06-02T18:01:30.000Z",
      "UploadId":
      "dfRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZlJF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC",
      "StorageClass": "STANDARD",
      "Key": "multipart/01",
      "Owner": {
        "DisplayName": "aws-account-name",
        "ID":
        "100719349fc3b6dcd7c820a124bf7aec408092c3d7b51b38494939801fc248b"
      }
    }
  ],
  "CommonPrefixes": []
}
```

進行中のマルチパートアップロードでは、Amazon S3 のストレージコストが発生します。アクティブなマルチパートアップロードを完了または中止して、その一部をアカウントから削除します。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMultipartUploads](#)」の「」を参照してください。

list-object-versions

次の例は、list-object-versions を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケット内のオブジェクトのバージョン情報を取得します。

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

出力:

```
{
  "DeleteMarkers": [
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": true,
      "VersionId": "B2VsEK5saUNNHKc0AJj7hIE86RozToyq",
      "Key": "index.html",
      "LastModified": "2015-11-10T00:57:03.000Z"
    },
    {
      "Owner": {
        "DisplayName": "my-username",
        "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
      },
      "IsLatest": false,
      "VersionId": ".FLQEZscLIcfxSq.jsFJ.szUkmng2Yw6",
      "Key": "index.html",
      "LastModified": "2015-11-09T23:32:20.000Z"
    }
  ],
  "Versions": [
```

```
{
  "LastModified": "2015-11-10T00:20:11.000Z",
  "VersionId": "Rb_l2T8UHDkFEwCgJjhlGPOZC0qJ.vpD",
  "ETag": "\"0622528de826c0df5db1258a23b80be5\"",
  "StorageClass": "STANDARD",
  "Key": "index.html",
  "Owner": {
    "DisplayName": "my-username",
    "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
  },
  "IsLatest": false,
  "Size": 38
},
{
  "LastModified": "2015-11-09T23:26:41.000Z",
  "VersionId": "rasWWGpgk9E4s0LyTJgusGeRQKLVIAff",
  "ETag": "\"06225825b8028de826c0df5db1a23be5\"",
  "StorageClass": "STANDARD",
  "Key": "index.html",
  "Owner": {
    "DisplayName": "my-username",
    "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
  },
  "IsLatest": false,
  "Size": 38
},
{
  "LastModified": "2015-11-09T22:50:50.000Z",
  "VersionId": "null",
  "ETag": "\"d1f45267a863c8392e07d24dd592f1b9\"",
  "StorageClass": "STANDARD",
  "Key": "index.html",
  "Owner": {
    "DisplayName": "my-username",
    "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
  },
  "IsLatest": false,
  "Size": 533823
}
]
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListObjectVersions](#)」の「」を参照してください。

list-objects-v2

次の例は、list-objects-v2 を使用する方法を説明しています。

AWS CLI

バケット内のオブジェクトのリストを取得するには

次の list-objects-v2 の例では、指定したバケットのオブジェクトを一覧表示します。

```
aws s3api list-objects-v2 \  
  --bucket my-bucket
```

出力:

```
{  
  "Contents": [  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"621503c373607d548b37cff8778d992c\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc1.rtf",  
      "Size": 391  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc2.rtf",  
      "Size": 373  
    },  
    {  
      "LastModified": "2019-11-05T23:11:50.000Z",  
      "ETag": "\"08210852f65a2e9cb999972539a64d68\"",  
      "StorageClass": "STANDARD",  
      "Key": "doc3.rtf",  
    }  
  ]  
}
```

```
        "Size": 399
      },
      {
        "LastModified": "2019-11-05T23:11:50.000Z",
        "ETag": "\"d1852dd683f404306569471af106988e\"",
        "StorageClass": "STANDARD",
        "Key": "doc4.rtf",
        "Size": 6225
      }
    ]
  }
}
```

- API 詳細については、「コマンドリファレンス」の[ListObjectsV2](#)を参照してください。
AWS CLI

list-objects

次の例は、list-objects を使用する方法を説明しています。

AWS CLI

次の例は、list-objects コマンドを使用して、指定されたバケット内のすべてのオブジェクトの名前を表示します。

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

この例では、--query 引数を使用して list-objects の出力を各オブジェクトのキー値とサイズでフィルタリングしています。

オブジェクトの詳細については、「Amazon S3 デベロッパーガイド」の「Working with Amazon S3 Objects」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListObjects](#)」の「」を参照してください。

list-parts

次の例は、list-parts を使用する方法を説明しています。

AWS CLI

次のコマンドは、バケット `multipart/01` 内のキーを使用してマルチパートアップロード用にアップロードされたすべてのパートを一覧表示します `my-bucket`。

```
aws s3api list-parts --bucket my-bucket --key 'multipart/01' --upload-id dfRtDYU0WMCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P3UR
```

出力:

```
{
  "Owner": {
    "DisplayName": "aws-account-name",
    "ID": "100719349fc3b6dcd7c820a124bf7aec408092c3d7b51b38494939801fc248b"
  },
  "Initiator": {
    "DisplayName": "username",
    "ID": "arn:aws:iam::0123456789012:user/username"
  },
  "Parts": [
    {
      "LastModified": "2015-06-02T18:07:35.000Z",
      "PartNumber": 1,
      "ETag": "\"e868e0f4719e394144ef36531ee6824c\"",
      "Size": 5242880
    },
    {
      "LastModified": "2015-06-02T18:07:42.000Z",
      "PartNumber": 2,
      "ETag": "\"6bb2b12753d66fe86da4998aa33fffb0\"",
      "Size": 5242880
    },
    {
      "LastModified": "2015-06-02T18:07:47.000Z",
      "PartNumber": 3,
      "ETag": "\"d0a0112e841abec9c9ec83406f0159c8\"",
      "Size": 5242880
    }
  ],
  "StorageClass": "STANDARD"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListParts](#)」の「」を参照してください。

ls

次のコード例は、ls を使用する方法を示しています。

AWS CLI

例 1: ユーザー所有のすべてのバケットを一覧表示する

次のlsコマンドは、ユーザーが所有しているすべてのバケットを一覧表示します。この例では、ユーザーはバケットmybucketとを所有していますmybucket2。タイムスタンプは、バケットが作成された日付で、マシンのタイムゾーンに表示されます。この日付は、バケットポリシーの編集など、バケットに変更を加えると変更される可能性があります。s3:// がパス引数 に使用される場合<S3Uri>、すべてのバケットも一覧表示されます。

```
aws s3 ls
```

出力:

```
2013-07-11 17:08:50 mybucket
2013-07-24 14:55:44 mybucket2
```

例 2: バケット内のすべてのプレフィックスとオブジェクトを一覧表示する

次のlsコマンドは、指定されたバケットとプレフィックスの下にあるオブジェクトと一般的なプレフィックスを一覧表示します。この例では、ユーザーは オブジェクトtest.txtとmybucketを使用してバケットを所有しますsomePrefix/test.txt。LastWriteTime とLengthは任意です。ls コマンドはローカルファイルシステムとやり取りしないため、あいまいさを解決するためにs3://URIスキームは不要であり、省略される場合があることに注意してください。

```
aws s3 ls s3://mybucket
```

出力:

```
PRE somePrefix/
```



```
2013-07-25 17:06:27      88 test.txt
```

例 3: 特定のバケットとプレフィックス内のすべてのプレフィックスとオブジェクトを一覧表示する

次のlsコマンドは、指定されたバケットとプレフィックスの下にあるオブジェクトと一般的なプレフィックスを一覧表示します。ただし、指定されたバケットとプレフィックスの下にオブジェクトや一般的なプレフィックスはありません。

```
aws s3 ls s3://mybucket/noExistPrefix
```

出力:

```
None
```

例 4: バケット内のすべてのプレフィックスとオブジェクトを再帰的に一覧表示する

次のlsコマンドは、バケット内のオブジェクトを再帰的に一覧表示します。出力PRE dirname/に表示されるのではなく、バケット内のすべてのコンテンツが順番に表示されます。

```
aws s3 ls s3://mybucket \
  --recursive
```

出力:

```
2013-09-02 21:37:53      10 a.txt
2013-09-02 21:37:53 2863288 foo.zip
2013-09-02 21:32:57      23 foo/bar/.baz/a
2013-09-02 21:32:58      41 foo/bar/.baz/b
2013-09-02 21:32:57     281 foo/bar/.baz/c
2013-09-02 21:32:57      73 foo/bar/.baz/d
2013-09-02 21:32:57     452 foo/bar/.baz/e
2013-09-02 21:32:57     896 foo/bar/.baz/hooks/bar
2013-09-02 21:32:57     189 foo/bar/.baz/hooks/foo
2013-09-02 21:32:57     398 z.txt
```

例 5: バケット内のすべてのプレフィックスとオブジェクトを要約する

次のlsコマンドは、--human-readable および --summarize オプションを使用して同じコマンドを示しています。--human-readable は、ファイルサイズを で表示しますBytes/MiB/KiB/GiB/TiB/

PiB/EiB。--summarize は、結果リストの最後にオブジェクトの合計数と合計サイズを表示します。

```
aws s3 ls s3://mybucket \  
  --recursive \  
  --human-readable \  
  --summarize
```

出力:

```
2013-09-02 21:37:53  10 Bytes a.txt  
2013-09-02 21:37:53 2.9 MiB foo.zip  
2013-09-02 21:32:57  23 Bytes foo/bar/.baz/a  
2013-09-02 21:32:58  41 Bytes foo/bar/.baz/b  
2013-09-02 21:32:57 281 Bytes foo/bar/.baz/c  
2013-09-02 21:32:57  73 Bytes foo/bar/.baz/d  
2013-09-02 21:32:57 452 Bytes foo/bar/.baz/e  
2013-09-02 21:32:57 896 Bytes foo/bar/.baz/hooks/bar  
2013-09-02 21:32:57 189 Bytes foo/bar/.baz/hooks/foo  
2013-09-02 21:32:57 398 Bytes z.txt  
  
Total Objects: 10  
Total Size: 2.9 MiB
```

例 6: S3 アクセスポイントからの一覧表示

次のlsコマンドは、アクセスポイント () からのオブジェクトを一覧表示しますmyaccesspoint。

```
aws s3 ls s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

出力:

```
                PRE somePrefix/  
2013-07-25 17:06:27      88 test.txt
```

- API 詳細については、AWS CLI 「コマンドリファレンス」の「[Ls](#)」を参照してください。

mb

次の例は、mb を使用する方法を説明しています。

AWS CLI

例 1: バケットを作成する

次のmbコマンドはバケットを作成します。この例では、ユーザーはバケットを作成しますmybucket。バケットは、ユーザーの設定ファイルで指定されたリージョンに作成されます。

```
aws s3 mb s3://mybucket
```

出力:

```
make_bucket: s3://mybucket
```

例 2: 指定されたリージョンにバケットを作成する

次のmbコマンドは、`--region`パラメータで指定されたリージョンにバケットを作成します。この例では、ユーザーはリージョン mybucket のバケットを作成しますus-west-1。

```
aws s3 mb s3://mybucket \  
--region us-west-1
```

出力:

```
make_bucket: s3://mybucket
```

- API 詳細については、「[コマンドリファレンス](#)」の「[MB](#)」を参照してください。AWS CLI

mv

次の例は、mv を使用する方法を説明しています。

AWS CLI

例 1: ローカルファイルを指定されたバケットに移動する

次のmvコマンドは、1つのファイルを指定されたバケットとキーに移動します。

```
aws s3 mv test.txt s3://mybucket/test2.txt
```

出力:

```
move: test.txt to s3://mybucket/test2.txt
```

例 2: オブジェクトを指定されたバケットとキーに移動する

次のmvコマンドは、単一の s3 オブジェクトを指定されたバケットとキーに移動します。

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt
```

出力:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

例 3: S3 オブジェクトをローカルディレクトリに移動する

次のmvコマンドは、1つのオブジェクトを指定したファイルにローカルに移動します。

```
aws s3 mv s3://mybucket/test.txt test2.txt
```

出力:

```
move: s3://mybucket/test.txt to test2.txt
```

例 4: 元の名前のオブジェクトを指定されたバケットに移動する

次のmvコマンドは、元の名前を保持しながら、単一のオブジェクトを指定されたバケットに移動します。

```
aws s3 mv s3://mybucket/test.txt s3://mybucket2/
```

出力:

```
move: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

例 5: バケット内のすべてのオブジェクトとプレフィックスをローカルディレクトリに移動する

パラメータで渡されると--recursive、次のmvコマンドは、指定されたプレフィックスとバケットのすべてのオブジェクトを指定されたディレクトリに再帰的に移動します。この例では、バケットにオブジェクトtest1.txtとmybucketがありますtest2.txt。

```
aws s3 mv s3://mybucket . \  
--recursive
```

出力:

```
move: s3://mybucket/test1.txt to test1.txt  
move: s3://mybucket/test2.txt to test2.txt
```

例 6: バケット内のすべてのオブジェクトとプレフィックスをローカルディレクトリに移動します。ただし、「.jpg」ファイルは除きます。

パラメータで渡されると--recursive、次のmvコマンドは、--excludeパラメータを使用して一部のファイルを除外しながら、指定されたディレクトリ内のすべてのファイルを指定されたバケットとプレフィックスに再帰的に移動します。この例では、ディレクトリにファイルtest1.txtとmyDirがありますtest2.jpg。

```
aws s3 mv myDir s3://mybucket/ \  
--recursive \  
--exclude "*.jpg"
```

出力:

```
move: myDir/test1.txt to s3://mybucket2/test1.txt
```

例 7: バケット内のすべてのオブジェクトとプレフィックスをローカルディレクトリに移動します。ただし、指定されたプレフィックスは除きます。

パラメータで渡されると--recursive、次のmvコマンドは、--excludeパラメータを使用して一部のオブジェクトを除外しながら、指定されたバケット内のすべてのオブジェクトを再帰的に別のバケットに移動します。この例では、バケットmybucketにオブジェクトtest1.txtとがありますanother/test1.txt。

```
aws s3 mv s3://mybucket/ s3://mybucket2/ \  
--recursive \  
--exclude "mybucket/another/*"
```

出力:

```
move: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

例 8: オブジェクトを指定されたバケットに移動し、 を設定する ACL

次のmvコマンドは、 を ACL に設定する際に、単一のオブジェクトを指定されたバケットとキーに移動しますpublic-read-write。

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt \  
--acl public-read-write
```

出力:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

例 9: ローカルファイルを指定されたバケットに移動し、アクセス許可を付与する

次のmvコマンドは、 --grantsオプションを使用して、すべてのユーザーに読み取りアクセス権を付与し、メールアドレスで識別された特定のユーザーにフルコントロールを付与する方法を示しています。

```
aws s3 mv file.txt s3://mybucket/ \  
--grants read=uri=http://acs.amazonaws.com/groups/global/  
AllUsers full=emailaddress=user@example.com
```

出力:

```
move: file.txt to s3://mybucket/file.txt
```

例 10: ファイルを S3 アクセスポイントに移動する

次のmvコマンドは、 という名前のキーmyaccesspointで という名前のアクセスポイントmydoc.txtに という名前の単一のファイルを移動しますmykey。

```
aws s3 mv mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

出力:

```
move: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/  
mykey
```

- API 詳細については、「[コマンドリファレンス](#)」の「[Mv](#)」を参照してください。AWS CLI

presign

次のコード例は、presign を使用する方法を示しています。

AWS CLI

例 1: S3 バケット内のオブジェクトにリンクするデフォルトの 1 時間の有効期間URLで署名付きを作成するには

次のpresignコマンドは、1 時間有効な指定されたバケットとキーURLに対して署名付き を生成します。

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt
```

出力:

```
https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

例 2: S3 バケット内のオブジェクトにリンクするカスタムライフタイムURLで署名付きを作成するには

次のpresignコマンドは、指定されたバケットと 1 週間有効なキーURLに対して署名付き を生成します。

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt \
  --expires-in 604800
```

出力:

```
https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/key?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAEXAMPLE123456789%2F20210621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Date=20210621T041609Z&X-Amz-Expires=604800&X-Amz-SignedHeaders=host&X-Amz-Signature=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

詳細については、「S3 デベロッパーガイド」の [「オブジェクトを他のユーザーと共有する」](#) を参照してください。

- API 詳細については、「コマンドリファレンス」の「[Presign](#)」を参照してください。AWS CLI

put-bucket-accelerate-configuration

次の例は、put-bucket-accelerate-configuration を使用する方法を説明しています。

AWS CLI

バケットの加速設定を指定するには

次の put-bucket-accelerate-configuration の例では、指定したバケットの加速設定を有効にします。

```
aws s3api put-bucket-accelerate-configuration \  
  --bucket my-bucket \  
  --accelerate-configuration Status=Enabled
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketAccelerateConfiguration](#)」の「」を参照してください。

put-bucket-acl

次の例は、put-bucket-acl を使用する方法を説明しています。

AWS CLI

この例ではfull control、2人のAWSユーザー (user1@example.com と user2@example.com) に を付与し、すべてのユーザーに read アクセス許可を付与します。

```
aws s3api put-bucket-acl --bucket MyBucket --grant-full-  
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-  
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

カスタムの詳細については、<http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> を参照してください ACLs (などの s3api ACL コマンドはput-bucket-acl、同じ短縮引数表記を使用します)。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketAcl](#)」の「」を参照してください。

put-bucket-analytics-configuration

次のコード例は、put-bucket-analytics-configuration を使用する方法を示しています。

AWS CLI

バケットの分析設定を設定するには

次のput-bucket-analytics-configuration例では、指定されたバケットの分析を設定します。

```
aws s3api put-bucket-analytics-configuration \  
  --bucket my-bucket --id 1 \  
  --analytics-configuration '{"Id": "1", "StorageClassAnalysis": {}}'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketAnalyticsConfiguration](#)」の「」を参照してください。

put-bucket-cors

次の例は、put-bucket-cors を使用する方法を説明しています。

AWS CLI

次の例は、www.example.com からの PUT、POST、および DELETE の各リクエストを有効化し、任意のドメインからの GET リクエストを有効化します。

```
aws s3api put-bucket-cors --bucket MyBucket --cors-configuration file://cors.json  
  
cors.json:  
{  
  "CORSRules": [  
    {  
      "AllowedOrigins": ["http://www.example.com"],  
      "AllowedHeaders": ["*"],
```

```

    "AllowedMethods": ["PUT", "POST", "DELETE"],
    "MaxAgeSeconds": 3000,
    "ExposeHeaders": ["x-amz-server-side-encryption"]
  },
  {
    "AllowedOrigins": ["*"],
    "AllowedHeaders": ["Authorization"],
    "AllowedMethods": ["GET"],
    "MaxAgeSeconds": 3000
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketCors](#)」の「」を参照してください。

put-bucket-encryption

次のコード例は、put-bucket-encryption を使用する方法を示しています。

AWS CLI

バケットのサーバー側の暗号化を設定するには

次のput-bucket-encryption例では、指定されたバケットのデフォルトとしてAES256暗号化を設定します。

```

aws s3api put-bucket-encryption \
  --bucket my-bucket \
  --server-side-encryption-configuration '{"Rules":
  [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'

```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketEncryption](#)」の「」を参照してください。

put-bucket-intelligent-tiering-configuration

次の例は、put-bucket-intelligent-tiering-configuration を使用する方法を説明しています。

AWS CLI

バケットの S3 Intelligent-Tiering 設定を更新するには

次のput-bucket-intelligent-tiering-configuration例では、という名前の S3 Intelligent-Tiering 設定をバケット ExampleConfigで更新します。この設定では、プレフィックスイメージでアクセスされていないオブジェクトは、90 日後にアーカイブアクセス、180 日後にディープアーカイブアクセスに移行されます。

```
aws s3api put-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id "ExampleConfig" \  
  --intelligent-tiering-configuration file://intelligent-tiering-  
configuration.json
```

intelligent-tiering-configuration.json の内容:

```
{  
  "Id": "ExampleConfig",  
  "Status": "Enabled",  
  "Filter": {  
    "Prefix": "images"  
  },  
  "Tierings": [  
    {  
      "Days": 90,  
      "AccessTier": "ARCHIVE_ACCESS"  
    },  
    {  
      "Days": 180,  
      "AccessTier": "DEEP_ARCHIVE_ACCESS"  
    }  
  ]  
}
```

このコマンドでは何も出力されません。

詳細については、Amazon S3ユーザーガイド」の「[既存のバケットでのオブジェクト所有権の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutBucketIntelligentTieringConfiguration](#)」の「」を参照してください。

put-bucket-inventory-configuration

次の例は、put-bucket-inventory-configuration を使用する方法を説明しています。

AWS CLI

例 1: バケットのインベントリ設定を設定するには

次のput-bucket-inventory-configuration例では、バケットの週ORC次形式のインベントリレポートを設定しますmy-bucket。

```
aws s3api put-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1 \  
  --inventory-configuration '{"Destination": { "S3BucketDestination":  
  { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::my-bucket", "Format":  
  "ORC" }}, "IsEnabled": true, "Id": "1", "IncludedObjectVersions": "Current",  
  "Schedule": { "Frequency": "Weekly" } }'
```

このコマンドでは何も出力されません。

例 2: バケットのインベントリ設定を設定するには

次のput-bucket-inventory-configuration例では、バケットの日CSV次形式のインベントリレポートを設定しますmy-bucket。

```
aws s3api put-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 2 \  
  --inventory-configuration '{"Destination": { "S3BucketDestination":  
  { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::my-bucket", "Format":  
  "CSV" }}, "IsEnabled": true, "Id": "2", "IncludedObjectVersions": "Current",  
  "Schedule": { "Frequency": "Daily" } }'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketInventoryConfiguration](#)」の「」を参照してください。

put-bucket-lifecycle-configuration

次の例は、put-bucket-lifecycle-configuration を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-bucket という名前のバケットにライフサイクル設定を適用します。

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-configuration file://lifecycle.json
```

ファイルは、2つのルールを指定する現在のフォルダ内のJSONドキュメント lifecycle.json です。

```
{
  "Rules": [
    {
      "ID": "Move rotated logs to Glacier",
      "Prefix": "rotated/",
      "Status": "Enabled",
      "Transitions": [
        {
          "Date": "2015-11-10T00:00:00.000Z",
          "StorageClass": "GLACIER"
        }
      ]
    },
    {
      "Status": "Enabled",
      "Prefix": "",
      "NoncurrentVersionTransitions": [
        {
          "NoncurrentDays": 2,
          "StorageClass": "GLACIER"
        }
      ],
      "ID": "Move old versions to Glacier"
    }
  ]
}
```

1つ目のルールは、指定した日付に、プレフィックス rotated の付いたファイルを Glacier に移動します。2つ目のルールは、最新でない古いオブジェクトバージョンを Glacier に移動します。許容可能なタイムスタンプ形式については、AWS CLI ユーザーガイドの「パラメータ値の指定」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketLifecycleConfiguration](#)」の「」を参照してください。

put-bucket-lifecycle

次の例は、put-bucket-lifecycle を使用する方法を説明しています。

AWS CLI

次のコマンドは、バケット にライフサイクル設定を適用しますmy-bucket。

```
aws s3api put-bucket-lifecycle --bucket my-bucket --lifecycle-configuration file://lifecycle.json
```

ファイルは、2 つのルールを指定する現在のフォルダ内のJSONドキュメントlifecycle.jsonです。

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (objects in logs/2015/)",
      "Prefix": "logs/2015/",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    },
    {
      "Expiration": {
        "Date": "2016-01-01T00:00:00.000Z"
      },
      "ID": "Delete 2014 logs in 2016.",
      "Prefix": "logs/2014/",
      "Status": "Enabled"
    }
  ]
}
```

最初のルールは、60 日後にファイルを Amazon Glacier に移動します。2 番目のルールは、指定した日付に Amazon S3 からファイルを削除します。許容可能なタイムスタンプ形式の詳細については、AWS CLI ユーザーガイドの「パラメータ値の指定」を参照してください。

上記の例の各ルールは、ポリシー (Transition または Expiration) と、適用されるファイルプレフィックス (フォルダ名) を指定します。空白のプレフィックスを指定して、バケット全体に適用されるルールを作成することもできます。

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (all objects in bucket)",
      "Prefix": "",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketLifecycle](#)」の「」を参照してください。

put-bucket-logging

次の例は、put-bucket-logging を使用する方法を説明しています。

AWS CLI

例 1: バケットポリシーのログ記録を設定するには

次のput-bucket-logging例では、 のログ記録ポリシーを設定しますMyBucket。まず、バケットポリシーを使用して、ログ記録サービスプリンシパルに許可を付与します。

```
aws s3api put-bucket-policy \
  --bucket MyBucket \
  --policy file://policy.json
```

policy.json の内容:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3ServerAccessLogsPolicy",
      "Effect": "Allow",
      "Principal": {"Service": "logging.s3.amazonaws.com"},
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::MyBucket/Logs/*",
      "Condition": {
        "ArnLike": {"aws:SourceARN": "arn:aws:s3:::SOURCE-BUCKET-NAME"},
        "StringEquals": {"aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"}
      }
    }
  ]
}
```

ログ記録ポリシーを適用するには、`put-bucket-logging` を使用します。

```
aws s3api put-bucket-logging \
  --bucket MyBucket \
  --bucket-logging-status file://logging.json
```

`logging.json` の内容:

```
{
  "LoggingEnabled": {
    "TargetBucket": "MyBucket",
    "TargetPrefix": "Logs/"
  }
}
```

`put-bucket-policy` コマンドは、ログ記録サービスプリンシパルに `s3:PutObject` アクセス許可を付与するために必要です。

詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[Amazon S3 サーバーアクセスのログ記録](#)」を参照してください。

例 2: 1 人のユーザーのみにアクセスをログ記録するためのバケットポリシーを設定するには

次のput-bucket-logging例では、のログ記録ポリシーを設定しますMyBucket。AWS ユーザー bob@example.com はログファイルを完全に制御でき、他の誰もアクセスできません。まず、put-bucket-acl で S3 アクセス許可を付与します。

```
aws s3api put-bucket-acl \  
  --bucket MyBucket \  
  --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery \  
  --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

次に、put-bucket-logging を使用してログ記録ポリシーを適用します。

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --bucket-logging-status file://logging.json
```

logging.json の内容:

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "MyBucket",  
    "TargetPrefix": "MyBucketLogs/",  
    "TargetGrants": [  
      {  
        "Grantee": {  
          "Type": "AmazonCustomerByEmail",  
          "EmailAddress": "bob@example.com"  
        },  
        "Permission": "FULL_CONTROL"  
      }  
    ]  
  }  
}
```

put-bucket-acl コマンドは、Amazon S3 のログ配信システムに必要なアクセス許可 (write および read-acp アクセス許可) を付与するために必要です。

詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「[Amazon S3 サーバーアクセスログ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketLogging](#)」の「」を参照してください。

put-bucket-metrics-configuration

次のコード例は、put-bucket-metrics-configuration を使用する方法を示しています。

AWS CLI

バケットのメトリクス設定を設定するには

次のput-bucket-metrics-configuration例では、指定されたバケットの ID 123 のメトリクス設定を設定します。

```
aws s3api put-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123 \  
  --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketMetricsConfiguration](#)」の「」を参照してください。

put-bucket-notification-configuration

次の例は、put-bucket-notification-configuration を使用する方法を説明しています。

AWS CLI

バケットへの指定された通知を有効にするには

次の put-bucket-notification-configuration の例では、my-bucket という名前のバケットに通知設定を適用します。ファイルは、モニタリングするSNSトピックとイベントタイプを指定する現在のフォルダ内のJSONドキュメントnotification.jsonです。

```
aws s3api put-bucket-notification-configuration \  
  --bucket my-bucket \  
  --notification-configuration file://notification.json
```

notification.json の内容:

```
{
```

```
"TopicConfigurations": [  
  {  
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",  
    "Events": [  
      "s3:ObjectCreated:*"  
    ]  
  }  
]
```

SNS トピックには、Amazon S3 が公開できるようにする IAM ポリシーがアタッチされている必要があります。

```
{  
  "Version": "2008-10-17",  
  "Id": "example-ID",  
  "Statement": [  
    {  
      "Sid": "example-statement-ID",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "s3.amazonaws.com"  
      },  
      "Action": [  
        "SNS:Publish"  
      ],  
      "Resource": "arn:aws:sns:us-west-2:123456789012::s3-notification-topic",  
      "Condition": {  
        "ArnLike": {  
          "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"  
        }  
      }  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutBucketNotificationConfiguration](#)」の「」を参照してください。

put-bucket-notification

次の例は、put-bucket-notification を使用する方法を説明しています。

AWS CLI

my-bucket という名前のバケットに通知設定を適用します。

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration file://notification.json
```

ファイルは、モニタリングするSNSトピックとイベントタイプを指定する現在のフォルダ内のJSONドキュメントnotification.jsonです。

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

SNS トピックには、Amazon S3 が公開できるようにするIAMポリシーがアタッチされている必要があります。

```
{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:us-west-2:123456789012:my-bucket",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
        }
      }
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketNotification](#)」の「」を参照してください。

put-bucket-ownership-controls

次のコード例は、put-bucket-ownership-controls を使用する方法を示しています。

AWS CLI

バケットのバケット所有権設定を更新するには

次のput-bucket-ownership-controls例では、バケットのバケット所有権設定を更新します。

```
aws s3api put-bucket-ownership-controls \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

このコマンドでは何も出力されません。

詳細については、Amazon S3ユーザーガイド」の「[既存のバケットのオブジェクト所有権の設定](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketOwnershipControls](#)」の「」を参照してください。

put-bucket-policy

次のコード例は、put-bucket-policy を使用する方法を示しています。

AWS CLI

この例では、すべてのユーザーが、内のオブジェクトMyBucketを除くすべてのオブジェクトを取得できますMySecretFolder。また、AWS アカウントのルートユーザーに putと アクセスdelete許可を付与します1234-5678-9012。

```
aws s3api put-bucket-policy --bucket MyBucket --policy file:///policy.json  
  
policy.json:  
{  
  "Statement": [  
    {  
      "Action": "s3:DeleteObject",  
      "Effect": "Deny",  
      "Principal": "*",  
      "Resource": "arn:aws:s3:::1234-5678-9012/MySecretFolder/*",  
      "Sid": "DenyDeleteSecretFolderObjects",  
      "Condition": {"StringNotEquals": {"s3:prefix": "MySecretFolder/"}}  
    }  
  ]  
}
```

```
{
  "Effect": "Allow",
  "Principal": "*",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::MyBucket/*"
},
{
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::MyBucket/MySecretFolder/*"
},
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:root"
  },
  "Action": [
    "s3:DeleteObject",
    "s3:PutObject"
  ],
  "Resource": "arn:aws:s3:::MyBucket/*"
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketPolicy](#)」の「」を参照してください。

put-bucket-replication

次のコード例は、put-bucket-replication を使用する方法を示しています。

AWS CLI

S3 バケットのレプリケーションを設定するには

次の put-bucket-replication の例では、指定した S3 バケットにレプリケーション設定を適用します。

```
aws s3api put-bucket-replication \
```

```
--bucket AWSDOC-EXAMPLE-BUCKET1 \  
--replication-configuration file://replication.json
```

replication.json の内容:

```
{  
  "Role": "arn:aws:iam::123456789012:role/s3-replication-role",  
  "Rules": [  
    {  
      "Status": "Enabled",  
      "Priority": 1,  
      "DeleteMarkerReplication": { "Status": "Disabled" },  
      "Filter" : { "Prefix": ""},  
      "Destination": {  
        "Bucket": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2"  
      }  
    }  
  ]  
}
```

レプリケーション先のバケットではバージョニングが有効になっている必要があります。指定したロールは、レプリケーション先のバケットへの書き込みアクセス許可を持ち、Amazon S3 がそのロールを引き受けることを許可する信頼関係が必要です。

ロールのアクセス許可ポリシーの例:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetReplicationConfiguration",  
        "s3:ListBucket"  
      ],  
      "Resource": [  
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1"  
      ]  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  

```

```
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
    ],
    "Resource": [
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags"
    ],
    "Resource": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2/*"
}
]
```

信頼関係ポリシーの例:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

このコマンドでは何も出力されません。

詳細については、「Amazon Simple Storage Service コンソールユーザーガイド」の「[チュートリアル: レプリケーションの設定例](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketReplication](#)」の「」を参照してください。

put-bucket-request-payment

次の例は、put-bucket-request-payment を使用する方法を説明しています。

AWS CLI

例 1: バケットの「リクエスト支払い」設定を有効にするには

次の put-bucket-request-payment の例では、指定したバケットの requester pays を有効にします。

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"Requester"}'
```

このコマンドでは何も出力されません。

例 2: バケットの「リクエスト支払い」設定を無効にするには

次の put-bucket-request-payment の例では、指定したバケットの requester pays を無効にします。

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"BucketOwner"}'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutBucketRequestPayment](#)」の「」を参照してください。

put-bucket-tagging

次のコード例は、put-bucket-tagging を使用する方法を示しています。

AWS CLI

次のコマンドは、タグ付けの設定を my-bucket という名前のバケットに適用します。

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

ファイルは、タグを指定する現在のフォルダ内のJSONドキュメントtagging.jsonです。

```
{
  "TagSet": [
    {
      "Key": "organization",
      "Value": "marketing"
    }
  ]
}
```

または、コマンドラインから直接タグ設定を `my-bucket` に適用します。

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
'TagSet=[{Key=organization, Value=marketing}]'
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutBucketTagging](#)」の「」を参照してください。

put-bucket-versioning

次の例は、`put-bucket-versioning` を使用する方法を説明しています。

AWS CLI

次のコマンドは、`my-bucket` という名前のバケットのバージョンングを有効にします。

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled
```

次のコマンドはバージョンングを有効にします。これは、MFA コードを使用します。

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled --mfa "SERIAL 123456"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutBucketVersioning](#)」の「」を参照してください。

put-bucket-website

次のコード例は、`put-bucket-website` を使用する方法を示しています。

AWS CLI

my-bucket という名前のバケットに静的なウェブサイト設定を適用します。

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://  
website.json
```

ファイルは、ウェブサイトのインデックスページとエラーページを指定する現在のフォルダ内の JSON ドキュメント `website.json` です。

```
{  
  "IndexDocument": {  
    "Suffix": "index.html"  
  },  
  "ErrorDocument": {  
    "Key": "error.html"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutBucketWebsite](#)」の「」を参照してください。

put-object-acl

次のコード例は、`put-object-acl` を使用する方法を示しています。

AWS CLI

次のコマンド `full control` は、2 人の AWS ユーザー (`user1@example.com` と `user2@example.com`) に を付与し、すべてのユーザーにアクセス `read` 許可を付与します。

```
aws s3api put-object-acl --bucket MyBucket --key file.txt --grant-full-  
control emailaddress=user1@example.com,emailaddress=user2@example.com --grant-  
read uri=http://acs.amazonaws.com/groups/global/AllUsers
```

カスタムの詳細については、<http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> を参照してください ACLs (などの `s3api` ACL コマンドは `put-object-acl`、同じ短縮指数表記を使用します)。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutObjectAcl](#)」の「」を参照してください。

put-object-legal-hold

次のコード例は、put-object-legal-hold を使用する方法を示しています。

AWS CLI

リーガルホールドをオブジェクトに適用するには

次の put-object-legal-hold 例では、doc1.rtf という名前のバケットのオブジェクトにリーガルホールドを設定します。

```
aws s3api put-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --legal-hold Status=ON
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutObjectLegalHold](#)」の「」を参照してください。

put-object-lock-configuration

次の例は、put-object-lock-configuration を使用する方法を説明しています。

AWS CLI

バケットにオブジェクトロック設定を設定するには

次の put-object-lock-configuration 例では、指定したバケットに 50 日間のオブジェクトロックを設定します。

```
aws s3api put-object-lock-configuration \  
  --bucket my-bucket-with-object-lock \  
  --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":  
  { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutObjectLockConfiguration](#)」の「」を参照してください。

put-object-retention

次の例は、put-object-retention を使用する方法を説明しています。

AWS CLI

オブジェクトのオブジェクト保持設定を設定するには

次の put-object-retention 例では、指定されたオブジェクトのオブジェクト保持設定を 2025-01-01 まで設定します。

```
aws s3api put-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutObjectRetention](#)」の「」を参照してください。

put-object-tagging

次のコード例は、put-object-tagging を使用する方法を示しています。

AWS CLI

オブジェクトにタグを設定するには

次のput-object-tagging例では、confidential指定されたオブジェクトにキー-designationと 値を持つタグを設定します。

```
aws s3api put-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf \  
  --tagging '{"TagSet": [{"Key": "designation", "Value": "confidential"}]}'
```

このコマンドでは何も出力されません。

次のput-object-tagging例では、指定されたオブジェクトに複数のタグセットを設定します。

```
aws s3api put-object-tagging \  
  --bucket my-bucket-example \  
  --key doc3.rtf \  
  --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" },  
  { "Key": "department", "Value": "finance" }, { "Key": "team", "Value":  
  "payroll" } ]}'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutObjectTagging](#)」の「」を参照してください。

put-object

次の例は、put-object を使用する方法を説明しています。

AWS CLI

次の例は、put-object コマンドを使用して、オブジェクトを Amazon S3 にアップロードします。

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --  
body my_images.tar.bz2
```

次の例は、動画ファイルのアップロードを示しています (動画ファイルは Windows ファイルシステム構文を使用して指定します)。

```
aws s3api put-object --bucket text-content --key dir-1/big-video-file.mp4 --body e:  
\media\videos\f-sharp-3-data-services.mp4
```

オブジェクトのアップロードの詳細については、「Amazon S3 ユーザーガイド」の「オブジェクトのアップロード」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutObject](#)」の「」を参照してください。

put-public-access-block

次の例は、put-public-access-block を使用する方法を説明しています。

AWS CLI

バケットのブロックパブリックアクセス設定を設定するには

次のput-public-access-block例では、指定されたバケットの制限付きブロックパブリックアクセス設定を設定します。

```
aws s3api put-public-access-block \  
  --bucket my-bucket \  
  --public-access-block-  
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutPublicAccessBlock](#)」の「」を参照してください。

rb

次のコード例は、rb を使用する方法を示しています。

AWS CLI

例 1: バケットを削除する

次のrbコマンドはバケットを削除します。この例では、ユーザーのバケットは `mybucket` です。削除するには、バケットが空である必要があります。

```
aws s3 rb s3://mybucket
```

出力:

```
remove_bucket: mybucket
```

例 2: バケットを強制的に削除する

次のrbコマンドは、`--force`パラメータを使用して、まずバケット内のすべてのオブジェクトを削除してから、バケット自体を削除します。この例では、ユーザーのバケットは `mybucket` で、のオブジェクト `mybucket` は `test1.txt` および `test2.txt` です。

```
aws s3 rb s3://mybucket \  
  --force
```

```
--force
```

出力:

```
delete: s3://mybucket/test1.txt
delete: s3://mybucket/test2.txt
remove_bucket: mybucket
```

- API 詳細については、「コマンドリファレンス」の「[Rb](#)」を参照してください。AWS CLI

restore-object

次の例は、`restore-object` を使用する方法を説明しています。

AWS CLI

オブジェクトの復元リクエストを作成するには

次の `restore-object` の例では、指定された Amazon S3 Glacier オブジェクトをバケット `my-glacier-bucket` に 10 日間復元します。

```
aws s3api restore-object \  
  --bucket my-glacier-bucket \  
  --key doc1.rtf \  
  --restore-request Days=10
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreObject](#)」の「」を参照してください。

rm

次のコード例は、`rm` を使用する方法を示しています。

AWS CLI

例 1: S3 オブジェクトを削除する

次の `rm` コマンドは、単一の s3 オブジェクトを削除します。


```
aws s3 rm s3://mybucket/test2.txt
```

出力:

```
delete: s3://mybucket/test2.txt
```

例 2: バケット内のすべてのコンテンツを削除する

次のrmコマンドは、パラメータで渡されると、指定されたバケットとプレフィックスのすべてのオブジェクトを再帰的に削除します--recursive。この例では、バケットにオブジェクトtest1.txtとmybucketが含まれていますtest2.txt。

```
aws s3 rm s3://mybucket \  
--recursive
```

出力:

```
delete: s3://mybucket/test1.txt  
delete: s3://mybucket/test2.txt
```

例 3: ``.jpg`` ファイルを除くバケット内のすべてのコンテンツを削除する

次のrmコマンドは、パラメータを使用して一部のオブジェクトを除外--recursiveしながら、--excludeパラメータで渡されたときに、指定されたバケットとプレフィックス内のすべてのオブジェクトを再帰的に削除します。この例では、バケットにオブジェクトtest1.txtとmybucketがありますtest2.jpg。

```
aws s3 rm s3://mybucket/ \  
--recursive \  
--exclude "*.jpg"
```

出力:

```
delete: s3://mybucket/test1.txt
```

例 4: 指定されたプレフィックスの下にあるオブジェクトを除く、バケット内のすべてのコンテンツを削除する

次のrmコマンドは、パラメータを使用して特定のプレフィックスを持つすべてのオブジェクトを除外--recursiveしながら、指定されたバケットとプレフィックスを持つすべてのオブジェクトを再帰的に削除します--exclude。この例では、バケットmybucketには オブジェクトtest1.txtと がありますanother/test.txt。

```
aws s3 rm s3://mybucket/ \  
--recursive \  
--exclude "another/*"
```

出力:

```
delete: s3://mybucket/test1.txt
```

例 5: S3 アクセスポイントからオブジェクトを削除する

次のrmコマンドは、アクセスポイント (mykey) から単一のオブジェクト () を削除しますmyaccesspoint。:: 次のrmコマンドは、アクセスポイント (mykey) から単一のオブジェクト () を削除しますmyaccesspoint。

```
aws s3 rm s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

出力:

```
delete: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

• API 詳細については、「[コマンドリファレンス](#)」の「[Rm](#)」を参照してください。AWS CLI

select-object-content

次のコード例は、select-object-content を使用する方法を示しています。

AWS CLI

SQLステートメントに基づいて Amazon S3 オブジェクトの内容をフィルタリングするには

次のselect-object-content例では、指定されたSQLステートメントmy-data-file.csvでオブジェクトをフィルタリングし、出力を ファイルに送信します。

```
aws s3api select-object-content \  
--bucket mybucket \  
--key mykey \  
--sql my-data-file.csv \  
--output my-output-file.csv
```

```
--bucket my-bucket \  
--key my-data-file.csv \  
--expression "select * from s3object limit 100" \  
--expression-type 'SQL' \  
--input-serialization '{"CSV": {}, "CompressionType": "NONE"}' \  
--output-serialization '{"CSV": {}}' "output.csv"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [SelectObjectContent](#)」の「」を参照してください。

sync

次のコード例は、sync を使用する方法を示しています。

AWS CLI

例 1: すべてのローカルオブジェクトを指定されたバケットに同期する

次のsyncコマンドは、ローカルファイルを S3 にアップロードすることで、ローカルディレクトリから指定されたプレフィックスとバケットにオブジェクトを同期します。ローカルファイルのサイズが S3 オブジェクトのサイズと異なる場合、ローカルファイルの最終変更時刻が S3 オブジェクトの最終変更時刻よりも新しい場合、またはローカルファイルが指定されたバケットとプレフィックスの下に存在しない場合、ローカルファイルのアップロードが必要になります。この例では、ユーザーはバケットをローカルの現在のディレクトリmybucketに同期します。ローカルの現在のディレクトリには、ファイルtest.txtとが含まれていますtest2.txt。バケットmybucketにはオブジェクトが含まれていません。

```
aws s3 sync . s3://mybucket
```

出力:

```
upload: test.txt to s3://mybucket/test.txt  
upload: test2.txt to s3://mybucket/test2.txt
```

例 2: 指定された S3 バケットから別のバケットにすべての S3 オブジェクトを同期する

次のsyncコマンドは、S3 オブジェクトをコピーすることで、指定されたプレフィックスとバケットの下のオブジェクトを、別の指定されたプレフィックスとバケットの下のオブジェクトに

同期します。S3 オブジェクトは、2 つの S3 オブジェクトのサイズが異なる場合、ソースの最終変更時刻が送信先の最終変更時刻よりも新しい場合、または指定されたバケットとプレフィックスの送信先の下に S3 オブジェクトが存在しない場合、コピーする必要があります。

この例では、ユーザーはバケットをバケット mybucket に同期します mybucket2。バケットには、オブジェクト test.txt と mybucket が含まれています test2.txt。バケット mybucket2 にはオブジェクトが含まれていません。

```
aws s3 sync s3://mybucket s3://mybucket2
```

出力:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
copy: s3://mybucket/test2.txt to s3://mybucket2/test2.txt
```

例 3: 指定された S3 バケットからローカルディレクトリにすべての S3 オブジェクトを同期する

次の sync コマンドは、S3 オブジェクトをダウンロードして、指定された S3 バケットからローカルディレクトリにファイルを同期します。S3 オブジェクトのサイズがローカルファイルのサイズと S3 異なる場合、S3 オブジェクトの最終変更時刻がローカルファイルの最終変更時刻よりも新しい場合、または S3 オブジェクトがローカルディレクトリに存在しない場合、S3 オブジェクトのダウンロードが必要になります。オブジェクトが S3 からダウンロードされると、ローカルファイルの最終変更時刻が S3 オブジェクトの最終変更時刻に変更されることに注意してください。この例では、ユーザーはバケットを現在のローカルディレクトリ mybucket に同期します。バケットには、オブジェクト test.txt と mybucket が含まれています test2.txt。現在のローカルディレクトリにはファイルがありません。

```
aws s3 sync s3://mybucket .
```

出力:

```
download: s3://mybucket/test.txt to test.txt
download: s3://mybucket/test2.txt to test2.txt
```

例 4: すべてのローカルオブジェクトを指定されたバケットに同期し、一致しないすべてのファイルを削除する

次の sync コマンドは、ローカルファイルを S3 にアップロードすることで、指定されたプレフィックスとバケットの下にあるオブジェクトをローカルディレクトリ内のファイルと同期しま

す。--delete パラメータにより、指定されたプレフィックスとバケットの下には存在するが、ローカルディレクトリには存在しないファイルはすべて削除されます。この例では、ユーザーはバケットをローカルの現在のディレクトリmybucketに同期します。ローカルの現在のディレクトリには、ファイルtest.txtとが含まれていますtest2.txt。バケットには オブジェクトmybucketが含まれますtest3.txt。

```
aws s3 sync . s3://mybucket \  
--delete
```

出力:

```
upload: test.txt to s3://mybucket/test.txt  
upload: test2.txt to s3://mybucket/test2.txt  
delete: s3://mybucket/test3.txt
```

例 5: ``.jpg`` ファイルを除くすべてのローカルオブジェクトを指定されたバケットに同期する

次のsyncコマンドは、ローカルファイルを S3 にアップロードすることで、指定されたプレフィックスとバケットの下にあるオブジェクトをローカルディレクトリ内のファイルと同期します。--exclude パラメータにより、S3 とローカルの両方に存在するパターンに一致するすべてのファイルは同期から除外されます。この例では、ユーザーはバケットをローカルの現在のディレクトリmybucketに同期します。ローカルの現在のディレクトリには、ファイルtest.jpgとが含まれていますtest2.txt。バケットには、ローカルとは異なるtest.jpgサイズのオブジェクトmybucketが含まれていますtest.jpg。

```
aws s3 sync . s3://mybucket \  
--exclude "*.jpg"
```

出力:

```
upload: test2.txt to s3://mybucket/test2.txt
```

例 6: ``.jpg`` ファイルを除くすべてのローカルオブジェクトを指定されたバケットに同期する

次のsyncコマンドは、S3 オブジェクトをダウンロードして、ローカルディレクトリの下にあるファイルを、指定されたプレフィックスとバケットの下にあるオブジェクトに同期します。この例では、--excludeパラメータフラグを使用して、指定されたディレクトリと S3 プレフィックスを sync コマンドから除外します。この例では、ユーザーはローカルの現在

のディレクトリをバケットに同期しますmybucket。ローカルの現在のディレクトリには、ファイルtest.txtとが含まれていますanother/test2.txt。バケットには、オブジェクトanother/test5.txtと mybucketが含まれますtest1.txt。

```
aws s3 sync s3://mybucket/ . \
  --exclude "*another/*"
```

出力:

```
download: s3://mybucket/test1.txt to test1.txt
```

例 7: 異なるリージョンのバケット間ですべてのオブジェクトを同期する

次のsyncコマンドは、異なるリージョンの2つのバケット間でファイルを同期します。

```
aws s3 sync s3://my-us-west-2-bucket s3://my-us-east-1-bucket \
  --source-region us-west-2 \
  --region us-east-1
```

出力:

```
download: s3://my-us-west-2-bucket/test1.txt to s3://my-us-east-1-bucket/test1.txt
```

例 8: S3 アクセスポイントへの同期

次のsyncコマンドは、現在のディレクトリをアクセスポイント () に同期しますmyaccesspoint。

```
aws s3 sync . s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

出力:

```
upload: test.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test.txt
upload: test2.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test2.txt
```

• API 詳細については、「[コマンドリファレンス](#)」の「[同期](#)」を参照してください。AWS CLI

upload-part-copy

次の例は、upload-part-copy を使用する方法を説明しています。

AWS CLI

既存のオブジェクトからデータソースとしてデータをコピーしてオブジェクトの一部をアップロードするには

次のupload-part-copy例では、既存のオブジェクトからデータをデータソースとしてコピーして、パートをアップロードします。

```
aws s3api upload-part-copy \  
  --bucket my-bucket \  
  --key "Map_Data_June.mp4" \  
  --copy-source "my-bucket/copy_of_Map_Data_June.mp4" \  
  --part-number 1 \  
  --upload-  
id "bq0tdE1CDpWQYRPLHuNG50xAT6pA5D.m_RiBy0gg0H6b13pVRY7QjvLLf75iFdJqp_2wztk5hvpUM2SesXgrzbeh"
```

出力:

```
{  
  "CopyPartResult": {  
    "LastModified": "2019-12-13T23:16:03.000Z",  
    "ETag": "\"711470fc377698c393d94aed6305e245\""  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadPartCopy](#)」の「」を参照してください。

upload-part

次のコード例は、upload-part を使用する方法を示しています。

AWS CLI

次のコマンドは、create-multipart-upload コマンドで開始されたマルチパートアップロードの最初の部分をアップロードします。

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --  
body part01 --upload-id  
"dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6YjlX.cph0gtNBtJ8P3UR
```

body オプションは、アップロードするローカルファイルの名前またはパスを指定します (file://プレフィックスは使用しないでください)。各パートの最小サイズは 5 MB です。アップロード ID は create-multipart-upload によって返され、list-multipart-uploads を使用して取得することもできます。バケットとキーは、マルチパートアップロードの作成時に指定されます。

出力:

```
{  
  "ETag": "\"e868e0f4719e394144ef36531ee6824c\""  
}
```

各パートのETag値を後で使用するために保存します。これらはマルチパートアップロードを完了するために必要です。

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadPart](#)」の「」を参照してください。

website

次の例は、website を使用する方法を説明しています。

AWS CLI

S3 バケットを静的ウェブサイトとして設定する

次のコマンドは、という名前のバケットを静的ウェブサイトmy-bucketとして設定します。インデックスドキュメントオプションは、訪問者my-bucketがウェブサイト に移動するときにリダイレクトされる のファイルを指定しますURL。この場合、バケットは us-west-2 リージョンにあるため、サイトは に表示されますhttp://my-bucket.s3-website-us-west-2.amazonaws.com。

静的サイトに表示されるバケット内のすべてのファイルは、訪問者が開くことができるように設定する必要があります。ファイルアクセス許可は、バケットウェブサイトの設定とは別に設定されます。


```
aws s3 website s3://my-bucket/ \  
--index-document index.html \  
--error-document error.html
```

Amazon S3 で静的ウェブサイトをホストする方法については、「Amazon Simple Storage Service デベロッパーガイド」の「[静的ウェブサイトをホストする](#)」を参照してください。

- API 詳細については、「コマンドリファレンス」の「[ウェブサイト](#)」を参照してください。

AWS CLI

を使用した Amazon S3 Control の例 AWS CLI

次のコード例は、Amazon S3 Control AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-access-point

次の例は、create-access-point を使用する方法を説明しています。

AWS CLI

アクセスポイントを作成するには

次のcreate-access-point例では、アカウント 123456789012 business-recordsのバケットfinance-apに という名前のアクセスポイントを作成します。この例を実行する前に、アクセスポイント名、バケット名、およびアカウント番号をユースケースに適した値に置き換えます。

```
aws s3control create-access-point \  
--bucket finance-ap \  
--access-point-name finance-ap
```

```
--account-id 123456789012 \  
--bucket business-records \  
--name finance-ap
```

このコマンドでは何も出力されません。

詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「[アクセスポイントの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAccessPoint](#)」の「」を参照してください。

create-job

次のコード例は、create-job を使用する方法を示しています。

AWS CLI

Amazon S3 バッチオペレーションジョブを作成するには

次のcreate-job例では、オブジェクトをとしてタグ付けする Amazon S3 バッチオペレーションジョブを作成しますconfidential` in the bucket ``employee-records。

```
aws s3control create-job \  
  --account-id 123456789012 \  
  --operation '{"S3PutObjectTagging": { "TagSet": [{"Key":"confidential",  
"Value":"true"}] }}' \  
  --report '{"Bucket":"arn:aws:s3:::employee-records-logs","Prefix":"batch-op-  
create-job",  
"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \  
  --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":  
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::employee-records-logs/inv-  
report/7a6a9be4-072c-407e-85a2-  
ec3e982f773e.csv","ETag":"69f52a4e9f797e987155d9c8f5880897"}}' \  
  --priority 42 \  
  --role-arn arn:aws:iam::123456789012:role/S3BatchJobRole
```

出力:

```
{  
  "JobId": "93735294-df46-44d5-8638-6356f335324e"
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateJob](#)」の「」を参照してください。

delete-access-point-policy

次のコード例は、delete-access-point-policy を使用する方法を示しています。

AWS CLI

アクセスポイントポリシーを削除するには

次のdelete-access-point-policy例では、アカウント 123456789012 finance-apで という名前のアクセスポイントからアクセスポイントポリシーを削除します。この例を実行する前に、アクセスポイント名とアカウント番号をユースケースに適した値に置き換えます。

```
aws s3control delete-access-point-policy \  
  --account-id 123456789012 \  
  --name finance-ap
```

このコマンドでは何も出力されません。

詳細については、[Amazon S3 アクセスポイントによるデータアクセスの管理](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccessPointPolicy](#)」の「」を参照してください。

delete-access-point

次の例は、delete-access-point を使用する方法を説明しています。

AWS CLI

アクセスポイントを削除するには

次のdelete-access-point例では、アカウント 123456789012 finance-apで という名前のアクセスポイントを削除します。この例を実行する前に、アクセスポイント名とアカウント番号をユースケースに適した値に置き換えます。

```
aws s3control delete-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

このコマンドでは何も出力されません。

詳細については、[Amazon S3 アクセスポイントによるデータアクセスの管理](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccessPoint](#)」の「」を参照してください。

delete-public-access-block

次のコード例は、delete-public-access-block を使用する方法を示しています。

AWS CLI

アカウントのブロックパブリックアクセス設定を削除するには

次のdelete-public-access-block例では、指定されたアカウントのブロックパブリックアクセス設定を削除します。

```
aws s3control delete-public-access-block \  
  --account-id 123456789012
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePublicAccessBlock](#)」の「」を参照してください。

describe-job

次の例は、describe-job を使用する方法を説明しています。

AWS CLI

Amazon S3 バッチオペレーションジョブを記述するには

以下にdescribe-job、指定されたバッチオペレーションジョブの設定パラメータとステータスを示します。

```
aws s3control describe-job \  
--account-id 123456789012 \  
--job-id 93735294-df46-44d5-8638-6356f335324e
```

出力:

```
{  
  "Job": {  
    "TerminationDate": "2019-10-03T21:49:53.944Z",  
    "JobId": "93735294-df46-44d5-8638-6356f335324e",  
    "FailureReasons": [],  
    "Manifest": {  
      "Spec": {  
        "Fields": [  
          "Bucket",  
          "Key"  
        ],  
        "Format": "S3BatchOperations_CSV_20180820"  
      },  
      "Location": {  
        "ETag": "69f52a4e9f797e987155d9c8f5880897",  
        "ObjectArn": "arn:aws:s3:::employee-records-logs/inv-report/7a6a9be4-072c-407e-85a2-ec3e982f773e.csv"  
      }  
    },  
    "Operation": {  
      "S3PutObjectTagging": {  
        "TagSet": [  
          {  
            "Value": "true",  
            "Key": "confidential"  
          }  
        ]  
      }  
    },  
    "RoleArn": "arn:aws:iam::123456789012:role/S3BatchJobRole",  
    "ProgressSummary": {  
      "TotalNumberOfTasks": 8,  
      "NumberOfTasksFailed": 0,  
      "NumberOfTasksSucceeded": 8  
    },  
    "Priority": 42,  
    "Report": {
```

```
        "ReportScope": "AllTasks",
        "Format": "Report_CSV_20180820",
        "Enabled": true,
        "Prefix": "batch-op-create-job",
        "Bucket": "arn:aws:s3:::employee-records-logs"
    },
    "JobArn": "arn:aws:s3:us-west-2:123456789012:job/93735294-
df46-44d5-8638-6356f335324e",
    "CreationTime": "2019-10-03T21:48:48.048Z",
    "Status": "Complete"
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeJob](#)」の「」を参照してください。

get-access-point-policy-status

次の例は、get-access-point-policy-status を使用する方法を説明しています。

AWS CLI

アクセスポイントポリシーのステータスを取得するには

次のget-access-point-policy-status例では、アカウント 123456789012 finance-ap という名前のアクセスポイントのアクセスポイントポリシーステータスを取得します。アクセスポイントポリシーのステータスは、アクセスポイントのポリシーがパブリックアクセスを許可するかどうかを示します。この例を実行する前に、アクセスポイント名とアカウント番号をユースケースに適した値に置き換えます。

```
aws s3control get-access-point-policy-status \
  --account-id 123456789012 \
  --name finance-ap
```

出力:

```
{
  "PolicyStatus": {
    "IsPublic": false
  }
}
```

アクセスポイントポリシーがパブリックと見なされるタイミングの詳細については、「[Amazon Simple Storage Service デベロッパーガイド](#)」の「[Public](#)」の意味を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAccessPointPolicyStatus](#)」の「」を参照してください。

get-access-point-policy

次のコード例は、get-access-point-policy を使用する方法を示しています。

AWS CLI

アクセスポイントポリシーを取得するには

次のget-access-point-policy例では、アカウント 123456789012 finance-apの という名前のアクセスポイントからアクセスポイントポリシーを取得します。この例を実行する前に、アクセスポイント名とアカウント番号をユースケースに適した値に置き換えます。

```
aws s3control get-access-point-policy \  
  --account-id 123456789012 \  
  --name finance-ap
```

出力:

```
{  
  "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",  
    \"Principal\":{\"AWS\":\"arn:aws:iam:123456789012:role/Admin\"},\"Action\":  
    \"s3:GetObject\",\"Resource\":\"arn:aws:s3:us-west-2:123456789012:accesspoint/  
    finance-ap/object/records/*\"}]}"  
}
```

詳細については、[Amazon S3 アクセスポイントによるデータアクセスの管理](#)」を参照してください。

- API 詳細については、「コマンドリファレンス[GetAccessPointPolicy](#)」の「」を参照してください。AWS CLI

get-access-point

次のコード例は、get-access-point を使用する方法を示しています。

AWS CLI

アクセスポイント設定の詳細を取得するには

次のget-access-point例では、アカウント 123456789012 finance-apで という名前のアクセスポイントの設定の詳細を取得します。この例を実行する前に、アクセスポイント名とアカウント番号をユースケースに適した値に置き換えます。

```
aws s3control get-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

出力:

```
{  
  "Name": "finance-ap",  
  "Bucket": "business-records",  
  "NetworkOrigin": "Internet",  
  "PublicAccessBlockConfiguration": {  
    "BlockPublicAcls": false,  
    "IgnorePublicAcls": false,  
    "BlockPublicPolicy": false,  
    "RestrictPublicBuckets": false  
  },  
  "CreationDate": "2020-01-01T00:00:00Z"  
}
```

詳細については、[Amazon S3 アクセスポイントによるデータアクセスの管理](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAccessPoint](#)」の「」を参照してください。

get-multi-region-access-point-routes

次のコード例は、get-multi-region-access-point-routes を使用する方法を示しています。

AWS CLI

現在のマルチリージョンアクセスポイントルート設定をクエリするには

次のget-multi-region-access-point-routes例では、指定されたマルチリージョンアクセスポイントの現在のルーティング設定を返します。

```
aws s3control get-multi-region-access-point-routes \  
  --region Region \  
  --account-id 111122223333 \  
  --mrap MultiRegionAccessPoint_ARN
```

出力:

```
{  
  "Mrap": "arn:aws:s3::111122223333:accesspoint/0000000000000000.mrap",  
  "Routes": [  
    {  
      "Bucket": "DOC-EXAMPLE-BUCKET-1",  
      "Region": "ap-southeast-2",  
      "TrafficDialPercentage": 100  
    },  
    {  
      "Bucket": "DOC-EXAMPLE-BUCKET-2",  
      "Region": "us-west-1",  
      "TrafficDialPercentage": 0  
    }  
  ]  
}
```

- API 詳細については、「コマンドリファレンス[GetMultiRegionAccessPointRoutes](#)」の「」を参照してください。AWS CLI

get-public-access-block

次の例は、get-public-access-block を使用する方法を説明しています。

AWS CLI

アカウントのパブリックブロックアクセス設定を一覧表示するには

次のget-public-access-block例では、指定されたアカウントのパブリックアクセスブロック設定を表示します。

```
aws s3control get-public-access-block \  
  --account-id 111122223333 \  
  --region Region \  
  --public-access-block PublicAccessBlock_ARN
```

```
--account-id 123456789012
```

出力:

```
{
  "PublicAccessBlockConfiguration": {
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true,
    "IgnorePublicAcls": true,
    "BlockPublicAcls": true
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPublicAccessBlock](#)」の「」を参照してください。

list-access-points

次の例は、list-access-points を使用する方法を説明しています。

AWS CLI

例 1: アカウントのすべてのアクセスポイントのリストを取得するには

次のlist-access-points例では、アカウント 123456789012 が所有するバケットにアタッチされたすべてのアクセスポイントのリストを表示します。

```
aws s3control list-access-points \
  --account-id 123456789012
```

出力:

```
{
  "AccessPointList": [
    {
      "Name": "finance-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "business-records"
    },
    {
```

```

        "Name": "managers-ap",
        "NetworkOrigin": "Internet",
        "Bucket": "business-records"
    },
    {
        "Name": "private-network-ap",
        "NetworkOrigin": "VPC",
        "VpcConfiguration": {
            "VpcId": "1a2b3c"
        },
        "Bucket": "business-records"
    },
    {
        "Name": "customer-ap",
        "NetworkOrigin": "Internet",
        "Bucket": "external-docs"
    },
    {
        "Name": "public-ap",
        "NetworkOrigin": "Internet",
        "Bucket": "external-docs"
    }
}
]
}

```

例 2: バケットのすべてのアクセスポイントのリストを取得するには

次のlist-access-points例では、アカウント 123456789012 external-docsが所有するバケットにアタッチされたすべてのアクセスポイントのリストを取得します。

```

aws s3control list-access-points \
  --account-id 123456789012 \
  --bucket external-docs

```

出力:

```

{
  "AccessPointList": [
    {
      "Name": "customer-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "external-docs"
    },

```

```
{
  "Name": "public-ap",
  "NetworkOrigin": "Internet",
  "Bucket": "external-docs"
}
]
```

詳細については、[Amazon S3 アクセスポイントによるデータアクセスの管理](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccessPoints](#)」の「」を参照してください。

list-jobs

次のコード例は、list-jobs を使用する方法を示しています。

AWS CLI

アカウント Amazon S3 バッチオペレーションジョブを一覧表示するには

次のlist-jobs例では、指定されたアカウントの最近のバッチオペレーションジョブをすべて一覧表示します。

```
aws s3control list-jobs \
  --account-id 123456789012
```

出力:

```
{
  "Jobs": [
    {
      "Operation": "S3PutObjectTagging",
      "ProgressSummary": {
        "NumberOfTasksFailed": 0,
        "NumberOfTasksSucceeded": 8,
        "TotalNumberOfTasks": 8
      },
      "CreationTime": "2019-10-03T21:48:48.048Z",
      "Status": "Complete",
    }
  ]
}
```

```

    "JobId": "93735294-df46-44d5-8638-6356f335324e",
    "Priority": 42
  },
  {
    "Operation": "S3PutObjectTagging",
    "ProgressSummary": {
      "NumberOfTasksFailed": 0,
      "NumberOfTasksSucceeded": 0,
      "TotalNumberOfTasks": 0
    },
    "CreationTime": "2019-10-03T21:46:07.084Z",
    "Status": "Failed",
    "JobId": "3f3c7619-02d3-4779-97f6-1d98dd313108",
    "Priority": 42
  },
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobs](#)」の「」を参照してください。

put-access-point-policy

次の例は、put-access-point-policy を使用する方法を説明しています。

AWS CLI

アクセスポイントポリシーを設定するには

次のput-access-point-policy例では、アクセスポイントの指定されたアクセスポイントポリシーをアカウント 123456789012 に配置しfinance-apます。アクセスポイントにfinance-ap既にポリシーがある場合、このコマンドは既存のポリシーをこのコマンドで指定されたポリシーに置き換えます。この例を実行する前に、アカウント番号、アクセスポイント名、ポリシーステートメントをユースケースに適した値に置き換えます。

```

aws s3control put-access-point-policy \
  --account-id 123456789012 \
  --name finance-ap \
  --policy file://ap-policy.json

```

ap-policy.json の内容:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Alice"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/finance-ap/
object/Alice/*"
    }
  ]
}
```

このコマンドでは何も出力されません。

詳細については、[Amazon S3 アクセスポイントによるデータアクセスの管理](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutAccessPointPolicy](#)」の「」を参照してください。

put-public-access-block

次の例は、put-public-access-block を使用する方法を説明しています。

AWS CLI

アカウントのブロックパブリックアクセス設定を編集するには

次のput-public-access-block例では、指定されたアカウントのすべてのブロックパブリックアクセス設定を に切り替えtrueます。

```
aws s3control put-public-access-block \
  --account-id 123456789012 \
  --public-access-block-configuration '{"BlockPublicAcls": true,
  "IgnorePublicAcls": true, "BlockPublicPolicy": true, "RestrictPublicBuckets":
  true}'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutPublicAccessBlock](#)」の「」を参照してください。

submit-multi-region-access-point-routes

次の例は、submit-multi-region-access-point-routes を使用する方法を説明しています。

AWS CLI

マルチリージョンアクセスポイントのルーティング設定を更新するには

次のsubmit-multi-region-access-point-routes例では、マルチリージョンアクセスポイントの ap-southeast-2 リージョンDOC-EXAMPLE-BUCKET-2の DOC-EXAMPLE-BUCKET-1および のルーティングステータスを更新します。

```
aws s3control submit-multi-region-access-point-routes \  
  --region ap-southeast-2 \  
  --account-id 111122223333 \  
  --mrp MultiRegionAccessPoint_ARN \  
  --route-updates Bucket=DOC-EXAMPLE-  
BUCKET-1,TrafficDialPercentage=100 Bucket=DOC-EXAMPLE-  
BUCKET-2,TrafficDialPercentage=0
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SubmitMultiRegionAccessPointRoutes](#)」の「」を参照してください。

update-job-priority

次のコード例は、update-job-priority を使用する方法を示しています。

AWS CLI

Amazon S3 バッチオペレーションジョブのジョブ優先度を更新するには

次のupdate-job-priority例では、指定されたジョブを新しい優先度に更新します。

```
aws s3control update-job-priority \  
  --account-id 123456789012 \  
  --job-id 123456789012 \  
  --priority High
```

```
--job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
--priority 52
```

出力:

```
{  
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386",  
  "Priority": 52  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateJobPriority](#)」の「」を参照してください。

update-job-status

次のコード例は、update-job-status を使用する方法を示しています。

AWS CLI

Amazon S3 バッチオペレーションジョブのステータスを更新するには

次のupdate-job-status例では、承認待ちの指定されたジョブをキャンセルします。

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
  --requested-job-status Cancelled
```

出力:

```
{  
  "Status": "Cancelled",  
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386"  
}
```

次のupdate-job-status例では、承認待ちの指定された を確認して実行します。

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \  
  --requested-job-status Cancelled
```



```
--requested-job-status Ready
```

Output::

```
{  
  "Status": "Ready",  
  "JobId": "5782949f-3301-4fb3-be34-8d5bab54dbca"  
}
```

次のupdate-job-status例では、実行中の指定されたジョブをキャンセルします。

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \  
  --requested-job-status Cancelled
```

Output::

```
{  
  "Status": "Cancelling",  
  "JobId": "5782949f-3301-4fb3-be34-8d5bab54dbca"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateJobStatus](#)」の「」を参照してください。

を使用した S3 Glacier の例 AWS CLI

次のコード例は、S3 Glacier AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

abort-multipart-upload

次のコード例は、abort-multipart-upload を使用する方法を示しています。

AWS CLI

次のコマンドは、 という名前のボールドへの進行中のマルチパートアップロードを削除しますmy-vault。

```
aws glacier abort-multipart-upload --account-id - --vault-name my-vault
--upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_LR7vgFuJV6NtcV5zpsJ
```

このコマンドでは、出力が生成されません。Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。アップロード ID は aws glacier initiate-multipart-upload コマンドによって返され、aws glacier list-multipart-uploads を使用して取得することもできます。

を使用した Amazon Glacier へのマルチパートアップロードの詳細については AWS CLI、AWS CLI ユーザーガイドの Amazon Glacier の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AbortMultipartUpload](#)」の「」を参照してください。

abort-vault-lock

次の例は、abort-vault-lock を使用する方法を説明しています。

AWS CLI

進行中のボールドロックプロセスを中止するには

次のabort-vault-lock例では、指定されたボールドからボールドロックポリシーを削除し、ボールドロックのロック状態をロック解除にリセットします。

```
aws glacier abort-vault-lock \
--account-id - \
--vault-name MyVaultName
```

このコマンドでは何も出力されません。

詳細については、Amazon Glacier APIデベロッパーガイド」の「[ボールドロックの中止 \(DELETE ロックポリシー\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AbortVaultLock](#)」の「」を参照してください。

add-tags-to-vault

次のコード例は、add-tags-to-vault を使用する方法を示しています。

AWS CLI

次のコマンドは、my-vault という名前のボールドに 2 つのタグを追加します。

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --  
tags id=1234,date=july2015
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTagsToVault](#)」の「」を参照してください。

complete-multipart-upload

次の例は、complete-multipart-upload を使用する方法を説明しています。

AWS CLI

次のコマンドは、3 MiB アーカイブのマルチパートアップロードを完了します。

```
aws glacier complete-multipart-upload --archive-size 3145728 --  
checksum 9628195fcdcbbe76cdde456d4646fa7de5f219fb39823836d81f0cc0e18aa67  
--upload-id 19gaRezEXAMPLES6Ry5YYdqthH0C_kGRCT03L9yetr220UmPtBYKk-  
0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ --account-id - --vault-name my-vault
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

アップロード ID は `aws glacier initiate-multipart-upload` コマンドによって返され、`aws glacier list-multipart-uploads` を使用して取得することもできます。チェックサムパラメータは、アーカイブの SHA-256 ツリーハッシュを 16 進数で取得します。

を使用した Amazon Glacier へのマルチパートアップロードの詳細については AWS CLI、AWS CLI ユーザーガイドの Amazon Glacier の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CompleteMultipartUpload](#)」の「」を参照してください。

complete-vault-lock

次のコード例は、`complete-vault-lock` を使用する方法を示しています。

AWS CLI

進行中のボールドロックプロセスを完了するには

次の `complete-vault-lock` 例では、指定されたボールドの進行中のロックの進行状況を完了し、ボールドロックのロック状態を に設定します Locked。を実行すると、`lock-id` パラメータの値を取得します `initiate-lock-process`。

```
aws glacier complete-vault-lock \  
  --account-id - \  
  --vault-name MyVaultName \  
  --lock-id 9QZgEXAMPLEPhvL6xEXAMPLE
```

このコマンドでは何も出力されません。

詳細については、Amazon Glacier API デベロッパーガイド」の [「完全なボールドロック \(POST lockId\)」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CompleteVaultLock](#)」の「」を参照してください。

create-vault

次の例は、`create-vault` を使用する方法を説明しています。

AWS CLI

次のコマンドでは、`my-vault` という名前の新しいボールドが作成されます。

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVault](#)」の「」を参照してください。

delete-archive

次の例は、delete-archive を使用する方法を説明しています。

AWS CLI

ボールドからアーカイブを削除するには

次の delete-archive の例では、example_vault から指定されたアーカイブを削除します。

```
aws glacier delete-archive \  
  --account-id 111122223333 \  
  --vault-name example_vault \  
  --archive-id Sc0u9ZP8yaWkmh-XGLIvAVprtLhaLCGnNwNl5I5x9HqPIkX5mjc0DrId3Ln-Gi_k2HzmlIDZUz117KSdVMdMXLuFwi9PJUitxw073edQ43eTlMwKH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteArchive](#)」の「」を参照してください。

delete-vault-access-policy

次のコード例は、delete-vault-access-policy を使用する方法を示しています。

AWS CLI

ボールドのアクセスポリシーを削除するには

次のdelete-vault-access-policy例では、指定されたボールドのアクセスポリシーを削除します。

```
aws glacier delete-vault-access-policy \  
  --account-id 111122223333 \  
  --vault-name example_vault \  
  --policy-name example_policy
```

```
--vault-name example_vault
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVaultAccessPolicy](#)」の「」を参照してください。

delete-vault-notifications

次のコード例は、delete-vault-notifications を使用する方法を示しています。

AWS CLI

ポールトSNSの通知を削除するには

次のdelete-vault-notifications例では、Amazon Simple Notification Service (Amazon SNS) によって指定されたポールトに対して送信された通知を削除します。

```
aws glacier delete-vault-notifications \  
  --account-id 111122223333 \  
  --vault-name example_vault
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVaultNotifications](#)」の「」を参照してください。

delete-vault

次のコード例は、delete-vault を使用する方法を示しています。

AWS CLI

次のコマンドでは、my-vault という名前のポールトが削除されます。

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

このコマンドでは、出力が生成されません。Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVault](#)」の「」を参照してください。

describe-job

次の例は、describe-job を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-vault という名前のボールドでのインベントリ取得ジョブに関する情報を取得します。

```
aws glacier describe-job --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW
```

出力:

```
{
  "InventoryRetrievalParameters": {
    "Format": "JSON"
  },
  "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
  "Completed": false,
  "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",
  "Action": "InventoryRetrieval",
  "CreationDate": "2015-07-17T20:23:41.616Z",
  "StatusCode": "InProgress"
}
```

ジョブ ID は、aws glacier initiate-job と aws glacier list-jobs の出力にあります。Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeJob](#)」の「」を参照してください。

describe-vault

次の例は、describe-vault を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-vault という名前のボールドに関するデータを取得します。

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeVault](#)」の「」を参照してください。

get-data-retrieval-policy

次のコード例は、`get-data-retrieval-policy` を使用する方法を示しています。

AWS CLI

次のコマンドは、使用中アカウントのデータ取得ポリシーを取得します。

```
aws glacier get-data-retrieval-policy --account-id -
```

出力:

```
{
  "Policy": {
    "Rules": [
      {
        "BytesPerHour": 10737418240,
        "Strategy": "BytesPerHour"
      }
    ]
  }
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDataRetrievalPolicy](#)」の「」を参照してください。

get-job-output

次のコード例は、`get-job-output` を使用する方法を示しています。

AWS CLI

次のコマンドは、ポールトインベントリジョブの出力を、現在のディレクトリの `output.json` という名前のファイルに保存します。

```
aws glacier get-job-output --account-id - --vault-name my-  
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RlOGduS7Eg-  
R047Yc6FxsDGBgf_Q2DK5Ejh18CnTS5XW4_XqLNHS61ds04CnMW output.json
```

`job-id` は、`aws glacier list-jobs` の出力にあります。出力ファイル名はオプション名の前に付いていない位置引数であることに注意してください。Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

出力:

```
{  
  "status": 200,  
  "acceptRanges": "bytes",  
  "contentType": "application/json"  
}
```

`output.json`:

```
{"VaultARN":"arn:aws:glacier:us-west-2:0123456789012:vaults/  
my-vault","InventoryDate":"2015-04-07T00:26:18Z","ArchiveList":  
[{"ArchiveId":"kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGEIWQX-  
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-  
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw","ArchiveDescription":"multipart  
upload  
test","CreationDate":"2015-04-06T22:24:34Z","Size":3145728,"SHA256TreeHash":"9628195fcdbcbcb
```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetJobOutput](#)」の「」を参照してください。

get-vault-access-policy

次の例は、`get-vault-access-policy` を使用する方法を説明しています。

AWS CLI

ポールのアクセスポリシーを取得するには

次のget-vault-access-policy例では、指定されたポールのアクセスポリシーを取得します。

```
aws glacier get-vault-access-policy \  
  --account-id 111122223333 \  
  --vault-name example_vault
```

出力:

```
{  
  "policy": {  
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\": [{  
      \"Effect\": \"Allow\", \"Principal\": {  
        \"AWS\": \"arn:aws:iam:444455556666:root\"  
      }, \"Action\": \"glacier:ListJobs\", \"Resource\":  
        \"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"  
      }, {  
        \"Effect\": \"Allow\", \"Principal\": {  
          \"AWS\": \"arn:aws:iam:444455556666:root\"  
        }, \"Action\": \"glacier:UploadArchive\",  
        \"Resource\": \"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"  
      }  
    ]}"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVaultAccessPolicy](#)」の「」を参照してください。

get-vault-lock

次のコード例は、get-vault-lock を使用する方法を示しています。

AWS CLI

ポールロックの詳細を取得するには

次のget-vault-lock例では、指定されたポールのロックに関する詳細を取得しました。

```
aws glacier get-vault-lock \  
  --account-id - \  
  --vault-name MyVaultName
```

出力:

```
{
  "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Sid\": \"Define-vault-lock\", \"Effect\": \"Deny\", \"Principal\": { \"AWS\": \"arn:aws:iam:999999999999:root\" }, \"Action\": \"glacier:DeleteArchive\", \"Resource\": \"arn:aws:glacier:us-west-2:999999999999:vaults/MyVaultName\", \"Condition\": { \"NumericLessThanEquals\": { \"glacier:ArchiveAgeinDays\": \"365\" } } } ] }\",
  \"State\": \"Locked\",
  \"CreationDate\": \"2019-07-29T22:25:28.640Z\"
}
```

詳細については、Amazon Glacier API デベロッパーガイドの「[ボルトロックの取得 \(GET ロックポリシー\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetVaultLock](#)」の「」を参照してください。

get-vault-notifications

次の例は、get-vault-notifications を使用する方法を説明しています。

AWS CLI

次のコマンドは、my-vault という名前のボルトの通知設定の説明を取得します。

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

出力:

```
{
  "vaultNotificationConfig": {
    "Events": [
      "InventoryRetrievalCompleted",
      "ArchiveRetrievalCompleted"
    ],
    "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
  }
}
```

ボルトに通知が設定されていない場合、エラーが返されます。Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVaultNotifications](#)」の「」を参照してください。

initiate-job

次のコード例は、initiate-job を使用する方法を示しています。

AWS CLI

次のコマンドは、ボールド のインベントリを取得するジョブを開始しますmy-vault。

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters  
'{"Type": "inventory-retrieval"}'
```

出力:

```
{  
  "location": "/0123456789012/vaults/my-vault/jobs/  
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-  
R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",  
  "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-  
R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW"  
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

次のコマンドは、ボールド からアーカイブを取得するジョブを開始しますmy-vault。

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-  
parameters file://job-archive-retrieval.json
```

job-archive-retrieval.json は、ジョブのタイプ、アーカイブ ID、およびいくつかのオプションパラメータを指定するローカルフォルダ内のJSONファイルです。

```
{  
  "Type": "archive-retrieval",  
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGEIWQX-  
ybtRDvc2VkJPSDtfKmQrj0IRQLSGsNuDp-  
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
```

```
"Description": "Retrieve archive on 2015-07-17",
"SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

アーカイブIDsは、`aws glacier upload-archive`および の出力で使えます `aws glacier get-job-output`。

出力:

```
{
  "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk205uLYaFdAY0i-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav",
  "jobId": "l7IL5-EkXy205uLYaFdAY0iEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav"
}
```

ジョブパラメータ形式の詳細については、「Amazon Glacier APIリファレンス」の「ジョブの開始」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[InitiateJob](#)」の「」を参照してください。

initiate-multipart-upload

次のコード例は、`initiate-multipart-upload` を使用する方法を示しています。

AWS CLI

次のコマンドは、ファイルごとに 1 MiB (1024 x 1024 バイト) のパートサイズ `my-vault` という名前のボールドへのマルチパートアップロードを開始します。

```
aws glacier initiate-multipart-upload --account-id - --part-size 1048576 --vault-
name my-vault --archive-description "multipart upload test"
```

アーカイブの説明パラメータはオプションです。Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

このコマンドは、成功するとアップロード ID を出力します。でアーカイブの各部分をアップロードするときは、アップロード ID を使します `aws glacier upload-multipart-part`。

を使用した Amazon Glacier へのマルチパートアップロードの詳細については AWS CLI、AWS CLI ユーザーガイドの Amazon Glacier の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [InitiateMultipartUpload](#)」の「」を参照してください。

initiate-vault-lock

次の例は、initiate-vault-lock を使用する方法を説明しています。

AWS CLI

ポールトロックプロセスを開始するには

次の initiate-vault-lock 例では、指定されたポールトにポールトロックポリシーをインストールし、ポールトロックのロック状態を に設定します InProgress。ポールトロックの状態を に設定するには、24 complete-vault-lock 時間以内に を呼び出してプロセスを完了する必要があります Locked。

```
aws glacier initiate-vault-lock \  
  --account-id - \  
  --vault-name MyVaultName \  
  --policy file://vault_lock_policy.json
```

vault_lock_policy.json の内容:

```
{"Policy": "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Sid\": \"Define-vault-lock\", \"Effect\": \"Deny\", \"Principal\": {\"AWS\": \"arn:aws:iam:999999999999:root\"}, \"Action\": \"glacier:DeleteArchive\", \"Resource\": \"arn:aws:glacier:us-west-2:999999999999:vaults/examplevault\", \"Condition\": {\"NumericLessThanEquals\": {\"glacier:ArchiveAgeInDays\": \"365\"}}}]}"}
```

出力は、ポールトロックプロセスを完了するために使用できるポールトロック ID です。

```
{  
  "lockId": "9QZgEXAMPLEPhvL6xEXAMPLE"  
}
```

詳細については、[「Amazon Glacier デベロッパーガイド」](#)の「[ポールトロックの開始 \(POST ロックポリシー\)](#)」を参照してください。 Amazon Glacier API

- API 詳細については、AWS CLI 「コマンドリファレンス [InitiateVaultLock](#)」の「」を参照してください。

list-jobs

次のコード例は、list-jobs を使用する方法を示しています。

AWS CLI

次のコマンドは、my-vault という名前のボルトで進行中のジョブと最近完了したジョブを一覧表示します。

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

出力:

```
{
  "JobList": [
    {
      "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
      "RetrievalByteRange": "0-3145727",
      "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
      "Completed": false,
      "SHA256TreeHash":
"9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
      "JobId": "17IL5-EkXyEY9Ws95fClzIbk205uLYaFdAY0i-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav",
      "ArchiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkpSDtFkMqrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUMzWkbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
      "JobDescription": "Retrieve archive on 2015-07-17",
      "ArchiveSizeInBytes": 3145728,
      "Action": "ArchiveRetrieval",
      "ArchiveSHA256TreeHash":
"9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
      "CreationDate": "2015-07-17T21:16:13.840Z",
      "StatusCode": "InProgress"
    },
    {
      "InventoryRetrievalParameters": {
        "Format": "JSON"
      }
    }
  ],
}
```

```
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "Completed": false,
    "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
R047Yc6FxsdBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61ds04CnMW",
    "Action": "InventoryRetrieval",
    "CreationDate": "2015-07-17T20:23:41.616Z",
    "StatusCode": ""InProgress""
  }
]
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobs](#)」の「」を参照してください。

list-multipart-uploads

次のコード例は、list-multipart-uploads を使用する方法を示しています。

AWS CLI

次のコマンドは、という名前のボールドに対して進行中のすべてのマルチパートアップロードを示していますmy-vault。

```
aws glacier list-multipart-uploads --account-id - --vault-name my-vault
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

を使用した Amazon Glacier へのマルチパートアップロードの詳細については AWS CLI、AWS CLI ユーザーガイドのAmazon Glacier の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMultipartUploads](#)」の「」を参照してください。

list-parts

次の例は、list-parts を使用する方法を説明しています。

AWS CLI

次のコマンドは、という名前のボールドへのマルチパートアップロード用にアップロードされたパートを一覧表示しますmy-vault。

```
aws glacier list-parts --account-id - --vault-name my-vault --upload-id "SYZi7qnL-YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdιCs8ut-D"
```

出力:

```
{
  "MultipartUploadId": "SYZi7qnL-
YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdιCs8ut-
D",
  "Parts": [
    {
      "RangeInBytes": "0-1048575",
      "SHA256TreeHash":
"e1f2a7cd6e047350f69b9f8cfa60fa606fe2f02802097a9a026360a7edc1f553"
    },
    {
      "RangeInBytes": "1048576-2097151",
      "SHA256TreeHash":
"43cf3061fb95796aed99a11a6aa3cd8f839eed15e655ab0a597126210636aee6"
    }
  ],
  "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
  "CreationDate": "2015-07-18T00:05:23.830Z",
  "PartSizeInBytes": 1048576
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

を使用した Amazon Glacier へのマルチパートアップロードの詳細については AWS CLI、AWS CLI ユーザーガイドの Amazon Glacier の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListParts](#)」の「」を参照してください。

list-provisioned-capacity

次の例は、list-provisioned-capacity を使用する方法を説明しています。

AWS CLI

プロビジョニングされたキャパシティーユニットを取得するには

次のlist-provisioned-capacity例では、指定されたアカウントのプロビジョニングされたキャパシティーユニットの詳細を取得します。

```
aws glacier list-provisioned-capacity \  
  --account-id 111122223333
```

出力:

```
{  
  "ProvisionedCapacityList": [  
    {  
      "CapacityId": "HpASAUvfRFiVDb0jMfEIcr8K",  
      "ExpirationDate": "2020-03-18T19:59:24.000Z",  
      "StartDate": "2020-02-18T19:59:24.912Z"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProvisionedCapacity](#)」の「」を参照してください。

list-tags-for-vault

次のコード例は、list-tags-for-vault を使用する方法を示しています。

AWS CLI

次のコマンドは、my-vault という名前のボールドに適用されたタグを一覧表示します。

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

出力:

```
{
  "Tags": {
    "date": "july2015",
    "id": "1234"
  }
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForVault](#)」の「」を参照してください。

list-vaults

次の例は、list-vaults を使用する方法を説明しています。

AWS CLI

次のコマンドは、デフォルトのアカウントとリージョンのボールドを一覧表示します。

```
aws glacier list-vaults --account-id -
```

出力:

```
{
  "VaultList": [
    {
      "SizeInBytes": 3178496,
      "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
      "LastInventoryDate": "2015-04-07T00:26:19.028Z",
      "VaultName": "my-vault",
      "NumberOfArchives": 1,
      "CreationDate": "2015-04-06T21:23:45.708Z"
    }
  ]
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVaults](#)」の「」を参照してください。

purchase-provisioned-capacity

次のコード例は、`purchase-provisioned-capacity` を使用する方法を示しています。

AWS CLI

プロビジョニングされたキャパシティーユニットを購入するには

次の`purchase-provisioned-capacity`例では、プロビジョニングされたキャパシティーユニットを購入します。

```
aws glacier purchase-provisioned-capacity \  
  --account-id 111122223333
```

出力:

```
{  
  "capacityId": "HpASAUvfRFiVDb0jMfEIcr8K"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PurchaseProvisionedCapacity](#)」の「」を参照してください。

remove-tags-from-vault

次のコード例は、`remove-tags-from-vault` を使用する方法を示しています。

AWS CLI

次のコマンドは、`date` という名前のボールドdateから キーを持つタグを削除しますmy-vault。

```
aws glacier remove-tags-from-vault --account-id - --vault-name my-vault --tag-  
keys date
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveTagsFromVault](#)」の「」を参照してください。

set-data-retrieval-policy

次の例は、set-data-retrieval-policy を使用する方法を説明しています。

AWS CLI

次のコマンドは、使用中アカウントのデータ取得ポリシーを設定します。

```
aws glacier set-data-retrieval-policy --account-id - --policy file://data-retrieval-policy.json
```

data-retrieval-policy.json は、データ取得ポリシーを指定する現在のフォルダ内の JSON ファイルです。

```
{
  "Rules": [
    {
      "Strategy": "BytesPerHour",
      "BytesPerHour": 10737418240
    }
  ]
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

次のコマンドは、インライン FreeTier を使用するようにデータ取得ポリシーを設定します JSON。

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules": [{"Strategy": "FreeTier"}]}'
```

ポリシー形式の詳細については、Amazon Glacier API リファレンスの「データ取得ポリシーの設定」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetDataRetrievalPolicy](#)」の「」を参照してください。

set-vault-access-policy

次のコード例は、set-vault-access-policy を使用する方法を示しています。

AWS CLI

ポールのアクセスポリシーを設定するには

次のset-vault-access-policy例では、指定されたポールのアクセス許可ポリシーをアタッチします。

```
aws glacier set-vault-access-policy \  
  --account-id 111122223333 \  
  --vault-name example_vault \  
  --policy '{"Policy": "{\n  \"Version\": \"2012-10-17\", \n  \"Statement\": [\n    {\n      \"Effect\": \"Allow\", \n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam::444455556666:root\"\n      }, \n      \"Action\": \"glacier:ListJobs\", \n      \"Resource\": \"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"\n    }, \n    {\n      \"Effect\": \"Allow\", \n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam::444455556666:root\"\n      }, \n      \"Action\": \"glacier:UploadArchive\", \n      \"Resource\": \"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault\"\n    }\n  ]\n}"}'
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetVaultAccessPolicy](#)」の「」を参照してください。

set-vault-notifications

次の例は、set-vault-notifications を使用する方法を説明しています。

AWS CLI

次のコマンドは、 という名前のポールのSNSの通知を設定しますmy-vault。

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-notification-config file://notificationconfig.json
```

notificationconfig.json は、発行するSNSトピックとイベントを指定する現在のフォルダ内のJSONファイルです。

```
{
```

```
"SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
"Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetVaultNotifications](#)」の「」を参照してください。

upload-archive

次のコード例は、upload-archive を使用する方法を示しています。

AWS CLI

次のコマンドは、archive.zip という名前の現在のフォルダにあるアーカイブを、my-vault という名前のポールトにアップロードします。

```
aws glacier upload-archive --account-id - --vault-name my-vault --body archive.zip
```

出力:

```
{
  "archiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGElWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
  "checksum": "969fb39823836d81f0cc028195fcdcbbbe76cdde932d4646fa7de5f21e18aa67",
  "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGElWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw"
}
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

アップロードしたアーカイブを取得するには、aws glacier initiate-job コマンドを使用して取得ジョブを開始します。

- API 詳細については、AWS CLI 「コマンドリファレンス[UploadArchive](#)」の「」を参照してください。

upload-multipart-part

次の例は、upload-multipart-part を使用する方法を説明しています。

AWS CLI

次のコマンドは、アーカイブの最初の 1 MiB (1024 x 1024 バイト) の部分をアップロードします。

```
aws glacier upload-multipart-part --body part1 --range 'bytes  
0-1048575/*' --account-id - --vault-name my-vault --upload-  
id 19gaRezEXAMPLES6Ry5YYdqthH0C_kGRCT03L9yetr220UmPtBYKk-  
0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ
```

Amazon Glacier では、オペレーションを実行する際にアカウント ID 引数が必要ですが、ハイフンを使用して使用中のアカウントを指定できます。

body パラメータは、ローカルファイルシステム上のパートファイルへのパスを受け取ります。range パラメータは、完了したアーカイブでパーツが占有するバイト数を示す HTTP コンテンツ範囲を取得します。アップロード ID は aws glacier initiate-multipart-upload コマンドによって返され、aws glacier list-multipart-uploads を使用して取得することもできます。

を使用した Amazon Glacier へのマルチパートアップロードの詳細については AWS CLI、AWS CLI ユーザーガイドの Amazon Glacier の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UploadMultipartPart](#)」の「」を参照してください。

を使用した Secrets Manager の例 AWS CLI

次のコード例は、Secrets Manager AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-get-secret-value

次のコード例は、batch-get-secret-value を使用する方法を示しています。

AWS CLI

例 1: 名前でリストされたシークレットのグループのシークレット値を取得するには

次のbatch-get-secret-value例では、3つのシークレットのシークレット値シークレットを取得します。

```
aws secretsmanager batch-get-secret-value \  
  --secret-id-list MySecret1 MySecret2 MySecret3
```

出力:

```
{  
  "SecretValues": [  
    {  
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-  
a1b2c3",  
      "Name": "MySecret1",  
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa",  
      "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-  
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-  
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":  
\"secretsmanagertutorial\"}",  
      "VersionStages": [  
        "AWSCURRENT"  
      ],  
      "CreateDate": "1523477145.729"  
    },  
    {  
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-  
a1b2c3",  
      "Name": "MySecret2",  
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb",
```

```

        "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\""},
        "VersionStages": [
            "AWSCURRENT"
        ],
        "CreateDate": "1673477781.275"
    },
    {
        "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
        "Name": "MySecret3",
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEcccc",
        "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\""},
        "VersionStages": [
            "AWSCURRENT"
        ],
        "CreateDate": "1373477721.124"
    }
],
"Errors": []
}

```

詳細については、[「Secrets Manager ユーザーガイド」の「バッチ内のシークレットのグループを取得するAWS」](#)を参照してください。

例 2: フィルターで選択されたシークレットのグループのシークレット値を取得するには

次のbatch-get-secret-value例では、というMySecret名前を持つアカウントのシークレット値シークレットを取得します。名前によるフィルタリングでは、大文字と小文字が区別されません。

```

aws secretsmanager batch-get-secret-value \
  --filters Key="name",Values="MySecret"

```

出力:

```

{
  "SecretValues": [
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",

```

```

        "Name": "MySecret1",
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
        "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
        "VersionStages": [
            "AWSCURRENT"
        ],
        "CreateDate": "1523477145.729"
    },
    {
        "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
        "Name": "MySecret2",
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb",
        "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"}",
        "VersionStages": [
            "AWSCURRENT"
        ],
        "CreateDate": "1673477781.275"
    },
    {
        "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
        "Name": "MySecret3",
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEccccc",
        "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\"}",
        "VersionStages": [
            "AWSCURRENT"
        ],
        "CreateDate": "1373477721.124"
    }
  ],
  "Errors": []
}

```

詳細については、[「Secrets Manager ユーザーガイド」の「バッチ内のシークレットのグループを取得するAWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetSecretValue](#)」の「」を参照してください。

cancel-rotate-secret

次のコード例は、cancel-rotate-secret を使用する方法を示しています。

AWS CLI

シークレットの自動ローテーションを無効にするには

次のcancel-rotate-secret例では、シークレットの自動ローテーションをオフにします。ローテーションを再開するには、 を呼び出します rotate-secret。

```
aws secretsmanager cancel-rotate-secret \  
  --secret-id MyTestSecret
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の [「シークレットのローテーション」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CancelRotateSecret](#)」の「」を参照してください。

create-secret

次のコード例は、create-secret を使用する方法を示しています。

AWS CLI

例 1: JSON ファイル内の認証情報からシークレットを作成するには

次の create-secret の例は、ファイル内の認証情報からシークレットを作成します。詳細については、「ユーザーガイド」の [「ファイルからのパラメータのロード AWS CLI」](#) を参照してください。AWS CLI

```
aws secretsmanager create-secret \  
  --name MyTestSecret \  
  --secret-string file://mycreds.json
```

mycreds.json の内容:

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

出力:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの作成](#)」を参照してください。

例 2: シークレットを作成するには

次に、2 つのキーと値のペアを持つシークレットを作成する、create-secret の例をします。コマンドシェルにコマンドを入力すると、コマンド履歴がアクセスされたり、ユーティリティからコマンドパラメータにアクセスされたりするリスクがあります。これは、コマンドにシークレットの値が含まれている場合に懸念されます。詳細については、「[Secrets Manager ユーザーガイド](#)」の「[コマンドラインツールを使用してシークレットを保存するリスクを軽減する](#)」を参照してください。

```
aws secretsmanager create-secret \
  --name MyTestSecret \
  --description "My test secret created with the CLI." \
  --secret-string "{\"user\": \"diegor\", \"password\": \"EXAMPLE-PASSWORD\"}"
```

出力:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
```

```
"Name": "MyTestSecret",  
"VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSecret](#)」の「」を参照してください。

delete-resource-policy

次のコード例は、delete-resource-policy を使用する方法を示しています。

AWS CLI

シークレットにアタッチされたリソースベースのポリシーを削除するには

次に、シークレットにアタッチされているリソースベースのポリシーを削除する、delete-resource-policy の例を示します。

```
aws secretsmanager delete-resource-policy \  
--secret-id MyTestSecret
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[認証とアクセスコントロール](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResourcePolicy](#)」の「」を参照してください。

delete-secret

次の例は、delete-secret を使用する方法を説明しています。

AWS CLI

例 1: シークレットを削除するには

次の `delete-secret` の例は、シークレットを削除します。 `restore-secret` レスポンスフィールドで示される日時までは、`DeletionDate` でこのシークレットを復旧できます。他のリージョンにレプリカが作成されているシークレットを削除する場合は、まずそのレプリカを `remove-regions-from-replication` で削除してから、`delete-secret` を呼び出します。

```
aws secretsmanager delete-secret \  
  --secret-id MyTestSecret \  
  --recovery-window-in-days 7
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
  "DeletionDate": 1524085349.095  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの削除](#)」を参照してください。

例 2: シークレットを直ちに削除するには

次の `delete-secret` の例は、復旧期間なしでシークレットを直ちに削除します。この場合のシークレットは復元できません。

```
aws secretsmanager delete-secret \  
  --secret-id MyTestSecret \  
  --force-delete-without-recovery
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
}
```

```
"DeletionDate": 1508750180.309
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSecret](#)」の「」を参照してください。

describe-secret

次の例は、describe-secret を使用する方法を説明しています。

AWS CLI

シークレットの詳細を取得するには

次の describe-secret の例は、シークレットの詳細を表示します。

```
aws secretsmanager describe-secret \
  --secret-id MyTestSecret
```

出力:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
Ca8JGt",
  "Name": "MyTestSecret",
  "Description": "My test secret",
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-
ba987EXAMPLE",
  "RotationEnabled": true,
  "RotationLambdaARN": "arn:aws:lambda:us-
west-2:123456789012:function:MyTestRotationLambda",
  "RotationRules": {
    "AutomaticallyAfterDays": 2,
    "Duration": "2h",
    "ScheduleExpression": "cron(0 16 1,15 * ? *)"
  },
  "LastRotatedDate": 1525747253.72,
  "LastChangedDate": 1523477145.729,
```



```
"LastAccessedDate": 1524572133.25,
"Tags": [
  {
    "Key": "SecondTag",
    "Value": "AnotherValue"
  },
  {
    "Key": "FirstTag",
    "Value": "SomeValue"
  }
],
"VersionIdsToStages": {
  "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
    "AWSPREVIOUS"
  ],
  "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [
    "AWSCURRENT"
  ],
  "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333": [
    "AWSPENDING"
  ]
},
"CreateDate": 1521534252.66,
"PrimaryRegion": "us-west-2",
"ReplicationStatus": [
  {
    "Region": "eu-west-3",
    "KmsKeyId": "alias/aws/secretsmanager",
    "Status": "InSync",
    "StatusMessage": "Replication succeeded"
  }
]
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレット](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSecret](#)」の「」を参照してください。

get-random-password

次のコード例は、get-random-password を使用する方法を示しています。

AWS CLI

ランダムパスワードを生成するには

次のget-random-password例では、少なくとも 1 つの大文字、小文字、数字、句読点を含む 20 文字のランダムなパスワードを生成します。

```
aws secretsmanager get-random-password \  
  --require-each-included-type \  
  --password-length 20
```

出力:

```
{  
  "RandomPassword": "EXAMPLE-PASSWORD"  
}
```

詳細については、[「Secrets Manager ユーザーガイド」の「シークレットの作成と管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRandomPassword](#)」の「」を参照してください。

get-resource-policy

次のコード例は、get-resource-policy を使用する方法を示しています。

AWS CLI

シークレットにアタッチされたリソースベースのポリシーを取得するには

次に、シークレットにアタッチされたリソースベースのポリシーを取得する、get-resource-policy の例を示します。

```
aws secretsmanager get-resource-policy \  
  --secret-id MyTestSecret
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
```

```
"Name": "MyTestSecret",
"ResourcePolicy": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [{\n    \"Effect\":\n    \\\"Allow\\\", \n    \n    \"Principal\": {\n      \"AWS\": \"arn:aws:iam::123456789012:root\"\n    }, \n    \"Action\":\n    \\\"secretsmanager:GetSecretValue\\\", \n    \n    \"Resource\": \"*\"\n  }]\n}"
}
```

詳細については、「Secrets Manager ユーザーガイド」の [「認証とアクセスコントロール」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetResourcePolicy](#)」の「」を参照してください。

get-secret-value

次のコード例は、get-secret-value を使用する方法を示しています。

AWS CLI

例 1: シークレットの暗号化されたシークレット値を取得するには

次の get-secret-value の例は、現在のシークレット値を取得します。

```
aws secretsmanager get-secret-value \
  --secret-id MyTestSecret
```

出力:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "SecretString": "{\"user\": \"diegor\", \"password\": \"EXAMPLE-PASSWORD\"}",
  "VersionStages": [
    "AWSCURRENT"
  ],
  "CreateDate": 1523477145.713
}
```

詳細については、「Secrets Manager ユーザーガイド」の [「シークレットの取得」](#) を参照してください。

例 2: 前のシークレット値を取得するには

次の `get-secret-value` の例は、前のシークレット値を取得します。

```
aws secretsmanager get-secret-value \  
  --secret-id MyTestSecret \  
  --version-stage AWSPREVIOUS
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "SecretString": "{\"user\": \"diegor\", \"password\": \"PREVIOUS-EXAMPLE-PASSWORD  
}\"",  
  "VersionStages": [  
    "AWSPREVIOUS"  
  ],  
  "CreateDate": 1523477145.713  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの取得](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSecretValue](#)」の「」を参照してください。

list-secret-version-ids

次の例は、`list-secret-version-ids` を使用する方法を説明しています。

AWS CLI

シークレットに関連付けられているすべてのシークレットバージョンを一覧表示するには

次の `list-secret-version-ids` 例では、シークレットのすべてのバージョンのリストを取得します。

```
aws secretsmanager list-secret-version-ids \  
  --secret-id MyTestSecret
```

出力:

```
{
  "Versions": [
    {
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "VersionStages": [
        "AWSPREVIOUS"
      ],
      "LastAccessedDate": 1523477145.713,
      "CreateDate": 1523477145.713
    },
    {
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "LastAccessedDate": 1523477145.713,
      "CreateDate": 1523486221.391
    },
    {
      "CreateDate": 1.51197446236E9,
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333;"
    }
  ],
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret"
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[バージョン](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSecretVersionIds](#)」の「」を参照してください。

list-secrets

次の例は、list-secrets を使用する方法を説明しています。

AWS CLI

例 1: アカウント内のシークレットを一覧表示するには

以下の `list-secrets` の例は、アカウント内のシークレットの一覧を取得します。

```
aws secretsmanager list-secrets
```

出力:

```
{
  "SecretList": [
    {
      "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
      "Name": "MyTestSecret",
      "LastChangedDate": 1523477145.729,
      "SecretVersionsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
          "AWSCURRENT"
        ]
      }
    },
    {
      "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:AnotherSecret-d4e5f6",
      "Name": "AnotherSecret",
      "LastChangedDate": 1523482025.685,
      "SecretVersionsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222": [
          "AWSCURRENT"
        ]
      }
    }
  ]
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[AWS Secrets Manager でシークレットを検索する](#)」を参照してください。

例 2: アカウント内のシークレットの一覧をフィルタリングするには

次の `list-secrets` の例は、アカウント内にあり、名前に `Test` が含まれているシークレットの一覧を取得します。名前によるフィルタリングでは、大文字と小文字が区別されます。

```
aws secretsmanager list-secrets \
```

```
--filter Key="name",Values="Test"
```

出力:

```
{
  "SecretList": [
    {
      "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
      "Name": "MyTestSecret",
      "LastChangedDate": 1523477145.729,
      "SecretVersionsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
          "AWSCURRENT"
        ]
      }
    }
  ]
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[AWS Secrets Manager でシークレットを検索する](#)」を参照してください。

例 3: 別のサービスで管理されているアカウントのシークレットを一覧表示するには

次のlist-secrets例では、Amazon によって管理されているアカウントのシークレットを返しますRDS。

```
aws secretsmanager list-secrets \
  --filter Key="owning-service",Values="rds"
```

出力:

```
{
  "SecretList": [
    {
      "Name": "rds!cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Tags": [
        {
          "Value": "arn:aws:rds:us-
west-2:123456789012:cluster:database-1",

```

```

        "Key": "aws:rds:primaryDBClusterArn"
      },
      {
        "Value": "rds",
        "Key": "aws:secretsmanager:owningService"
      }
    ],
    "RotationRules": {
      "AutomaticallyAfterDays": 1
    },
    "LastChangedDate": 1673477781.275,
    "LastRotatedDate": 1673477781.26,
    "SecretVersionsToStages": {
      "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa": [
        "AWSPREVIOUS"
      ],
      "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb": [
        "AWSCURRENT",
        "AWSPENDING"
      ]
    },
    "OwningService": "rds",
    "RotationEnabled": true,
    "CreatedDate": 1673467300.7,
    "LastAccessedDate": 1673395200.0,
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:rds!
cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-a1b2c3",
    "Description": "Secret associated with primary RDS DB cluster:
arn:aws:rds:us-west-2:123456789012:cluster:database-1"
  }
]
}

```

詳細については、「Secrets Manager ユーザーガイド」の「[他の AWS のサービスによって管理される AWS Secrets Manager のシークレット](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSecrets](#)」の「」を参照してください。

put-resource-policy

次のコード例は、put-resource-policy を使用する方法を示しています。

AWS CLI

リソーススペースのポリシーをシークレットに追加するには

次の `put-resource-policy` の例では、ポリシーが広範なアクセスをシークレットに提供していないことを最初に確認しながら、シークレットに許可ポリシーを追加しています。このポリシーは、ファイルから読み込まれます。詳細については、「[ユーザーガイド](#)」の「[ファイルからのパラメータのロード AWS CLI](#)」を参照してください。AWS CLI

```
aws secretsmanager put-resource-policy \  
  --secret-id MyTestSecret \  
  --resource-policy file://mypolicy.json \  
  --block-public-policy
```

`mypolicy.json` の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::123456789012:role/MyRole"  
      },  
      "Action": "secretsmanager:GetSecretValue",  
      "Resource": "*"   
    }  
  ]  
}
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「[Secrets Manager ユーザーガイド](#)」の「[シークレットにアクセス許可ポリシーをアタッチする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutResourcePolicy](#)」の「」を参照してください。

put-secret-value

次のコード例は、put-secret-value を使用する方法を示しています。

AWS CLI

例 1: シークレットの新しいシークレット値を保存するには

次の put-secret-value の例は、キーと値の 2 つのペアを含むシークレットの新しいバージョンを作成します。

```
aws secretsmanager put-secret-value \  
  --secret-id MyTestSecret \  
  --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "VersionStages": [  
    "AWSCURRENT"  
  ]  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの変更](#)」を参照してください。

例 2: 認証情報から新しいシークレット値を JSON ファイルに保存するには

次の put-secret-value の例では、ファイル内の認証情報からシークレットの新しいバージョンが作成されます。詳細については、「ユーザーガイド」の「[ファイルからのパラメータのロード AWS CLI](#)」を参照してください。AWS CLI

```
aws secretsmanager put-secret-value \  
  --secret-id MyTestSecret \  
  --secret-file secret.json
```

```
--secret-string file://mycreds.json
```

mycreds.json の内容:

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

出力:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "VersionStages": [
    "AWSCURRENT"
  ]
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutSecretValue](#)」の「」を参照してください。

remove-regions-from-replication

次の例は、remove-regions-from-replication を使用する方法を説明しています。

AWS CLI

レプリカシークレットを削除するには

次の remove-regions-from-replication の例では、eu-west-3 にあるレプリカシークレットを削除しています。他のリージョンにレプリカが作成されているプライマリシークレットを削除するには、まずそのレプリカを削除してから delete-secret を呼び出します。

```
aws secretsmanager remove-regions-from-replication \  
--secret-id MyTestSecret \  
--remove-replica-regions eu-west-3
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "ReplicationStatus": []  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[レプリカシークレットの削除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveRegionsFromReplication](#)」の「」を参照してください。

replicate-secret-to-regions

次の例は、`replicate-secret-to-regions` を使用する方法を説明しています。

AWS CLI

シークレットを別のリージョンにレプリケートするには

次に、シークレットを `eu-west-3` にレプリケートする、`replicate-secret-to-regions` の例を示します。レプリカは、AWS マネージドキーで暗号化されます `aws/secretsmanager`。

```
aws secretsmanager replicate-secret-to-regions \  
--secret-id MyTestSecret \  
--add-replica-regions Region=eu-west-3
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "ReplicationStatus": [  
    {
```

```
        "Region": "eu-west-3",
        "KmsKeyId": "alias/aws/secretsmanager",
        "Status": "InProgress"
    }
]
}
```

詳細については、Secrets Manager ユーザーガイドの [「別のリージョンにシークレットをレプリケートする」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ReplicateSecretToRegions](#)」の「」を参照してください。

restore-secret

次のコード例は、restore-secret を使用する方法を示しています。

AWS CLI

以前に削除したシークレットを復元するには

次の restore-secret の例では、スケジュールにより以前に削除されたシークレットを復元します。

```
aws secretsmanager restore-secret \  
  --secret-id MyTestSecret
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の [「シークレットの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RestoreSecret](#)」の「」を参照してください。

rotate-secret

次の例は、rotate-secret を使用する方法を説明しています。

AWS CLI

例 1: シークレットの自動ローテーションを設定して開始するには

次のrotate-secret例では、シークレットの自動ローテーションを設定および開始します。Secrets Manager は、シークレットをすぐに 1 回ローテーションし、その後 8 時間ごとに 2 時間ウィンドウでローテーションします。出力には、ローテーションによって作成されたVersionId新しいシークレットバージョンの が表示されます。

```
aws secretsmanager rotate-secret \  
  --secret-id MyTestDatabaseSecret \  
  --rotation-lambda-arn arn:aws:lambda:us-  
west-2:1234566789012:function:SecretsManagerTestRotationLambda \  
  --rotation-rules "{\"ScheduleExpression\": \"cron(0 8/8 * * ? *)\"\", \"Duration  
\": \"2h\"}"
```

出力:

```
{  
  "ARN": "aws:arn:secretsmanager:us-  
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",  
  "Name": "MyTestDatabaseSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[「Secrets Manager ユーザーガイド」の「シークレットのローテーション」](#)を参照してください。

例 2: ローテーション間隔で自動ローテーションを設定および開始するには

次のrotate-secret例では、シークレットの自動ローテーションを設定および開始します。Secrets Manager はシークレットをすぐに 1 回ローテーションし、その後は 10 日ごとにローテーションします。出力には、ローテーションによって作成されたVersionId新しいシークレットバージョンの が表示されます。

```
aws secretsmanager rotate-secret \  

```

```
--secret-id MyTestDatabaseSecret \  
--rotation-lambda-arn arn:aws:lambda:us-west-2:123456789012:function:SecretsManagerTestRotationLambda \  
--rotation-rules "{\"ScheduleExpression\": \"rate(10 days)\"}"
```

出力:

```
{  
  "ARN": "aws:arn:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",  
  "Name": "MyTestDatabaseSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[「Secrets Manager ユーザーガイド」の「シークレットのローテーション」](#)を参照してください。

例 3: シークレットをすぐにローテーションするには

次の rotate-secret の例では、すぐにローテーションが開始されます。出力には、ローテーションによって作成されたVersionId新しいシークレットバージョンの が表示されます。シークレットのローテーションは、すでに設定されている必要があります。

```
aws secretsmanager rotate-secret \  
--secret-id MyTestDatabaseSecret
```

出力:

```
{  
  "ARN": "aws:arn:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",  
  "Name": "MyTestDatabaseSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、[「Secrets Manager ユーザーガイド」の「シークレットのローテーション」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RotateSecret](#)」の「」を参照してください。

stop-replication-to-replica

次の例は、stop-replication-to-replica を使用する方法を説明しています。

AWS CLI

レプリカシークレットをプライマリに昇格させるには

次の stop-replication-to-replica の例は、レプリカシークレットからプライマリへのリンクを削除します。このレプリカシークレットは、レプリカのリージョンでプライマリシークレットに昇格されます。stop-replication-to-replica は、レプリカリージョン内から呼び出す必要があります。

```
aws secretsmanager stop-replication-to-replica \  
  --secret-id MyTestSecret
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の [「レプリカシークレットの昇格」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StopReplicationToReplica](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

例 1: シークレットにタグを追加するには

次の の例は、短縮構文を使用してタグをアタッチする方法を示しています。

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tag-key MyTagKey \  
  --tag-value MyTagValue
```



```
--tags Key=FirstTag,Value=FirstValue
```

このコマンドでは何も出力されません。

詳細については、「[Secrets Manager ユーザーガイド](#)」の「[シークレットにタグを付ける](#)」を参照してください。

例 2: シークレットに複数のタグを追加するには

次の `tag-resource` の例では、キーと値のタグ 2 個がシークレットにアタッチされます。

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",  
  "Value": "SecondValue"}]'
```

このコマンドでは何も出力されません。

詳細については、「[Secrets Manager ユーザーガイド](#)」の「[タグシークレット](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンス TagResource](#)」の「」を参照してください。

untag-resource

次の例は、`untag-resource` を使用する方法を説明しています。

AWS CLI

シークレットからタグを削除するには

次の `untag-resource` の例では、シークレットから 2 個のタグが削除されます。タグごとに、キーと値の両方が削除されます。

```
aws secretsmanager untag-resource \  
  --secret-id MyTestSecret \  
  --tag-keys '[ "FirstTag", "SecondTag" ]'
```

このコマンドでは何も出力されません。

詳細については、「Secrets Manager ユーザーガイド」の「[タグシークレット](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-secret-version-stage

次のコード例は、update-secret-version-stage を使用する方法を示しています。

AWS CLI

例 1: シークレットを以前のバージョンに戻すには

次のupdate-secret-version-stage例では、AWS CURRENTステージングラベルを以前のバージョンのシークレットに移動し、シークレットを以前のバージョンに戻します。以前のバージョンの ID を検索するには、を使用しますlist-secret-version-ids。この例では、ラベル付きの AWS CURRENTバージョンは a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 で、ラベル付きの AWS PREVIOUSバージョンは a1b2c3d4-5678-90ab-cdef-EXAMPLE22222。この例では、ラベルを AWS CURRENTバージョン 11111 から 22222 に移動します。ラベルはバージョンから削除されるため AWS CURRENT、はラベルを AWS PREVIOUSそのバージョン (11111) update-secret-version-stageに自動的に移動します。その結果、と AWS PREVIOUS AWS CURRENTのバージョンがスワップされます。

```
aws secretsmanager update-secret-version-stage \  
  --secret-id MyTestSecret \  
  --version-stage AWSCURRENT \  
  --move-to-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \  
  --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[バージョン](#)」を参照してください。

例 2: シークレットのバージョンにアタッチされたステージングラベルを追加するには

次のupdate-secret-version-stage例では、シークレットのバージョンにステージングラベルを追加します。影響のあるバージョンのVersionStagesレスポンスフィールドを実行して表示することでlist-secret-version-ids、結果を確認できます。

```
aws secretsmanager update-secret-version-stage \  
  --secret-id MyTestSecret \  
  --version-stage STAGINGLABEL1 \  
  --move-to-version-id EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[バージョン](#)」を参照してください。

例 3: シークレットのバージョンにアタッチされたステージングラベルを削除するには

次のupdate-secret-version-stage例では、シークレットのバージョンにアタッチされているステージングラベルを削除します。影響のあるバージョンのVersionStagesレスポンスフィールドを実行して表示することでlist-secret-version-ids、結果を確認できます。

```
aws secretsmanager update-secret-version-stage \  
  --secret-id MyTestSecret \  
  --version-stage STAGINGLABEL1 \  
  --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[バージョン](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSecretVersionStage](#)」の「」を参照してください。

update-secret

次の例は、update-secret を使用する方法を説明しています。

AWS CLI

例 1: シークレットの記述を更新するには

次の update-secret の例は、シークレットの記述を更新します。

```
aws secretsmanager update-secret \  
  --secret-id MyTestSecret \  
  --description "This is a new description for the secret."
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの変更](#)」を参照してください。

例 2: シークレットに関連付けられた暗号化キーを更新するには

次の update-secret 例では、シークレット値の暗号化に使用される KMS キーを更新します。KMS キーはシークレットと同じリージョンにある必要があります。

```
aws secretsmanager update-secret \  
  --secret-id MyTestSecret \  
  --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-  
ba987EXAMPLE
```

出力:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

詳細については、「Secrets Manager ユーザーガイド」の「[シークレットの変更](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSecret](#)」の「」を参照してください。

validate-resource-policy

次の例は、validate-resource-policy を使用する方法を説明しています。

AWS CLI

リソースポリシーを検証するには

次のvalidate-resource-policy例では、リソースポリシーがシークレットへの広範なアクセスを許可していないことを確認します。ポリシーはディスク上のファイルから読み取られます。詳細については、「[ユーザーガイド](#)」の「[ファイルからのパラメータのロード AWS CLI](#)」を参照してください。AWS CLI

```
aws secretsmanager validate-resource-policy \  
--resource-policy file://mypolicy.json
```

mypolicy.json の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::123456789012:role/MyRole"  
      },  
      "Action": "secretsmanager:GetSecretValue",  
      "Resource": "*"   
    }   
  ]  
}
```

出力:

```
{  
  "PolicyValidationPassed": true,  
}
```

```
"ValidationErrors": []
}
```

詳細については、[「Secrets Manager ユーザーガイド」の「Secrets Manager のアクセス許可リファレンス」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ValidateResourcePolicy](#)」の「」を参照してください。

を使用した Security Hub の例 AWS CLI

次のコード例は、Security Hub AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-administrator-invitation

次のコード例は、accept-administrator-invitation を使用する方法を示しています。

AWS CLI

管理者アカウントからの招待を受け入れるには

次のaccept-administrator-invitation例では、指定された管理者アカウントからの指定された招待を受け入れます。

```
aws securityhub accept-invitation \  
  --administrator-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptAdministratorInvitation](#)」の「」を参照してください。

accept-invitation

次の例は、accept-invitation を使用する方法を説明しています。

AWS CLI

管理者アカウントからの招待を受け入れるには

次のaccept-invitation例では、指定された管理者アカウントからの指定された招待を受け入れます。

```
aws securityhub accept-invitation \  
  --master-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

このコマンドでは何も出力されません。

詳細については、Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptInvitation](#)」の「」を参照してください。

batch-delete-automation-rules

次のコード例は、batch-delete-automation-rules を使用する方法を示しています。

AWS CLI

オートメーションルールを削除するには

次のbatch-delete-automation-rules例では、指定されたオートメーションルールを削除します。1つのコマンドで1つ以上のルールを削除できます。このコマンドを実行できるのはSecurity Hub 管理者アカウントのみです。

```
aws securityhub batch-delete-automation-rules \  
  --automation-rules-arns '["arn:aws:securityhub:us-  
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

出力:

```
{  
  "ProcessedAutomationRules": [  
    "arn:aws:securityhub:us-east-1:123456789012:automation-rule/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  ],  
  "UnprocessedAutomationRules": []  
}
```

詳細については、Security Hub ユーザーガイドの [「オートメーションルールの削除」](#) を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchDeleteAutomationRules](#)」の「」を参照してください。

batch-disable-standards

次の例は、batch-disable-standards を使用する方法を説明しています。

AWS CLI

標準を無効にするには

次のbatch-disable-standards例では、指定されたサブスクリプションに関連付けられた標準を無効にしますARN。

```
aws securityhub batch-disable-standards \  
  --standards-subscription-arns "arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

出力:

```
{  
  "StandardsSubscriptions": [  
    {  
      "StandardsArn": "arn:aws:securityhub:eu-central-1::standards/pci-dss/  
v/3.2.1",
```



```

        "StandardsInput": { },
        "StandardsStatus": "DELETING",
        "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
    }
]
}

```

詳細については、AWS 「Security Hub ユーザーガイド」の「[セキュリティスタンダードの無効化または有効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchDisableStandards](#)」の「」を参照してください。

batch-enable-standards

次のコード例は、batch-enable-standards を使用する方法を示しています。

AWS CLI

標準を有効にするには

次のbatch-enable-standards例では、リクエスト元のアカウントのPCIDSS標準を有効にします。

```

aws securityhub batch-enable-standards \
  --standards-subscription-requests '{"StandardsArn":"arn:aws:securityhub:us-
west-1::standards/pci-dss/v/3.2.1"}'

```

出力:

```

{
  "StandardsSubscriptions": [
    {
      "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
      "StandardsInput": { },
      "StandardsStatus": "PENDING",
      "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
    }
  ]
}

```

```
}
```

詳細については、AWS「[Security Hub ユーザーガイド](#)」の「[セキュリティスタンダードの無効化または有効化](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスBatchEnableStandards](#)」の「」を参照してください。

batch-get-automation-rules

次の例は、batch-get-automation-rules を使用する方法を説明しています。

AWS CLI

オートメーションルールの詳細を取得するには

次のbatch-get-automation-rules例では、指定されたオートメーションルールの詳細を取得します。1つのコマンドで1つ以上のオートメーションルールの詳細を取得できます。

```
aws securityhub batch-get-automation-rules \  
  --automation-rules-arns '["arn:aws:securityhub:us-  
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

出力:

```
{  
  "Rules": [  
    {  
      "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "RuleStatus": "ENABLED",  
      "RuleOrder": 1,  
      "RuleName": "Suppress informational findings",  
      "Description": "Suppress GuardDuty findings with Informational  
severity",  
      "IsTerminal": false,  
      "Criteria": {  
        "ProductName": [  
          {  
            "Value": "GuardDuty",  
            "Comparison": "EQUALS"  
          }  
        ],  
      }  
    }  
  ],  
}
```

```
    "SeverityLabel": [
      {
        "Value": "INFORMATIONAL",
        "Comparison": "EQUALS"
      }
    ],
    "WorkflowStatus": [
      {
        "Value": "NEW",
        "Comparison": "EQUALS"
      }
    ],
    "RecordState": [
      {
        "Value": "ACTIVE",
        "Comparison": "EQUALS"
      }
    ]
  },
  "Actions": [
    {
      "Type": "FINDING_FIELDS_UPDATE",
      "FindingFieldsUpdate": {
        "Note": {
          "Text": "Automatically suppress GuardDuty findings with
Informational severity",
          "UpdatedBy": "sechub-automation"
        },
        "Workflow": {
          "Status": "SUPPRESSED"
        }
      }
    }
  ],
  "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
  "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
  "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
}
],
"UnprocessedAutomationRules": []
}
```

詳細については、AWS Security Hub ユーザーガイドの「[オートメーションルールの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetAutomationRules](#)」の「」を参照してください。

batch-get-configuration-policy-associations

次のコード例は、batch-get-configuration-policy-associations を使用する方法を示しています。

AWS CLI

ターゲットのバッチの設定関連付けの詳細を取得するには

次のbatch-get-configuration-policy-associations例では、指定されたターゲットの関連付けの詳細を取得します。アカウント IDs、組織単位 IDs、またはターゲットのルート ID を指定できます。

```
aws securityhub batch-get-configuration-policy-associations \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}
```

出力:

```
{
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "TargetId": "ou-6hi7-8j91k12m",
  "TargetType": "ORGANIZATIONAL_UNIT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
  "AssociationStatus": "SUCCESS",
  "AssociationStatusMessage": "Association applied successfully on this target."
}
```

詳細については、[Security Hub ユーザーガイドの「Security Hub 設定ポリシーの表示](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetConfigurationPolicyAssociations](#)」の「」を参照してください。

batch-get-security-controls

次の例は、batch-get-security-controls を使用する方法を説明しています。

AWS CLI

セキュリティコントロールの詳細を取得するには

次のbatch-get-security-controls例では、現在の AWS アカウントと AWS リージョンのセキュリティコントロール ACM.1 と IAM.1 の詳細を取得します。

```
aws securityhub batch-get-security-controls \  
--security-control-ids '["ACM.1", "IAM.1"]'
```

出力:

```
{  
  "SecurityControls": [  
    {  
      "SecurityControlId": "ACM.1",  
      "SecurityControlArn": "arn:aws:securityhub:us-  
east-2:123456789012:security-control/ACM.1",  
      "Title": "Imported and ACM-issued certificates should be renewed after a  
specified time period",  
      "Description": "This control checks whether an AWS Certificate Manager  
(ACM) certificate is renewed within the specified time period. It checks both  
imported certificates and certificates provided by ACM. The control fails if the  
certificate isn't renewed within the specified time period. Unless you provide a  
custom parameter value for the renewal period, Security Hub uses a default value of  
30 days.",  
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/  
ACM.1/remediation",  
      "SeverityRating": "MEDIUM",  
      "SecurityControlStatus": "ENABLED",  
      "UpdateStatus": "READY",  
      "Parameters": {  
        "daysToExpiration": {  
          "ValueType": CUSTOM,  
          "Value": {  
            "Integer": 15  
          }  
        }  
      }  
    },  
  ],  
}
```

```
        "LastUpdateReason": "Updated control parameter"
    },
    {
        "SecurityControlId": "IAM.1",
        "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/IAM.1",
        "Title": "IAM policies should not allow full \"*\" administrative
privileges",
        "Description": "This AWS control checks whether the default version of
AWS Identity and Access Management (IAM) policies (also known as customer managed
policies) do not have administrator access with a statement that has \"Effect\":
\"Allow\" with \"Action\": \"*\" over \"Resource\": \"*\". It only checks for
the Customer Managed Policies that you created, but not inline and AWS Managed
Policies.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
IAM.1/remediation",
        "SeverityRating": "HIGH",
        "SecurityControlStatus": "ENABLED"
        "UpdateStatus": "READY",
        "Parameters": {}
    }
]
}
```

詳細については、AWS Security Hub ユーザーガイドの [「コントロールの詳細の表示」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchGetSecurityControls](#)」の「」を参照してください。

batch-get-standards-control-associations

次のコード例は、batch-get-standards-control-associations を使用する方法を示しています。

AWS CLI

コントロールの有効化ステータスを取得するには

次のbatch-get-standards-control-associations例では、指定されたコントロールが指定された標準で有効になっているかどうかを示します。

```
aws securityhub batch-get-standards-control-associations \
```

```
--standards-control-association-ids ' [{"SecurityControlId":  
"Config.1", "StandardsArn": "arn:aws:securityhub:us-east-1:123456789012:ruleset/cis-  
aws-foundations-benchmark/v/1.2.0"}, {"SecurityControlId": "IAM.6", "StandardsArn":  
"arn:aws:securityhub:us-east-1:123456789012:standards/aws-foundational-security-  
best-practices/v/1.0.0"} ]'
```

出力:

```
{  
  "StandardsControlAssociationDetails": [  
    {  
      "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-  
benchmark/v/1.2.0",  
      "SecurityControlId": "Config.1",  
      "SecurityControlArn": "arn:aws:securityhub:us-  
east-1:068873283051:security-control/Config.1",  
      "AssociationStatus": "ENABLED",  
      "RelatedRequirements": [  
        "CIS AWS Foundations 2.5"  
      ],  
      "UpdatedAt": "2022-10-27T16:07:12.960000+00:00",  
      "StandardsControlTitle": "Ensure AWS Config is enabled",  
      "StandardsControlDescription": "AWS Config is a web service that  
performs configuration management of supported AWS resources within your account  
and delivers log files to you. The recorded information includes the configuration  
item (AWS resource), relationships between configuration items (AWS resources), and  
any configuration changes between resources. It is recommended to enable AWS Config  
in all regions.",  
      "StandardsControlArns": [  
        "arn:aws:securityhub:us-east-1:068873283051:control/cis-aws-  
foundations-benchmark/v/1.2.0/2.5"  
      ]  
    },  
    {  
      "StandardsArn": "arn:aws:securityhub:us-east-1::standards/aws-  
foundational-security-best-practices/v/1.0.0",  
      "SecurityControlId": "IAM.6",  
      "SecurityControlArn": "arn:aws:securityhub:us-  
east-1:068873283051:security-control/IAM.6",  
      "AssociationStatus": "DISABLED",  
      "RelatedRequirements": [],  
      "UpdatedAt": "2022-11-22T21:30:35.080000+00:00",  
      "UpdatedReason": "test",  
    }  
  ]  
}
```

```

        "StandardsControlTitle": "Hardware MFA should be enabled for the root
user",
        "StandardsControlDescription": "This AWS control checks whether your AWS
account is enabled to use a hardware multi-factor authentication (MFA) device to
sign in with root user credentials.",
        "StandardsControlArns": [
            "arn:aws:securityhub:us-east-1:068873283051:control/aws-
foundational-security-best-practices/v/1.0.0/IAM.6"
        ]
    }
]
}

```

詳細については、AWS 「Security Hub ユーザーガイド」の「[特定の標準でのコントロールの有効化と無効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [BatchGetStandardsControlAssociations](#)」の「」を参照してください。

batch-import-findings

次のコード例は、batch-import-findings を使用する方法を示しています。

AWS CLI

結果を更新するには

次のbatch-import-findings例では、検出結果を更新します。

```

aws securityhub batch-import-findings \
  --findings '
  [{
    "AwsAccountId": "123456789012",
    "CreatedAt": "2020-05-27T17:05:54.832Z",
    "Description": "Vulnerability in a CloudTrail trail",
    "FindingProviderFields": {
      "Severity": {
        "Label": "LOW",
        "Original": "10"
      },
      "Types": [
        "Software and Configuration Checks/Vulnerabilities/CVE"
      ]
    }
  ]

```



```
    ]
  },
  "GeneratorId": "TestGeneratorId",
  "Id": "Id1",
  "ProductArn": "arn:aws:securityhub:us-
west-1:123456789012:product/123456789012/default",
  "Resources": [
    {
      "Id": "arn:aws:cloudtrail:us-west-1:123456789012:trail/
TrailName",
      "Partition": "aws",
      "Region": "us-west-1",
      "Type": "AwsCloudTrailTrail"
    }
  ],
  "SchemaVersion": "2018-10-08",
  "Title": "CloudTrail trail vulnerability",
  "UpdatedAt": "2020-06-02T16:05:54.832Z"
}]'
```

出力:

```
{
  "FailedCount": 0,
  "SuccessCount": 1,
  "FailedFindings": []
}
```

詳細については、AWS Security Hub ユーザーガイドの「[を使用して検出結果を作成および更新 BatchImportFindings する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchImportFindings](#)」の「」を参照してください。

batch-update-automation-rules

次の例は、batch-update-automation-rules を使用する方法を説明しています。

AWS CLI

オートメーションルールを更新するには

次のbatch-update-automation-rules例では、指定されたオートメーションルールを更新します。1つのコマンドで1つ以上のルールを更新できます。このコマンドを実行できるのはSecurity Hub 管理者アカウントのみです。

```
aws securityhub batch-update-automation-rules \
  --update-automation-rules-request-items '[ \
    { \
      "Actions": [{ \
        "Type": "FINDING_FIELDS_UPDATE", \
        "FindingFieldsUpdate": { \
          "Note": { \
            "Text": "Known issue that is a risk", \
            "UpdatedBy": "sechub-automation" \
          }, \
          "Workflow": { \
            "Status": "NEW" \
          } \
        } \
      } \
    ], \
    "Criteria": { \
      "SeverityLabel": [{ \
        "Value": "LOW", \
        "Comparison": "EQUALS" \
      }] \
    }, \
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", \
    "RuleOrder": 1, \
    "RuleStatus": "DISABLED" \
  ]'
```

出力:

```
{
  "ProcessedAutomationRules": [
    "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  ],
  "UnprocessedAutomationRules": []
}
```

詳細については、AWS Security Hub ユーザーガイドの「[オートメーションルールの編集](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchUpdateAutomationRules](#)」の「」を参照してください。

batch-update-findings

次の例は、batch-update-findings を使用する方法を説明しています。

AWS CLI

例 1: 結果を更新するには

次のbatch-update-findings例では、2 つの検出結果を更新してメモを追加し、重要度ラベルを変更して解決します。

```
aws securityhub batch-update-findings \  
  --finding-identifiers '[{"Id": "arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111", "ProductArn": "arn:aws:securityhub:us-  
west-1::product/aws/securityhub"}, {"Id": "arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222", "ProductArn": "arn:aws:securityhub:us-  
west-1::product/aws/securityhub"}]' \  
  --note '{"Text": "Known issue that is not a risk.", "UpdatedBy": "user1"}' \  
  --severity '{"Label": "LOW"}' \  
  --workflow '{"Status": "RESOLVED"}'
```

出力:

```
{  
  "ProcessedFindings": [  
    {  
      "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/  
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"  
    },  
    {  
      "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/  
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"  
    }  
  ]  
}
```

```

    }
  ],
  "UnprocessedFindings": []
}

```

詳細については、AWS Security Hub ユーザーガイドの「[を使用して結果を更新する BatchUpdateFindings](#)」を参照してください。

例 2: 省略構文を使用して結果を更新するには

次のbatch-update-findings例では、2つの検出結果を更新してメモを追加し、重要度ラベルを変更し、省略構文を使用して解決します。

```

aws securityhub batch-update-findings \
  --finding-identifiers Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" Id="arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-
west-1::product/aws/securityhub" \
  --note Text="Known issue that is not a risk.",UpdatedBy="user1" \
  --severity Label="LOW" \
  --workflow Status="RESOLVED"

```

出力:

```

{
  "ProcessedFindings": [
    {
      "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
    },
    {
      "Id": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/
v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub"
    }
  ],
  "UnprocessedFindings": []
}

```

詳細については、AWS Security Hub ユーザーガイドの「[を使用して結果を更新する BatchUpdateFindings](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchUpdateFindings](#)」の「」を参照してください。

batch-update-standards-control-associations

次の例は、batch-update-standards-control-associations を使用する方法を説明しています。

AWS CLI

有効な標準でコントロールの有効化ステータスを更新するには

次のbatch-update-standards-control-associations例では、指定された標準で CloudTrail.1 を無効にします。

```
aws securityhub batch-update-standards-control-associations \  
  --standards-control-association-updates '[{"SecurityControlId": "CloudTrail.1",  
  "StandardsArn": "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/  
v/1.2.0", "AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable  
to environment"}, {"SecurityControlId": "CloudTrail.1", "StandardsArn":  
"arn:aws:securityhub::standards/cis-aws-foundations-benchmark/v/1.4.0",  
"AssociationStatus": "DISABLED", "UpdatedReason": "Not applicable to  
environment"}]'
```

このコマンドが成功した場合、出力は生成されません。

詳細については、「[Security Hub ユーザーガイド](#)」の「[特定の標準でのコントロールの有効化と無効化](#)」および「[すべての標準でのコントロールの有効化と無効化](#)」を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchUpdateStandardsControlAssociations](#)」の「」を参照してください。

create-action-target

次の例は、create-action-target を使用する方法を説明しています。

AWS CLI

カスタムアクションを作成するには

次のcreate-action-target例では、カスタムアクションを作成します。アクションの名前、説明、識別子を提供します。

```
aws securityhub create-action-target \  
  --name "Send to remediation" \  
  --description "Action to send the finding for remediation tracking" \  
  --id "Remediation"
```

出力:

```
{  
  "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation"  
}
```

詳細については、AWS Security Hub ユーザーガイドの「[カスタムアクションの作成](#)」および [CloudWatch「イベントルールとの関連付け」](#)を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスCreateActionTarget](#)」の「」を参照してください。

create-automation-rule

次のコード例は、create-automation-rule を使用する方法を示しています。

AWS CLI

オートメーションルールを作成するには

次のcreate-automation-rule例では、現在の AWS アカウントと AWS リージョンにオートメーションルールを作成します。Security Hub は、指定された基準に基づいて検出結果をフィルタリングし、一致する検出結果にアクションを適用します。このコマンドを実行できるのは Security Hub 管理者アカウントのみです。

```
aws securityhub create-automation-rule \  
  --actions '[{ \  
    "Type": "FINDING_FIELDS_UPDATE", \  
    "FindingFieldsUpdate": { \  

```

```

    "Severity": { \
      "Label": "HIGH" \
    }, \
    "Note": { \
      "Text": "Known issue that is a risk. Updated by automation rules", \
      "UpdatedBy": "sechub-automation" \
    } \
  } \
}]' \
--criteria '{ \
  "SeverityLabel": [{ \
    "Value": "INFORMATIONAL", \
    "Comparison": "EQUALS" \
  }] \
}' \
--description "A sample rule" \
--no-is-terminal \
--rule-name "sample rule" \
--rule-order 1 \
--rule-status "ENABLED"

```

出力:

```

{
  "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE111111"
}

```

詳細については、Security Hub ユーザーガイドの「[オートメーションルールの作成](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAutomationRule](#)」の「」を参照してください。

create-configuration-policy

次の例は、create-configuration-policy を使用する方法を説明しています。

AWS CLI

設定ポリシーを作成するには

次のcreate-configuration-policy例では、指定された設定で設定ポリシーを作成します。

```
aws securityhub create-configuration-policy \
  --name "SampleConfigurationPolicy" \
  --description "SampleDescription" \
  --configuration-policy '{"SecurityHub": {"ServiceEnabled":
true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-
central-1::standards/aws-foundational-security-best-practices/
v/1.0.0", "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"], "SecurityControlsConfiguration": {"DisabledSecurityControlIdentifiers":
["CloudTrail.2"], "SecurityControlCustomParameters": [{"SecurityControlId":
"ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value":
{"Integer": 15}}}}]}'} \
  --tags '{"Environment": "Prod"}'
```

出力:

```
{
  "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Name": "SampleConfigurationPolicy",
  "Description": "SampleDescription",
  "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
  "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
  "ConfigurationPolicy": {
    "SecurityHub": {
      "ServiceEnabled": true,
      "EnabledStandardIdentifiers": [
        "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
        "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
      ],
      "SecurityControlsConfiguration": {
        "DisabledSecurityControlIdentifiers": [
          "CloudTrail.2"
        ],
        "SecurityControlCustomParameters": [
          {
            "SecurityControlId": "ACM.1",
            "Parameters": {
              "daysToExpiration": {
                "ValueType": "CUSTOM",
                "Value": {
```



```
    "Integer": 15
  }
}
]
}
}
}
```

詳細については、[Security Hub ユーザーガイドの「Security Hub 設定ポリシーの作成と関連付け」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateConfigurationPolicy](#)」の「」を参照してください。

create-finding-aggregator

次の例は、create-finding-aggregator を使用する方法を説明しています。

AWS CLI

検出結果の集約を有効にするには

次のcreate-finding-aggregator例では、検出結果の集約を設定します。これは、米国東部 (バージニア) を集約リージョンとして指定する米国東部 (バージニア) から実行されます。指定されたリージョンのみをリンクし、新しいリージョンを自動的にリンクしないことを示します。リンクされたリージョンとして米国西部 (北カリフォルニア) と米国西部 (オレゴン) を選択します。

```
aws securityhub create-finding-aggregator \
  --region us-east-1 \
  --region-linking-mode SPECIFIED_REGIONS \
  --regions us-west-1,us-west-2
```

出力:

```
{
  "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000",
  "FindingAggregationRegion": "us-east-1",
  "RegionLinkingMode": "SPECIFIED_REGIONS",
```

```
"Regions": "us-west-1,us-west-2"
}
```

詳細については、AWS Security Hub ユーザーガイドの「[結果集約の有効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFindingAggregator](#)」の「」を参照してください。

create-insight

次のコード例は、create-insight を使用する方法を示しています。

AWS CLI

カスタムインサイトを作成するには

次のcreate-insight例では、ロールに関連する重要な検出結果を返す、重要な AWS ロールの検出結果という名前のカスタムインサイトを作成します。

```
aws securityhub create-insight \  
  --filters '{"ResourceType": [{"Comparison": "EQUALS", "Value": "AwsIamRole"}],  
  "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' \  
  --group-by-attribute "ResourceId" \  
  --name "Critical role findings"
```

出力:

```
{  
  "InsightArn": "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

詳細については、「Security Hub ユーザーガイド」の「[カスタムインサイトの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateInsight](#)」の「」を参照してください。

create-members

次の例は、create-members を使用する方法を説明しています。

AWS CLI

アカウントをメンバーアカウントとして追加するには

次のcreate-members例では、リクエストする管理者アカウントにメンバーアカウントとして2つのアカウントを追加します。

```
aws securityhub create-members \  
  --account-details '[{"AccountId": "123456789111"}, {"AccountId":  
  "123456789222"}]'
```

出力:

```
{  
  "UnprocessedAccounts": []  
}
```

詳細については、Security Hub ユーザーガイドの [「管理者アカウントとメンバーアカウントの管理」](#) を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateMembers](#)」の「」を参照してください。

decline-invitations

次の例は、decline-invitations を使用する方法を説明しています。

AWS CLI

メンバーアカウントへの招待を拒否するには

次のdecline-invitations例では、指定された管理者アカウントのメンバーアカウントへの招待を拒否します。メンバーアカウントはリクエスト元のアカウントです。

```
aws securityhub decline-invitations \  
  --account-ids "123456789012"
```

出力:

```
{
```

```
"UnprocessedAccounts": []
}
```

詳細については、AWS Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeclineInvitations](#)」の「」を参照してください。

delete-action-target

次の例は、delete-action-target を使用する方法を説明しています。

AWS CLI

カスタムアクションを削除するには

次のdelete-action-target例では、指定された `arn` によって識別されたカスタムアクションを削除しますARN。

```
aws securityhub delete-action-target \
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
  Remediation"
```

出力:

```
{
  "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
  Remediation"
}
```

詳細については、AWS 「Security Hub ユーザーガイド」の「[カスタムアクションの作成](#)」および CloudWatch 「[イベントルールとの関連付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteActionTarget](#)」の「」を参照してください。

delete-configuration-policy

次の例は、delete-configuration-policy を使用する方法を説明しています。

AWS CLI

設定ポリシーを削除するには

次のdelete-configuration-policy例では、指定された設定ポリシーを削除します。

```
aws securityhub delete-configuration-policy \  
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-  
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

このコマンドでは何も出力されません。

詳細については、[Security Hub ユーザーガイドの「Security Hub 設定ポリシーの削除と関連付け解除」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteConfigurationPolicy](#)」の「」を参照してください。

delete-finding-aggregator

次の例は、delete-finding-aggregator を使用する方法を説明しています。

AWS CLI

検出結果の集約を停止するには

次のdelete-finding-aggregator例では、集計の検出を停止します。これは、集約リージョンである米国東部 (バージニア) から実行されます。

```
aws securityhub delete-finding-aggregator \  
  --region us-east-1 \  
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-  
aggregator/123e4567-e89b-12d3-a456-426652340000
```

このコマンドでは何も出力されません。

詳細については、AWS 「Security Hub ユーザーガイド」の[「検出結果の集約の停止」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFindingAggregator](#)」の「」を参照してください。

delete-insight

次の例は、delete-insight を使用する方法を説明しています。

AWS CLI

カスタムインサイトを削除するには

次のdelete-insight例では、指定された を使用してカスタムインサイトを削除しますARN。

```
aws securityhub delete-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

出力:

```
{  
  "InsightArn": "arn:aws:securityhub:eu-  
  central-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-  
  EXAMPLE11111"  
}
```

詳細については、AWS Security Hub ユーザーガイドの [「カスタムインサイトの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteInsight](#)」の「」を参照してください。

delete-invitations

次のコード例は、delete-invitations を使用する方法を示しています。

AWS CLI

メンバーアカウントへの招待を削除するには

次のdelete-invitations例では、指定された管理者アカウントのメンバーアカウントになるための招待を削除します。メンバーアカウントはリクエスト元のアカウントです。

```
aws securityhub delete-invitations \  
  --account-ids "123456789012"
```

出力:

```
{
  "UnprocessedAccounts": []
}
```

詳細については、Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInvitations](#)」の「」を参照してください。

delete-members

次のコード例は、delete-members を使用する方法を示しています。

AWS CLI

メンバーアカウントを削除するには

次のdelete-members例では、リクエストする管理者アカウントから指定されたメンバーアカウントを削除します。

```
aws securityhub delete-members \
  --account-ids "123456789111" "123456789222"
```

出力:

```
{
  "UnprocessedAccounts": []
}
```

詳細については、Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMembers](#)」の「」を参照してください。

describe-action-targets

次の例は、describe-action-targets を使用する方法を説明しています。

AWS CLI

カスタムアクションの詳細を取得するには

次のdescribe-action-targets例では、指定された `arn` によって識別されたカスタムアクションに関する情報を取得しますARN。

```
aws securityhub describe-action-targets \  
  --action-target-arns "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
  Remediation"
```

出力:

```
{  
  "ActionTargets": [  
    {  
      "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/  
      custom/Remediation",  
      "Description": "Action to send the finding for remediation tracking",  
      "Name": "Send to remediation"  
    }  
  ]  
}
```

詳細については、AWS Security Hub ユーザーガイドの [「カスタムアクションの作成」](#) および [CloudWatch 「イベントルールとの関連付け」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeActionTargets](#)」の「」を参照してください。

describe-hub

次の例は、describe-hub を使用する方法を説明しています。

AWS CLI

ハブリソースに関する情報を取得するには

次のdescribe-hub例では、指定されたハブリソースのサブスクリプション日を返します。ハブリソースは `arn` によって識別されますARN。

```
aws securityhub describe-hub \  
  --hub-arns "arn:aws:securityhub:us-west-1:123456789012:hub/123456789012"
```



```
--hub-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

出力:

```
{
  "HubArn": "arn:aws:securityhub:us-west-1:123456789012:hub/default",
  "SubscribedAt": "2019-11-19T23:15:10.046Z"
}
```

詳細については、「ユーザーガイド」の[AWS「: SecurityHub:::Hub」](#)を参照してください。

AWS CloudFormation

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeHub](#)」の「」を参照してください。

describe-organization-configuration

次の例は、describe-organization-configuration を使用する方法を説明しています。

AWS CLI

Security Hub が組織に対してどのように設定されているかを表示するには

次のdescribe-organization-configuration例では、Security Hub で組織を設定する方法に関する情報を返します。この例では、組織は中央設定を使用します。このコマンドを実行できるのは Security Hub 管理者アカウントのみです。

```
aws securityhub describe-organization-configuration
```

出力:

```
{
  "AutoEnable": false,
  "MemberAccountLimitReached": false,
  "AutoEnableStandards": "NONE",
  "OrganizationConfiguration": {
    "ConfigurationType": "LOCAL",
    "Status": "ENABLED",
    "StatusMessage": "Central configuration has been enabled successfully"
  }
}
```

```
}
```

詳細については、AWS Security Hub ユーザーガイドの[AWS 「Organizations でのアカウントの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOrganizationConfiguration](#)」の「」を参照してください。

describe-products

次の例は、describe-products を使用する方法を説明しています。

AWS CLI

利用可能な製品統合に関する情報を返すには

次のdescribe-products例では、使用可能な製品統合を一度に 1 つずつ返します。

```
aws securityhub describe-products \  
  --max-results 1
```

出力:

```
{  
  "NextToken": "U2FsdGVkX18vvP10qb7RD1rWRWVFBJI46M0IAb+nZmRJmR15NoRi2gm13sdQEn30/  
pq/78dGs+bKpgA+7HMPH00qX33/zoRI+uIG/F9yLNhc0r0WzFUdy36JcXLQji3Rpnn/  
cD1SVkGA98qI3zPOSDg==",  
  "Products": [  
    {  
      "ProductArn": "arn:aws:securityhub:us-west-1:123456789333:product/  
crowdstrike/crowdstrike-falcon",  
      "ProductName": "CrowdStrike Falcon",  
      "CompanyName": "CrowdStrike",  
      "Description": "CrowdStrike Falcon's single lightweight sensor unifies  
next-gen antivirus, endpoint detection and response, and 24/7 managed hunting, via  
the cloud.",  
      "Categories": [  
        "Endpoint Detection and Response (EDR)",  
        "AV Scanning and Sandboxing",  
        "Threat Intelligence Feeds and Reports",  
        "Endpoint Forensics",  
        "Network Forensics"  
      ],  
    },  
  ],  
}
```

```

    "IntegrationTypes": [
      "SEND_FINDINGS_TO_SECURITY_HUB"
    ],
    "MarketplaceUrl": "https://aws.amazon.com/marketplace/seller-profile?id=a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "ActivationUrl": "https://falcon.crowdstrike.com/support/documentation",
    "ProductSubscriptionResourcePolicy": "{\"Version\": \"2012-10-17\", \"Statement\": [{\"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"123456789333\"}, \"Action\": [\"securityhub:BatchImportFindings\"], \"Resource\": \"arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon\", \"Condition\": {\"StringEquals\": {\"securityhub:TargetAccount\": \"123456789012\"}}}, {\"Effect\": \"Allow\", \"Principal\": {\"AWS\": \"123456789012\"}, \"Action\": [\"securityhub:BatchImportFindings\"], \"Resource\": \"arn:aws:securityhub:us-west-1:123456789333:product/crowdstrike/crowdstrike-falcon\", \"Condition\": {\"StringEquals\": {\"securityhub:TargetAccount\": \"123456789012\"}}}]}"
  }
]
}

```

詳細については、AWS Security Hub ユーザーガイドの「[製品統合の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeProducts](#)」の「」を参照してください。

describe-standards-controls

次の例は、describe-standards-controls を使用する方法を説明しています。

AWS CLI

有効な標準でコントロールのリストをリクエストするには

次のdescribe-standards-controls例では、リクエストアカウントの PCI DSS 標準へのサブスクリプションのコントロールのリストをリクエストします。リクエストは、一度に 2 つのコントロールを返します。

```

aws securityhub describe-standards-controls \
  --standards-subscription-arn "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1" \
  --max-results 2

```

出力:

```
{
  "Controls": [
    {
      "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.AutoScaling.1",
      "ControlStatus": "ENABLED",
      "ControlStatusUpdatedAt": "2020-05-15T18:49:04.473000+00:00",
      "ControlId": "PCI.AutoScaling.1",
      "Title": "Auto scaling groups associated with a load balancer should use
health checks",
      "Description": "This AWS control checks whether your Auto Scaling groups
that are associated with a load balancer are using Elastic Load Balancing health
checks.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.AutoScaling.1/remediation",
      "SeverityRating": "LOW",
      "RelatedRequirements": [
        "PCI DSS 2.2"
      ]
    },
    {
      "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.CW.1",
      "ControlStatus": "ENABLED",
      "ControlStatusUpdatedAt": "2020-05-15T18:49:04.498000+00:00",
      "ControlId": "PCI.CW.1",
      "Title": "A log metric filter and alarm should exist for usage of the
\"root\" user",
      "Description": "This control checks for the CloudWatch metric
filters using the following pattern { $.userIdentity.type = \"Root\" &&
$.userIdentity.invokedBy NOT EXISTS && $.eventType != \"AwsServiceEvent\" }
It checks that the log group name is configured for use with active multi-
region CloudTrail, that there is at least one Event Selector for a Trail with
IncludeManagementEvents set to true and ReadWriteType set to All, and that there is
at least one active subscriber to an SNS topic associated with the alarm.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
PCI.CW.1/remediation",
      "SeverityRating": "MEDIUM",
      "RelatedRequirements": [
        "PCI DSS 7.2.1"
      ]
    }
  ]
}
```

```
  ],
  "NextToken": "U2FsdGvKX1+eNkPoZHVl11ip5HUYQPWSWZGmftcmJiHL8JoKEsCDuaKayiPDyLK
+LiTkShveo0dvfxXCk0BaGhohIXhsIedN+LSjQV/
17kfCfJcq4PziNC1N9xe9aq2pjlLVZnznTfSImrodT5bRNHe4fELCQq/z+5ka
+5Lzmc11axcwTd5lKgQyQqmUVoeriHZhyIiBgWKf7oNYdBVG80EortVWvSkoUTt
+B2ThcnC7l43kI0UNx1kZ6sc64AsW"
}
```

詳細については、AWS Security Hub ユーザーガイドの「[コントロールの詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeStandardsControls](#)」の「」を参照してください。

describe-standards

次のコード例は、describe-standards を使用する方法を示しています。

AWS CLI

使用可能な標準のリストを返すには

次のdescribe-standards例では、使用可能な標準のリストを返します。

```
aws securityhub describe-standards
```

出力:

```
{
  "Standards": [
    {
      "StandardsArn": "arn:aws:securityhub:us-west-1::standards/aws-
foundational-security-best-practices/v/1.0.0",
      "Name": "AWS Foundational Security Best Practices v1.0.0",
      "Description": "The AWS Foundational Security Best Practices standard
is a set of automated security checks that detect when AWS accounts and deployed
resources do not align to security best practices. The standard is defined by AWS
security experts. This curated set of controls helps improve your security posture
in AWS, and cover AWS's most popular and foundational services.",
      "EnabledByDefault": true
    },
    {
```

```

    "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
    "Name": "CIS AWS Foundations Benchmark v1.2.0",
    "Description": "The Center for Internet Security (CIS) AWS Foundations
Benchmark v1.2.0 is a set of security configuration best practices for AWS. This
Security Hub standard automatically checks for your compliance readiness against a
subset of CIS requirements.",
    "EnabledByDefault": true
  },
  {
    "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
    "Name": "PCI DSS v3.2.1",
    "Description": "The Payment Card Industry Data Security Standard (PCI
DSS) v3.2.1 is an information security standard for entities that store, process,
and/or transmit cardholder data. This Security Hub standard automatically checks
for your compliance readiness against a subset of PCI DSS requirements.",
    "EnabledByDefault": false
  }
]
}

```

詳細については、[「Security Hub ユーザーガイド」](#)のAWS「[Security Hub のセキュリティ標準](#)」を参照してください。AWS

- API 詳細については、AWS CLI「[コマンドリファレンスDescribeStandards](#)」の「」を参照してください。

disable-import-findings-for-product

次のコード例は、`disable-import-findings-for-product` を使用する方法を示しています。

AWS CLI

製品統合からの調査結果の受信を停止するには

次の`disable-import-findings-for-product`例では、製品統合への指定されたサブスクリプションの検出結果の流れを無効にします。

```

aws securityhub disable-import-findings-for-product \
  --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon"

```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの「[製品統合の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableImportFindingsForProduct](#)」の「」を参照してください。

disable-organization-admin-account

次の例は、`disable-organization-admin-account` を使用する方法を説明しています。

AWS CLI

Security Hub 管理者アカウントを削除するには

次の`disable-organization-admin-account`例では、指定されたアカウントの AWS Organizations の Security Hub 管理者アカウントとしての割り当てを取り消します。

```
aws securityhub disable-organization-admin-account \  
  --admin-account-id 777788889999
```

このコマンドでは何も出力されません。

詳細については、[Security Hub ユーザーガイドの「Security Hub 管理者アカウントの指定」](#)を参照してください。 AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableOrganizationAdminAccount](#)」の「」を参照してください。

disable-security-hub

次のコード例は、`disable-security-hub` を使用する方法を示しています。

AWS CLI

AWS Security Hub を無効にするには

次の`disable-security-hub`例では、リクエスト元のアカウントの AWS Security Hub を無効にします。

```
aws securityhub disable-security-hub
```

このコマンドでは何も出力されません。

詳細については、[AWS Security Hub ユーザーガイドの「Security Hub の無効化」](#)を参照してください。 AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DisableSecurityHub](#)」の「」を参照してください。

disassociate-from-administrator-account

次のコード例は、`disassociate-from-administrator-account` を使用する方法を示しています。

AWS CLI

管理者アカウントからの関連付けを解除するには

次の`disassociate-from-administrator-account`例では、リクエスト元のアカウントと現在の管理者アカウントの関連付けを解除します。

```
aws securityhub disassociate-from-administrator-account
```

このコマンドでは何も出力されません。

詳細については、Security Hub ユーザーガイドの[「管理者アカウントとメンバーアカウントの管理」](#)を参照してください。 AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateFromAdministratorAccount](#)」の「」を参照してください。

disassociate-from-master-account

次の例は、`disassociate-from-master-account` を使用する方法を説明しています。

AWS CLI

管理者アカウントとの関連付けを解除するには

次の`disassociate-from-master-account`例では、リクエスト元のアカウントと現在の管理者アカウントの関連付けを解除します。

```
aws securityhub disassociate-from-master-account
```


このコマンドでは何も出力されません。

詳細については、Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateFromMasterAccount](#)」の「」を参照してください。

disassociate-members

次のコード例は、disassociate-members を使用する方法を示しています。

AWS CLI

メンバーアカウントの関連付けを解除するには

次のdisassociate-members例では、指定されたメンバーアカウントをリクエストする管理者アカウントから関連付け解除します。

```
aws securityhub disassociate-members \  
  --account-ids "123456789111" "123456789222"
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateMembers](#)」の「」を参照してください。

enable-import-findings-for-product

次のコード例は、enable-import-findings-for-product を使用する方法を示しています。

AWS CLI

製品統合から結果の受信を開始するには

次のenable-import-findings-for-product例では、指定された製品統合からの結果の流れを有効にします。

```
aws securityhub enable-import-findings-for-product \  
  --product-arns "arn:aws:securityhub:::product/arn:aws:iam::123456789111:root" "arn:aws:iam::123456789111:root"
```

```
--product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/crowdstrike-falcon"
```

出力:

```
{
  "ProductSubscriptionArn": "arn:aws:securityhub:us-east-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon"
}
```

詳細については、AWS Security Hub ユーザーガイドの「[製品統合の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableImportFindingsForProduct](#)」の「」を参照してください。

enable-organization-admin-account

次のコード例は、enable-organization-admin-account を使用する方法を示しています。

AWS CLI

組織アカウントを Security Hub 管理者アカウントとして指定するには

次のenable-organization-admin-account例では、指定されたアカウントを Security Hub 管理者アカウントとして指定します。

```
aws securityhub enable-organization-admin-account \
  --admin-account-id 777788889999
```

このコマンドでは何も出力されません。

詳細については、[Security Hub ユーザーガイドの「Security Hub 管理者アカウントの指定」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableOrganizationAdminAccount](#)」の「」を参照してください。

enable-security-hub

次のコード例は、enable-security-hub を使用する方法を示しています。

AWS CLI

AWS Security Hub を有効にするには

次のenable-security-hub例では、リクエスト元のアカウントの AWS Security Hub を有効にします。デフォルトの標準を有効にするように Security Hub を設定します。ハブリソースの場合、タグ Securityに値を割り当てますDepartment。

```
aws securityhub enable-security-hub \  
  --enable-default-standards \  
  --tags '{"Department": "Security"}'
```

このコマンドでは何も出力されません。

詳細については、[「Security Hub ユーザーガイド」の「Security Hub の有効化AWS」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[EnableSecurityHub](#)」の「」を参照してください。

get-administrator-account

次の例は、get-administrator-account を使用する方法を説明しています。

AWS CLI

管理者アカウントに関する情報を取得するには

次のget-administrator-account例では、リクエスト元のアカウントの管理者アカウントに関する情報を取得します。

```
aws securityhub get-administrator-account
```

出力:

```
{  
  "Master": {  
    "AccountId": "123456789012",  
    "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",  
    "InvitedAt": 2020-06-01T20:21:18.042000+00:00,  
    "MemberStatus": "ASSOCIATED"  }  
}
```

```
}  
}
```

詳細については、Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAdministratorAccount](#)」の「」を参照してください。

get-configuration-policy-association

次の例は、get-configuration-policy-association を使用する方法を説明しています。

AWS CLI

ターゲットの設定関連付けの詳細を取得するには

次のget-configuration-policy-association例では、指定されたターゲットの関連付けの詳細を取得します。ターゲットのアカウント ID、組織単位 ID、またはルート ID を指定できます。

```
aws securityhub get-configuration-policy-association \  
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

出力:

```
{  
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
  "TargetId": "ou-6hi7-8j91k12m",  
  "TargetType": "ORGANIZATIONAL_UNIT",  
  "AssociationType": "APPLIED",  
  "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",  
  "AssociationStatus": "SUCCESS",  
  "AssociationStatusMessage": "Association applied successfully on this target."  
}
```

詳細については、[Security Hub ユーザーガイドの「Security Hub 設定ポリシーの表示」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConfigurationPolicyAssociation](#)」の「」を参照してください。

get-configuration-policy

次のコード例は、get-configuration-policy を使用する方法を示しています。

AWS CLI

設定ポリシーの詳細を表示するには

次のget-configuration-policy例では、指定された設定ポリシーの詳細を取得します。

```
aws securityhub get-configuration-policy \  
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

出力:

```
{  
  "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "Id": "ce5ed1e7-9639-4e2f-9313-fa87fcef944b",  
  "Name": "SampleConfigurationPolicy",  
  "Description": "SampleDescription",  
  "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",  
  "CreatedAt": "2023-11-28T20:28:04.494000+00:00",  
  "ConfigurationPolicy": {  
    "SecurityHub": {  
      "ServiceEnabled": true,  
      "EnabledStandardIdentifiers": [  
        "arn:aws:securityhub:eu-central-1::standards/aws-foundational-  
security-best-practices/v/1.0.0",  
        "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/  
v/1.2.0"  
      ],  
      "SecurityControlsConfiguration": {  
        "DisabledSecurityControlIdentifiers": [  
          "CloudTrail.2"  
        ],  
        "SecurityControlCustomParameters": [  
          {  
            "SecurityControlId": "ACM.1",  
            "Parameters": {  
              "daysToExpiration": {  
                "ValueType": "CUSTOM",  
                "Value": {
```

```
    "Integer": 15
  }
}
}
}
]
}
}
}
}
```

詳細については、[「Security Hub ユーザーガイド」の「Security Hub 設定ポリシーの表示」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConfigurationPolicy](#)」の「」を参照してください。

get-enabled-standards

次のコード例は、get-enabled-standards を使用する方法を示しています。

AWS CLI

有効な標準に関する情報を取得するには

次のget-enabled-standards例では、PCI DSS 標準に関する情報を取得します。

```
aws securityhub get-enabled-standards \
  --standards-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

出力:

```
{
  "StandardsSubscriptions": [
    {
      "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/v/3.2.1",
      "StandardsInput": { },
      "StandardsStatus": "READY",
      "StandardsSubscriptionArn": "arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/v/3.2.1"
    }
  ]
}
```

```
]
}
```

詳細については、[「Security Hub ユーザーガイド」](#)の AWS [「Security Hub のセキュリティ標準」](#)を参照してください。AWS

- API 詳細については、AWS CLI [「コマンドリファレンスGetEnabledStandards」](#)の「」を参照してください。

get-finding-aggregator

次の例は、get-finding-aggregator を使用する方法を説明しています。

AWS CLI

現在の検出結果集約設定を取得するには

次のget-finding-aggregator例では、現在の検出結果集約設定を取得します。

```
aws securityhub get-finding-aggregator \
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
  aggregator/123e4567-e89b-12d3-a456-426652340000
```

出力:

```
{
  "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
  aggregator/123e4567-e89b-12d3-a456-426652340000",
  "FindingAggregationRegion": "us-east-1",
  "RegionLinkingMode": "SPECIFIED_REGIONS",
  "Regions": "us-west-1,us-west-2"
}
```

詳細については、AWS Security Hub ユーザーガイドの [「現在の検出結果集約設定の表示」](#)を参照してください。

- API 詳細については、AWS CLI [「コマンドリファレンスGetFindingAggregator」](#)の「」を参照してください。

get-finding-history

次の例は、get-finding-history を使用する方法を説明しています。

AWS CLI

検出結果の履歴を取得するには

次の`get-finding-history`例では、指定された検出結果の過去 90 日間の履歴を取得します。この例では、結果は検出結果履歴の 2 つのレコードに制限されています。

```
aws securityhub get-finding-history \
  --finding-identifier Id="arn:aws:securityhub:us-
  east-1:123456789012:security-control/S3.17/finding/a1b2c3d4-5678-90ab-cdef-
  EXAMPLE11111",ProductArn="arn:aws:securityhub:us-east-1::product/aws/securityhub"
```

出力:

```
{
  "Records": [
    {
      "FindingIdentifier": {
        "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
        S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
        securityhub"
      },
      "UpdateTime": "2023-06-02T03:15:25.685000+00:00",
      "FindingCreated": false,
      "UpdateSource": {
        "Type": "BATCH_IMPORT_FINDINGS",
        "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
      },
      "Updates": [
        {
          "UpdatedField": "Compliance.RelatedRequirements",
          "OldValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
          SC-12(3)\",\"NIST.800-53.r5 SC-12(6)\",\"NIST.800-53.r5 CM-3(6)\",\"NIST.800-53.r5
          SC-13\", \"NIST.800-53.r5 SC-28\", \"NIST.800-53.r5 SC-28(1)\", \"NIST.800-53.r5
          SC-7(10)\"]",
          "NewValue": "[\"NIST.800-53.r5 SC-12(2)\",\"NIST.800-53.r5
          CM-3(6)\",\"NIST.800-53.r5 SC-13\", \"NIST.800-53.r5 SC-28\", \"NIST.800-53.r5
          SC-28(1)\", \"NIST.800-53.r5 SC-7(10)\", \"NIST.800-53.r5 CA-9(1)\", \"NIST.800-53.r5
          SI-7(6)\", \"NIST.800-53.r5 AU-9\"]"
        },
        {
          "UpdatedField": "LastObservedAt",
```



```
        "OldValue": "2023-06-01T09:15:38.587Z",
        "NewValue": "2023-06-02T03:15:22.946Z"
      },
      {
        "UpdatedField": "UpdatedAt",
        "OldValue": "2023-06-01T09:15:31.049Z",
        "NewValue": "2023-06-02T03:15:14.861Z"
      },
      {
        "UpdatedField": "ProcessedAt",
        "OldValue": "2023-06-01T09:15:41.058Z",
        "NewValue": "2023-06-02T03:15:25.685Z"
      }
    ]
  },
  {
    "FindingIdentifier": {
      "Id": "arn:aws:securityhub:us-east-1:123456789012:security-control/
S3.17/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/
securityhub"
    },
    "UpdateTime": "2023-05-23T02:06:51.518000+00:00",
    "FindingCreated": "true",
    "UpdateSource": {
      "Type": "BATCH_IMPORT_FINDINGS",
      "Identity": "arn:aws:securityhub:us-east-1::product/aws/securityhub"
    },
    "Updates": []
  }
]
}
```

詳細については、AWS Security Hub ユーザーガイドの「[検出結果履歴](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFindingHistory](#)」の「」を参照してください。

get-findings

次の例は、get-findings を使用する方法を説明しています。

AWS CLI

例 1: 特定の標準に対して生成された結果を返すには

次のget-findings例では、PCI DSS 標準の結果を返します。

```
aws securityhub get-findings \  
  --filters '{"GeneratorId":[{"Value": "pci-dss", "Comparison": "PREFIX"}]}' \  
  --max-items 1
```

出力:

```
{  
  "Findings": [  
    {  
      "SchemaVersion": "2018-10-08",  
      "Id": "arn:aws:securityhub:eu-central-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ProductArn": "arn:aws:securityhub:us-west-1::product/aws/securityhub",  
      "GeneratorId": "pci-dss/v/3.2.1/PCI.Lambda.2",  
      "AwsAccountId": "123456789012",  
      "Types": [  
        "Software and Configuration Checks/Industry and Regulatory Standards/PCI-DSS"  
      ],  
      "FindingProviderFields": {  
        "Severity": {  
          "Original": 0,  
          "Label": "INFORMATIONAL"  
        },  
        "Types": [  
          "Software and Configuration Checks/Industry and Regulatory Standards/PCI-DSS"  
        ]  
      },  
      "FirstObservedAt": "2020-06-02T14:02:49.159Z",  
      "LastObservedAt": "2020-06-02T14:02:52.397Z",  
      "CreatedAt": "2020-06-02T14:02:49.159Z",  
      "UpdatedAt": "2020-06-02T14:02:52.397Z",  
      "Severity": {  
        "Original": 0,  
        "Label": "INFORMATIONAL",  
        "Normalized": 0  
      }  
    }  
  ]  
}
```

```
    },
    "Title": "PCI.Lambda.2 Lambda functions should be in a VPC",
    "Description": "This AWS control checks whether a Lambda function is in
a VPC.",
    "Remediation": {
      "Recommendation": {
        "Text": "For directions on how to fix this issue, please consult
the AWS Security Hub PCI DSS documentation.",
        "Url": "https://docs.aws.amazon.com/console/securityhub/
PCI.Lambda.2/remediation"
      }
    },
    "ProductFields": {
      "StandardsArn": "arn:aws:securityhub::standards/pci-dss/v/3.2.1",
      "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1",
      "ControlId": "PCI.Lambda.2",
      "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/PCI.Lambda.2/remediation",
      "RelatedAWSResources:0/name": "securityhub-lambda-inside-
vpc-0e904a3b",
      "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
      "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/pci-dss/v/3.2.1/PCI.Lambda.2",
      "aws/securityhub/SeverityLabel": "INFORMATIONAL",
      "aws/securityhub/ProductName": "Security Hub",
      "aws/securityhub/CompanyName": "AWS",
      "aws/securityhub/FindingId": "arn:aws:securityhub:eu-
central-1::product/aws/securityhub/arn:aws:securityhub:eu-
central-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
  },
  "Resources": [
    {
      "Type": "AwsAccount",
      "Id": "AWS:::Account:123456789012",
      "Partition": "aws",
      "Region": "us-west-1"
    }
  ],
  "Compliance": {
    "Status": "PASSED",
    "RelatedRequirements": [
      "PCI DSS 1.2.1",
```

```

        "PCI DSS 1.3.1",
        "PCI DSS 1.3.2",
        "PCI DSS 1.3.4"
    ]
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ARCHIVED"
}
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ=="
}

```

例 2: ワークフローステータスが の重大度の検出結果を返すには NOTIFIED

次のget-findings例では、重要度ラベル値が CRITICALで、ワークフローステータスが である結果を返しますNOTIFIED。結果は、信頼度の値で降順にソートされます。

```

aws securityhub get-findings \
  --filters '{"SeverityLabel":[{"Value":
"CRITICAL","Comparison":"EQUALS"}],"WorkflowStatus":
[{"Value":"NOTIFIED","Comparison":"EQUALS"}]}' \
  --sort-criteria '{ "Field": "Confidence", "SortOrder": "desc"}' \
  --max-items 1

```

出力:

```

{
  "Findings": [
    {
      "SchemaVersion": "2018-10-08",
      "Id": "arn:aws:securityhub:us-west-1: 123456789012:subscription/cis-aws-
foundations-benchmark/v/1.2.0/1.13/finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/securityhub",
      "GeneratorId": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0/rule/1.13",
      "AwsAccountId": "123456789012",
      "Types": [
        "Software and Configuration Checks/Industry and Regulatory
Standards/CIS AWS Foundations Benchmark"
      ],
    }
  ],
}

```

```
"FindingProviderFields" {
  "Severity": {
    "Original": 90,
    "Label": "CRITICAL"
  },
  "Types": [
    "Software and Configuration Checks/Industry and Regulatory
Standards/CIS AWS Foundations Benchmark"
  ]
},
"FirstObservedAt": "2020-05-21T20:16:34.752Z",
"LastObservedAt": "2020-06-09T08:16:37.171Z",
"CreatedAt": "2020-05-21T20:16:34.752Z",
"UpdatedAt": "2020-06-09T08:16:36.430Z",
"Severity": {
  "Original": 90,
  "Label": "CRITICAL",
  "Normalized": 90
},
"Title": "1.13 Ensure MFA is enabled for the \"root\" account",
"Description": "The root account is the most privileged user in an AWS
account. MFA adds an extra layer of protection on top of a user name and password.
With MFA enabled, when a user signs in to an AWS website, they will be prompted for
their user name and password as well as for an authentication code from their AWS
MFA device.",
"Remediation": {
  "Recommendation": {
    "Text": "For directions on how to fix this issue, please consult
the AWS Security Hub CIS documentation.",
    "Url": "https://docs.aws.amazon.com/console/securityhub/
standards-cis-1.13/remediation"
  }
},
"ProductFields": {
  "StandardsGuideArn": "arn:aws:securityhub:::ruleset/cis-aws-
foundations-benchmark/v/1.2.0",
  "StandardsGuideSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0",
  "RuleId": "1.13",
  "RecommendationUrl": "https://docs.aws.amazon.com/console/
securityhub/standards-cis-1.13/remediation",
  "RelatedAWSResources:0/name": "securityhub-root-account-mfa-
enabled-5pftha",
  "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
```

```
    "StandardsControlArn": "arn:aws:securityhub:us-
west-1:123456789012:control/cis-aws-foundations-benchmark/v/1.2.0/1.13",
    "aws/securityhub/SeverityLabel": "CRITICAL",
    "aws/securityhub/ProductName": "Security Hub",
    "aws/securityhub/CompanyName": "AWS",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-
west-1::product/aws/securityhub/arn:aws:securityhub:us-
west-1:123456789012:subscription/cis-aws-foundations-benchmark/v/1.2.0/1.13/finding/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  },
  "Resources": [
    {
      "Type": "AwsAccount",
      "Id": "AWS:::Account:123456789012",
      "Partition": "aws",
      "Region": "us-west-1"
    }
  ],
  "Compliance": {
    "Status": "FAILED"
  },
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NOTIFIED"
  },
  "RecordState": "ACTIVE"
}
]
```

詳細については、AWS Security Hub ユーザーガイドの「[結果のフィルタリングとグループ化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFindings](#)」の「」を参照してください。

get-insight-results

次の例は、get-insight-results を使用する方法を説明しています。

AWS CLI

インサイトの結果を取得するには

次のget-insight-results例では、指定された を持つインサイトのインサイト結果のリストを返しますARN。

```
aws securityhub get-insight-results \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

出力:

```
{  
  "InsightResults": {  
    "GroupByAttribute": "ResourceId",  
    "InsightArn": "arn:aws:securityhub:us-  
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111",  
    "ResultValues": [  
      {  
        "Count": 10,  
        "GroupByAttributeValue": "AWS:::Account:123456789111"  
      },  
      {  
        "Count": 3,  
        "GroupByAttributeValue": "AWS:::Account:123456789222"  
      }  
    ]  
  }  
}
```

詳細については、AWS「[Security Hub ユーザーガイド](#)」の「[インサイトの結果と結果の表示と対応](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetInsightResults](#)」の「」を参照してください。

get-insights

次の例は、get-insights を使用する方法を説明しています。

AWS CLI

インサイトの詳細を取得するには

次のget-insights例では、指定された を使用してインサイトの設定の詳細を取得します
ARN。

```
aws securityhub get-insights \  
  --insight-arns "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

出力:

```
{  
  "Insights": [  
    {  
      "Filters": {  
        "ResourceType": [  
          {  
            "Comparison": "EQUALS",  
            "Value": "AwsIamRole"  
          }  
        ],  
        "SeverityLabel": [  
          {  
            "Comparison": "EQUALS",  
            "Value": "CRITICAL"  
          }  
        ],  
      },  
      "GroupByAttribute": "ResourceId",  
      "InsightArn": "arn:aws:securityhub:us-  
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111",  
      "Name": "Critical role findings"  
    }  
  ]  
}
```

詳細については、[AWS Security Hub ユーザーガイドの「Security Hub のインサイト」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInsights](#)」の「」を参照してください。

get-invitations-count

次の例は、get-invitations-count を使用する方法を説明しています。

AWS CLI

受け入れられなかった招待の数を取得するには

次のget-invitations-count例では、リクエスト元のアカウントが拒否または応答しなかった招待の数を取得します。

```
aws securityhub get-invitations-count
```

出力:

```
{
  "InvitationsCount": 3
}
```

詳細については、AWS Security Hub ユーザーガイドの [「管理者アカウントとメンバーアカウントの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInvitationsCount](#)」の「」を参照してください。

get-master-account

次のコード例は、get-master-account を使用する方法を示しています。

AWS CLI

管理者アカウントに関する情報を取得するには

次のget-master-account例では、リクエスト元のアカウントの管理者アカウントに関する情報を取得します。

```
aws securityhub get-master-account
```

出力:

```
{
  "Master": {
    "AccountId": "123456789012",
    "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
    "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
    "MemberStatus": "ASSOCIATED"
  }
}
```

詳細については、Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMasterAccount](#)」の「」を参照してください。

get-members

次の例は、get-members を使用する方法を説明しています。

AWS CLI

選択したメンバーアカウントに関する情報を取得するには

次のget-members例では、指定されたメンバーアカウントに関する情報を取得します。

```
aws securityhub get-members \
  --account-ids "444455556666" "777788889999"
```

出力:

```
{
  "Members": [
    {
      "AccountId": "123456789111",
      "AdministratorId": "123456789012",
      "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
      "MasterId": "123456789012",
      "MemberStatus": "ASSOCIATED",
      "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
    },
    {
```

```
    "AccountId": "123456789222",
    "AdministratorId": "123456789012",
    "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
    "MasterId": "123456789012",
    "MemberStatus": "ASSOCIATED",
    "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
  }
],
"UnprocessedAccounts": [ ]
}
```

詳細については、AWS Security Hub ユーザーガイドの [「管理者アカウントとメンバーアカウントの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetMembers](#)」の「」を参照してください。

get-security-control-definition

次のコード例は、get-security-control-definition を使用する方法を示しています。

AWS CLI

セキュリティコントロール定義の詳細を取得するには

次のget-security-control-definition例では、Security Hub セキュリティコントロールの定義の詳細を取得します。詳細には、コントロールタイトル、説明、リージョンの可用性、パラメータ、その他の情報が含まれます。

```
aws securityhub get-security-control-definition \
  --security-control-id ACM.1
```

出力:

```
{
  "SecurityControlDefinition": {
    "SecurityControlId": "ACM.1",
    "Title": "Imported and ACM-issued certificates should be renewed after a
specified time period",
    "Description": "This control checks whether an AWS Certificate Manager
(ACM) certificate is renewed within the specified time period. It checks both
```

```

imported certificates and certificates provided by ACM. The control fails if the
certificate isn't renewed within the specified time period. Unless you provide a
custom parameter value for the renewal period, Security Hub uses a default value of
30 days.",
  "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/ACM.1/
remediation",
  "SeverityRating": "MEDIUM",
  "CurrentRegionAvailability": "AVAILABLE",
  "ParameterDefinitions": {
    "daysToExpiration": {
      "Description": "Number of days within which the ACM certificate must
be renewed",
      "ConfigurationOptions": {
        "Integer": {
          "DefaultValue": 30,
          "Min": 14,
          "Max": 365
        }
      }
    }
  }
}

```

詳細については、Security Hub ユーザーガイドの「[カスタムコントロールパラメータ](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSecurityControlDefinition](#)」の「」を参照してください。

invite-members

次のコード例は、invite-members を使用する方法を示しています。

AWS CLI

メンバーアカウントに招待を送信するには

次のinvite-members例では、指定されたメンバーアカウントに招待を送信します。

```

aws securityhub invite-members \
  --account-ids "123456789111" "123456789222"

```

出力:

```
{
  "UnprocessedAccounts": []
}
```

詳細については、Security Hub ユーザーガイドの [「管理者アカウントとメンバーアカウントの管理」](#) を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [InviteMembers](#)」の「」を参照してください。

list-automation-rules

次の例は、list-automation-rules を使用する方法を説明しています。

AWS CLI

オートメーションルールのリストを表示するには

次のlist-automation-rules例では、AWS アカウントのオートメーションルールを一覧表示します。このコマンドを実行できるのは Security Hub 管理者アカウントのみです。

```
aws securityhub list-automation-rules \
  --max-results 3 \
  --next-token NULL
```

出力:

```
{
  "AutomationRulesMetadata": [
    {
      "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "RuleStatus": "ENABLED",
      "RuleOrder": 1,
      "RuleName": "Suppress informational findings",
      "Description": "Suppress GuardDuty findings with Informational severity",
      "IsTerminal": false,
      "CreatedAt": "2023-05-31T17:56:14.837000+00:00",
    }
  ]
}
```

```
    "UpdatedAt": "2023-05-31T17:59:38.466000+00:00",
    "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
  },
  {
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "RuleStatus": "ENABLED",
    "RuleOrder": 1,
    "RuleName": "sample rule",
    "Description": "A sample rule",
    "IsTerminal": false,
    "CreatedAt": "2023-07-15T23:37:20.223000+00:00",
    "UpdatedAt": "2023-07-15T23:37:20.223000+00:00",
    "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
  },
  {
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "RuleStatus": "ENABLED",
    "RuleOrder": 1,
    "RuleName": "sample rule",
    "Description": "A sample rule",
    "IsTerminal": false,
    "CreatedAt": "2023-07-15T23:45:25.126000+00:00",
    "UpdatedAt": "2023-07-15T23:45:25.126000+00:00",
    "CreatedBy": "arn:aws:iam::123456789012:role/Admin"
  }
]
}
```

詳細については、AWS Security Hub ユーザーガイドの「[オートメーションルールの表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAutomationRules](#)」の「」を参照してください。

list-configuration-policies

次の例は、list-configuration-policies を使用する方法を説明しています。

AWS CLI

設定ポリシーの概要を一覧表示するには

次のlist-configuration-policies例は、組織の設定ポリシーの概要を示しています。

```
aws securityhub list-configuration-policies \  
  --max-items 3
```

出力:

```
{  
  "ConfigurationPolicySummaries": [  
    {  
      "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-  
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Name": "SampleConfigurationPolicy1",  
      "Description": "SampleDescription1",  
      "UpdatedAt": "2023-09-26T21:08:36.214000+00:00",  
      "ServiceEnabled": true  
    },  
    {  
      "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-  
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "Name": "SampleConfigurationPolicy2",  
      "Description": "SampleDescription2",  
      "UpdatedAt": "2023-11-28T19:26:25.207000+00:00",  
      "ServiceEnabled": true  
    },  
    {  
      "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-  
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
      "Name": "SampleConfigurationPolicy3",  
      "Description": "SampleDescription3",  
      "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",  
      "ServiceEnabled": true  
    }  
  ]  
}
```

詳細については、[「Security Hub ユーザーガイド」の「Security Hub 設定ポリシーの表示」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListConfigurationPolicies](#)」の「」を参照してください。

list-configuration-policy-associations

次のコード例は、list-configuration-policy-associations を使用する方法を示しています。

AWS CLI

設定の関連付けを一覧表示するには

次のlist-configuration-policy-associations例は、組織の設定関連付けの概要を示しています。レスポンスには、設定ポリシーとセルフマネージド動作との関連付けが含まれます。

```
aws securityhub list-configuration-policy-associations \  
  --association-type "APPLIED" \  
  --max-items 4
```

出力:

```
{  
  "ConfigurationPolicyAssociationSummaries": [  
    {  
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "TargetId": "r-1ab2",  
      "TargetType": "ROOT",  
      "AssociationType": "APPLIED",  
      "UpdatedAt": "2023-11-28T19:26:49.417000+00:00",  
      "AssociationStatus": "FAILED",  
      "AssociationStatusMessage": "Policy association failed because 2  
organizational units or accounts under this root failed."  
    },  
    {  
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "TargetId": "ou-1ab2-c3de4f5g",  
      "TargetType": "ORGANIZATIONAL_UNIT",  
      "AssociationType": "APPLIED",  
      "UpdatedAt": "2023-09-26T21:14:05.283000+00:00",  
      "AssociationStatus": "FAILED",  
      "AssociationStatusMessage": "One or more children under this target  
failed association."  
    },  
    {  
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
      "TargetId": "ou-6hi7-8j91kl2m",
```



```
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
    "AssociationStatus": "SUCCESS",
    "AssociationStatusMessage": "Association applied successfully on this
target."
  },
  {
    "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
    "TargetId": "111122223333",
    "TargetType": "ACCOUNT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-11-28T22:01:26.409000+00:00",
    "AssociationStatus": "SUCCESS"
  }
}
```

詳細については、[Security Hub ユーザーガイドの「Security Hub 設定ポリシーの表示」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListConfigurationPolicyAssociations](#)」の「」を参照してください。

list-enabled-products-for-import

次のコード例は、list-enabled-products-for-import を使用する方法を示しています。

AWS CLI

有効な製品統合のリストを返すには

次のlist-enabled-products-for-import例では、ARNs現在有効な製品統合のサブスクリプションのリストを返します。

```
aws securityhub list-enabled-products-for-import
```

出力:

```
{
  "ProductSubscriptions": [ "arn:aws:securityhub:us-west-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon", "arn:aws:securityhub:us-
west-1:123456789012:product-subscription/aws/securityhub" ]
}
```

```
}
```

詳細については、AWS Security Hub ユーザーガイドの「[製品統合の管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEnabledProductsForImport](#)」の「」を参照してください。

list-finding-aggregators

次の例は、list-finding-aggregators を使用する方法を説明しています。

AWS CLI

使用可能なウィジェットを一覧表示するには

次のlist-finding-aggregators例では、検出結果集約設定ARNのを返します。

```
aws securityhub list-finding-aggregators
```

出力:

```
{
  "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000"
}
```

詳細については、AWS Security Hub ユーザーガイドの「[現在の検出結果集約設定の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFindingAggregators](#)」の「」を参照してください。

list-invitations

次の例は、list-invitations を使用する方法を説明しています。

AWS CLI

招待のリストを表示するには

次のlist-invitations例では、リクエスト元のアカウントに送信された招待のリストを取得します。

```
aws securityhub list-invitations
```

出力:

```
{
  "Invitations": [
    {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
    }
  ],
}
```

詳細については、Security Hub ユーザーガイドの「[管理者アカウントとメンバーアカウントの管理](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInvitations](#)」の「」を参照してください。

list-members

次のコード例は、list-members を使用する方法を示しています。

AWS CLI

メンバーアカウントのリストを取得するには

次のlist-members例では、リクエストする管理者アカウントのメンバーアカウントのリストを返します。

```
aws securityhub list-members
```

出力:

```
{
  "Members": [
    {
```

```
    "AccountId": "123456789111",
    "AdministratorId": "123456789012",
    "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
    "MasterId": "123456789012",
    "MemberStatus": "ASSOCIATED",
    "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
  },
  {
    "AccountId": "123456789222",
    "AdministratorId": "123456789012",
    "InvitedAt": 2020-06-01T20:15:15.289000+00:00,
    "MasterId": "123456789012",
    "MemberStatus": "ASSOCIATED",
    "UpdatedAt": 2020-06-01T20:15:15.289000+00:00
  }
],
}
```

詳細については、AWS Security Hub ユーザーガイドの [「管理者アカウントとメンバーアカウントの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListMembers](#)」の「」を参照してください。

list-organization-admin-accounts

次のコード例は、list-organization-admin-accounts を使用する方法を示しています。

AWS CLI

指定された Security Hub 管理者アカウントを一覧表示するには

次のlist-organization-admin-accounts例では、組織の Security Hub 管理者アカウントを一覧表示します。

```
aws securityhub list-organization-admin-accounts
```

出力:

```
{
  AdminAccounts": [
    { "AccountId": "777788889999" },
    { "Status": "ENABLED" }
  ]
}
```

```
]
}
```

詳細については、[Security Hub ユーザーガイドの「Security Hub 管理者アカウントの指定」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOrganizationAdminAccounts](#)」の「」を参照してください。

list-security-control-definitions

次の例は、list-security-control-definitions を使用する方法を説明しています。

AWS CLI

例 1: 使用可能なすべてのセキュリティコントロールを一覧表示するには

次のlist-security-control-definitions例では、すべての Security Hub 標準で使用可能なセキュリティコントロールを一覧表示します。この例では、結果を3つのコントロールに制限します。

```
aws securityhub list-security-control-definitions \
  --max-items 3
```

出力:

```
{
  "SecurityControlDefinitions": [
    {
      "SecurityControlId": "ACM.1",
      "Title": "Imported and ACM-issued certificates should be renewed after a
specified time period",
      "Description": "This control checks whether an AWS Certificate Manager
(ACM) certificate is renewed within the specified time period. It checks both
imported certificates and certificates provided by ACM. The control fails if the
certificate isn't renewed within the specified time period. Unless you provide a
custom parameter value for the renewal period, Security Hub uses a default value of
30 days.",
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.1/remediation",
      "SeverityRating": "MEDIUM",
      "CurrentRegionAvailability": "AVAILABLE",
```

```

        "CustomizableProperties": [
            "Parameters"
        ]
    },
    {
        "SecurityControlId": "ACM.2",
        "Title": "RSA certificates managed by ACM should use a key length of at
least 2,048 bits",
        "Description": "This control checks whether RSA certificates managed by
AWS Certificate Manager use a key length of at least 2,048 bits. The control fails
if the key length is smaller than 2,048 bits.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
ACM.2/remediation",
        "SeverityRating": "HIGH",
        "CurrentRegionAvailability": "AVAILABLE",
        "CustomizableProperties": []
    },
    {
        "SecurityControlId": "APIGateway.1",
        "Title": "API Gateway REST and WebSocket API execution logging should be
enabled",
        "Description": "This control checks whether all stages of an Amazon
API Gateway REST or WebSocket API have logging enabled. The control fails if
the 'loggingLevel' isn't 'ERROR' or 'INFO' for all stages of the API. Unless you
provide custom parameter values to indicate that a specific log type should be
enabled, Security Hub produces a passed finding if the logging level is either
'ERROR' or 'INFO'.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
APIGateway.1/remediation",
        "SeverityRating": "MEDIUM",
        "CurrentRegionAvailability": "AVAILABLE",
        "CustomizableProperties": [
            "Parameters"
        ]
    }
],
    "NextToken": "U2FsdGVkX1/UprCPzxVbkDeHikDXbDxfgJZ1w2RG1XWsFPTMTIQPVE0m/
FduIGxS70bRtAbaUt/8/RCQcg2PU0YXI20hH/Grho0Tgv+Tsm0qvQVFhkJepWmqh
+NYawjocVBeos6xzn/8qnbF9IuwGg=="
}

```

詳細については、AWS「[Security Hub ユーザーガイド](#)」の「[標準の詳細の表示](#)」を参照してください。

例 2: 特定の標準で使用可能なセキュリティコントロールを一覧表示するには

次のlist-security-control-definitions例では、CIS AWS Foundations Benchmark v1.4.0 で使用可能なセキュリティコントロールを一覧表示します。この例では、結果を3つのコントロールに制限します。

```
aws securityhub list-security-control-definitions \  
  --standards-arn "arn:aws:securityhub:us-east-1::standards/cis-aws-foundations-  
benchmark/v/1.4.0" \  
  --max-items 3
```

出力:

```
{  
  "SecurityControlDefinitions": [  
    {  
      "SecurityControlId": "CloudTrail.1",  
      "Title": "CloudTrail should be enabled and configured with at least one  
multi-Region trail that includes read and write management events",  
      "Description": "This AWS control checks that there is at least one  
multi-region AWS CloudTrail trail includes read and write management events.",  
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/  
CloudTrail.1/remediation",  
      "SeverityRating": "HIGH",  
      "CurrentRegionAvailability": "AVAILABLE",  
      "CustomizableProperties": []  
    },  
    {  
      "SecurityControlId": "CloudTrail.2",  
      "Title": "CloudTrail should have encryption at-rest enabled",  
      "Description": "This AWS control checks whether AWS CloudTrail is  
configured to use the server side encryption (SSE) AWS Key Management Service (AWS  
KMS) customer master key (CMK) encryption. The check will pass if the KmsKeyId is  
defined.",  
      "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/  
CloudTrail.2/remediation",  
      "SeverityRating": "MEDIUM",  
      "CurrentRegionAvailability": "AVAILABLE",  
      "CustomizableProperties": []  
    },  
    {  
      "SecurityControlId": "CloudTrail.4",  
      "Title": "CloudTrail log file validation should be enabled",
```

```

        "Description": "This AWS control checks whether CloudTrail log file
validation is enabled.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/
CloudTrail.4/remediation",
        "SeverityRating": "MEDIUM",
        "CurrentRegionAvailability": "AVAILABLE",
        "CustomizableProperties": []
    }
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
}

```

詳細については、AWS 「Security Hub ユーザーガイド」の「[標準の詳細の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSecurityControlDefinitions](#)」の「」を参照してください。

list-standards-control-associations

次の例は、list-standards-control-associations を使用する方法を説明しています。

AWS CLI

有効な各標準でコントロールの有効化ステータスを取得するには

次のlist-standards-control-associations例は、有効な各標準で CloudTrail.1 の有効化ステータスを示しています。

```
aws securityhub list-standards-control-associations \
--security-control-id CloudTrail.1
```

出力:

```

{
  "StandardsControlAssociationSummaries": [
    {
      "StandardsArn": "arn:aws:securityhub:us-east-2::standards/nist-800-53/
v/5.0.0",
      "SecurityControlId": "CloudTrail.1",
      "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
      "AssociationStatus": "ENABLED",
    }
  ]
}

```



```

    "RelatedRequirements": [
      "NIST.800-53.r5 AC-2(4)",
      "NIST.800-53.r5 AC-4(26)",
      "NIST.800-53.r5 AC-6(9)",
      "NIST.800-53.r5 AU-10",
      "NIST.800-53.r5 AU-12",
      "NIST.800-53.r5 AU-2",
      "NIST.800-53.r5 AU-3",
      "NIST.800-53.r5 AU-6(3)",
      "NIST.800-53.r5 AU-6(4)",
      "NIST.800-53.r5 AU-14(1)",
      "NIST.800-53.r5 CA-7",
      "NIST.800-53.r5 SC-7(9)",
      "NIST.800-53.r5 SI-3(8)",
      "NIST.800-53.r5 SI-4(20)",
      "NIST.800-53.r5 SI-7(8)",
      "NIST.800-53.r5 SA-8(22)"
    ],
    "UpdatedAt": "2023-05-15T17:52:21.304000+00:00",
    "StandardsControlTitle": "CloudTrail should be enabled and configured
with at least one multi-Region trail that includes read and write management
events",
    "StandardsControlDescription": "This AWS control checks that there is
at least one multi-region AWS CloudTrail trail includes read and write management
events."
  },
  {
    "StandardsArn": "arn:aws:securityhub:::ruleset/cis-aws-foundations-
benchmark/v/1.2.0",
    "SecurityControlId": "CloudTrail.1",
    "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
    "AssociationStatus": "ENABLED",
    "RelatedRequirements": [
      "CIS AWS Foundations 2.1"
    ],
    "UpdatedAt": "2020-02-10T21:22:53.998000+00:00",
    "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
    "StandardsControlDescription": "AWS CloudTrail is a web service that
records AWS API calls for your account and delivers log files to you. The recorded
information includes the identity of the API caller, the time of the API call,
the source IP address of the API caller, the request parameters, and the response
elements returned by the AWS service."
  },

```

```
{
  "StandardsArn": "arn:aws:securityhub:us-east-2::standards/aws-
foundational-security-best-practices/v/1.0.0",
  "SecurityControlId": "CloudTrail.1",
  "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
  "AssociationStatus": "DISABLED",
  "RelatedRequirements": [],
  "UpdatedAt": "2023-05-15T19:31:52.671000+00:00",
  "UpdatedReason": "Alternative compensating controls are in place",
  "StandardsControlTitle": "CloudTrail should be enabled and configured
with at least one multi-Region trail that includes read and write management
events",
  "StandardsControlDescription": "This AWS control checks that there is
at least one multi-region AWS CloudTrail trail includes read and write management
events."
},
{
  "StandardsArn": "arn:aws:securityhub:us-east-2::standards/cis-aws-
foundations-benchmark/v/1.4.0",
  "SecurityControlId": "CloudTrail.1",
  "SecurityControlArn": "arn:aws:securityhub:us-
east-2:123456789012:security-control/CloudTrail.1",
  "AssociationStatus": "ENABLED",
  "RelatedRequirements": [
    "CIS AWS Foundations Benchmark v1.4.0/3.1"
  ],
  "UpdatedAt": "2022-11-10T15:40:36.021000+00:00",
  "StandardsControlTitle": "Ensure CloudTrail is enabled in all regions",
  "StandardsControlDescription": "AWS CloudTrail is a web service that
records AWS API calls for your account and delivers log files to you. The recorded
information includes the identity of the API caller, the time of the API call,
the source IP address of the API caller, the request parameters, and the response
elements returned by the AWS service. CloudTrail provides a history of AWS API
calls for an account, including API calls made via the Management Console, SDKs,
command line tools, and higher-level AWS services (such as CloudFormation)."
}
]
}
```

詳細については、AWS「[Security Hub ユーザーガイド](#)」の「[特定の標準でのコントロールの有効化と無効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListStandardsControlAssociations](#)」の「」を参照してください。

list-tags-for-resource

次のコード例は、list-tags-for-resource を使用する方法を示しています。

AWS CLI

リソースに割り当てられたタグを取得するには

次のlist-tags-for-resource例では、指定されたハブリソースに割り当てられたタグを返します。

```
aws securityhub list-tags-for-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

出力:

```
{  
  "Tags": {  
    "Department" : "Operations",  
    "Area" : "USMidwest"  
  }  
}
```

詳細については、AWS CloudFormation ユーザーガイドの[AWS 「 : SecurityHub:::Hub](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

start-configuration-policy-association

次の例は、start-configuration-policy-association を使用する方法を説明しています。

AWS CLI

例 1: 設定ポリシーを関連付けるには

次のstart-configuration-policy-association例では、指定された設定ポリシーを指定された組織単位に関連付けます。設定は、ターゲットアカウント、組織単位、またはルートに関連付けることができます。

```
aws securityhub start-configuration-policy-association \  
  --configuration-policy-identifier "arn:aws:securityhub:eu-  
central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \  
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

出力:

```
{  
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
  "TargetId": "ou-6hi7-8j91kl2m",  
  "TargetType": "ORGANIZATIONAL_UNIT",  
  "AssociationType": "APPLIED",  
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",  
  "AssociationStatus": "PENDING"  
}
```

詳細については、[Security Hub ユーザーガイドの「Security Hub 設定ポリシーの作成と関連付け」](#)を参照してください。AWS

例 2: セルフマネージド設定を関連付けるには

次のstart-configuration-policy-association例では、セルフマネージド設定を指定されたアカウントと関連付けます。

```
aws securityhub start-configuration-policy-association \  
  --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \  
  --target '{"OrganizationalUnitId": "123456789012"}'
```

出力:

```
{  
  "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",  
  "TargetId": "123456789012",  
  "TargetType": "ACCOUNT",  
  "AssociationType": "APPLIED",  
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",  
  "AssociationStatus": "PENDING"  
}
```

```
}
```

詳細については、[Security Hub ユーザーガイドの「Security Hub 設定ポリシーの作成と関連付け」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[StartConfigurationPolicyAssociation](#)」の「」を参照してください。

start-configuration-policy-disassociation

次のコード例は、start-configuration-policy-disassociation を使用する方法を示しています。

AWS CLI

例 1: 設定ポリシーの関連付けを解除するには

次のstart-configuration-policy-disassociation例では、指定された組織単位から設定ポリシーの関連付けを解除します。設定は、ターゲットアカウント、組織単位、またはルートから関連付け解除できます。

```
aws securityhub start-configuration-policy-disassociation \
  --configuration-policy-identifier "arn:aws:securityhub:eu-
  central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの[「アカウントと設定の関連付け解除 OUs」](#)を参照してください。

例 2: セルフマネージド設定の関連付けを解除するには

次のstart-configuration-policy-disassociation例では、セルフマネージド設定と指定されたアカウントとの関連付けを解除します。

```
aws securityhub start-configuration-policy-disassociation \
  --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
  --target '{"AccountId": "123456789012"}'
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの「[アカウントと設定の関連付け解除 OUs](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [StartConfigurationPolicyDisassociation](#)」の「」を参照してください。

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

リソースにタグを割り当てるには

次のtag-resource例では、部門タグとエリアタグの値を指定されたハブリソースに割り当てます。

```
aws securityhub tag-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
  --tags '{"Department":"Operations", "Area":"USMidwest"}'
```

このコマンドでは何も出力されません。

詳細については、AWS CloudFormation ユーザーガイドの[AWS「: SecurityHub:::Hub」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースからタグ値を削除するには

次のuntag-resource例では、指定されたハブリソースから Department タグを削除します。

```
aws securityhub untag-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
  --tag-key Department
```

```
--tag-keys "Department"
```

このコマンドでは何も出力されません。

詳細については、「ユーザーガイド」の[AWS「: SecurityHub:::Hub」](#)を参照してください。

AWS CloudFormation

- API 詳細については、AWS CLI「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-action-target

次のコード例は、update-action-target を使用する方法を示しています。

AWS CLI

カスタムアクションを更新するには

次のupdate-action-target例では、指定された `arn` によって識別されるカスタムアクションの名前を更新しますARN。

```
aws securityhub update-action-target \  
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation" \  
  --name "Send to remediation"
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの「[カスタムアクションの作成](#)」および [CloudWatch「イベントルールとの関連付け](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateActionTarget](#)」の「」を参照してください。

update-configuration-policy

次のコード例は、update-configuration-policy を使用する方法を示しています。

AWS CLI

設定ポリシーを更新するには

次のupdate-configuration-policy例では、既存の設定ポリシーを更新して、指定された設定を使用します。

```
aws securityhub update-configuration-policy \
  --identifier "arn:aws:securityhub:eu-central-1:508236694226:configuration-
policy/09f37766-57d8-4ede-9d33-5d8b0fecf70e" \
  --name "SampleConfigurationPolicyUpdated" \
  --description "SampleDescriptionUpdated" \
  --configuration-policy '{"SecurityHub": {"ServiceEnabled":
true, "EnabledStandardIdentifiers": ["arn:aws:securityhub:eu-
central-1::standards/aws-foundational-security-best-practices/
v/1.0.0", "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"], "SecurityControlsConfiguration": {"DisabledSecurityControlIdentifiers":
["CloudWatch.1"], "SecurityControlCustomParameters": [{"SecurityControlId":
"ACM.1", "Parameters": {"daysToExpiration": {"ValueType": "CUSTOM", "Value":
{"Integer": 21}}}]}}}' \
  --updated-reason "Disabling CloudWatch.1 and changing parameter value"
```

出力:

```
{
  "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "Name": "SampleConfigurationPolicyUpdated",
  "Description": "SampleDescriptionUpdated",
  "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
  "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
  "ConfigurationPolicy": {
    "SecurityHub": {
      "ServiceEnabled": true,
      "EnabledStandardIdentifiers": [
        "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
        "arn:aws:securityhub::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
      ],
      "SecurityControlsConfiguration": {
        "DisabledSecurityControlIdentifiers": [
          "CloudWatch.1"
        ],
        "SecurityControlCustomParameters": [
          {

```



```
    "SecurityControlId": "ACM.1",
    "Parameters": {
      "daysToExpiration": {
        "ValueType": "CUSTOM",
        "Value": {
          "Integer": 21
        }
      }
    }
  }
]
}
}
```

詳細については、[「Security Hub ユーザーガイド」の「Security Hub 設定ポリシーの更新」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateConfigurationPolicy](#)」の「」を参照してください。

update-finding-aggregator

次のコード例は、update-finding-aggregator を使用する方法を示しています。

AWS CLI

現在の検出結果集約設定を更新するには

次のupdate-finding-aggregator例では、選択したリージョンからリンクするように検出結果集約設定を変更します。これは、集約リージョンである米国東部 (バージニア) から実行されます。リンクされたリージョンとして米国西部 (北カリフォルニア) と米国西部 (オレゴン) を選択します。

```
aws securityhub update-finding-aggregator \  
  --region us-east-1 \  
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-  
aggregator/123e4567-e89b-12d3-a456-426652340000 \  
  --region-linking-mode SPECIFIED_REGIONS \  
  --regions us-west-1,us-west-2
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの[「検出結果集約設定の更新」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFindingAggregator](#)」の「」を参照してください。

update-insight

次の例は、update-insight を使用する方法を説明しています。

AWS CLI

例 1: カスタムインサイトのフィルターを変更するには

次のupdate-insight例では、カスタムインサイトのフィルターを変更します。更新されたインサイトは、AWS ロールに関連する重要度の高い結果を検索します。

```
aws securityhub update-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \  
  --filters '{"ResourceType": [{"Comparison": "EQUALS", "Value": "AwsIamRole"}],  
"SeverityLabel": [{"Comparison": "EQUALS", "Value": "HIGH"}]}' \  
  --name "High severity role findings"
```

例 2: カスタムインサイトのグループ化属性を変更するには

次のupdate-insight例では、カスタムインサイトのグループ化属性を指定されたに変更しますARN。新しいグループ化属性はリソース ID です。

```
aws securityhub update-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \  
  --group-by-attribute "ResourceId" \  
  --name "Critical role findings"
```

出力:

```
{
```

```
"Insights": [
  {
    "InsightArn": "arn:aws:securityhub:us-
west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
    "Name": "Critical role findings",
    "Filters": {
      "SeverityLabel": [
        {
          "Value": "CRITICAL",
          "Comparison": "EQUALS"
        }
      ],
      "ResourceType": [
        {
          "Value": "AwsIamRole",
          "Comparison": "EQUALS"
        }
      ]
    },
    "GroupByAttribute": "ResourceId"
  }
]
```

詳細については、AWS Security Hub ユーザーガイドの「[カスタムインサイトの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateInsight](#)」の「」を参照してください。

update-organization-configuration

次の例は、update-organization-configuration を使用する方法を説明しています。

AWS CLI

Security Hub が組織に構成されている方法を更新するには

次のupdate-organization-configuration例では、Security Hub が中央設定を使用して組織を設定するように指定しています。このコマンドを実行すると、委任された Security Hub 管理者は、組織を設定するための設定ポリシーを作成および管理できます。委任された管理者は、こ

のコマンドを使用して中央設定からローカル設定に切り替えることもできます。ローカル設定が設定タイプである場合、委任された管理者は、新しい組織アカウントで Security Hub とデフォルトのセキュリティ標準を自動的に有効にするかどうかを選択できます。

```
aws securityhub update-organization-configuration \  
  --no-auto-enable \  
  --organization-configuration '{"ConfigurationType": "CENTRAL"}'
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの[AWS 「Organizations でのアカウントの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateOrganizationConfiguration](#)」の「」を参照してください。

update-security-control

次のコード例は、update-security-control を使用する方法を示しています。

AWS CLI

セキュリティコントロールのプロパティを更新するには

次のupdate-security-control例では、Security Hub セキュリティコントロールパラメータのカスタム値を指定します。

```
aws securityhub update-security-control \  
  --security-control-id ACM.1 \  
  --parameters '{"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer":  
15}}}' \  
  --last-update-reason "Internal compliance requirement"
```

このコマンドでは何も出力されません。

詳細については、Security Hub ユーザーガイドの[「カスタムコントロールパラメータ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSecurityControl](#)」の「」を参照してください。

update-security-hub-configuration

次の例は、update-security-hub-configuration を使用する方法を説明しています。

AWS CLI

Security Hub 設定を更新するには

次のupdate-security-hub-configuration例では、Security Hub を設定して、有効な標準に対して新しいコントロールを自動的に有効にします。

```
aws securityhub update-security-hub-configuration \  
  --auto-enable-controls
```

このコマンドでは何も出力されません。

詳細については、Security Hub ユーザーガイドの [「新しいコントロールを自動的に有効にする」](#) を参照してください。 AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSecurityHubConfiguration](#)」の「」を参照してください。

update-standards-control

次のコード例は、update-standards-control を使用する方法を示しています。

AWS CLI

例 1: コントロールを無効にするには

次のupdate-standards-control例では、PCI.AutoScaling.1 コントロールを無効にします。

```
aws securityhub update-standards-control \  
  --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-  
dss/v/3.2.1/PCI.AutoScaling.1" \  
  --control-status "DISABLED" \  
  --disabled-reason "Not applicable for my service"
```

このコマンドでは何も出力されません。

例 2: コントロールを有効にするには

次のupdate-standards-control例では、PCI.AutoScaling.1 コントロールを有効にします。

```
aws securityhub update-standards-control \  
  --standards-control-arn "arn:aws:securityhub:us-west-1:123456789012:control/pci-  
dss/v/3.2.1/PCI.AutoScaling.1" \  
  --control-status "ENABLED"
```

このコマンドでは何も出力されません。

詳細については、AWS Security Hub ユーザーガイドの「[個々のコントロールの無効化と有効化](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateStandardsControl](#)」の「」を参照してください。

を使用した Security Lake の例 AWS CLI

次のコード例は、Security Lake AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-aws-logsource

次の例は、create-aws-logsource を使用する方法を説明しています。

AWS CLI

Amazon Security Lake ソースとしてネイティブにサポートされている Amazon Web Service を追加するには

次のcreate-aws-logsources例では、指定されたアカウントとリージョンにセキュリティレイクソースとしてVPCフローログを追加します。

```
aws securitylake create-aws-log-source \  
  --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],  
  "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

出力:

```
{  
  "failed": [  
    "123456789012"  
  ]  
}
```

詳細については、[「Amazon Security Lake ユーザーガイド」の「ソースとして AWS サービスを追加する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAwsLogsource](#)」の「」を参照してください。

create-custom-logsource

次の例は、create-custom-logsource を使用する方法を説明しています。

AWS CLI

Amazon Security Lake ソースとしてカスタムソースを追加するには

次のcreate-custom-logsource例では、指定されたログプロバイダーアカウントと指定されたリージョンに、Security Lake ソースとしてカスタムソースを追加します。

```
aws securitylake create-custom-log-source \  
  --source-name "VPC_FLOW" \  
  --event-classes '["DNS_ACTIVITY", "NETWORK_ACTIVITY"] \  
  --configuration '{"crawlerConfiguration": {"roleArn": "arn:aws:glue:eu-west-2:123456789012:crawler/E1WG1ZNPRT0D4"},"providerIdentity": {"principal":  
  "029189416600", "externalId": "123456789012"}}' --region "us-east-1"
```

出力:

```
{
  "customLogSource": {
    "attributes": {
      "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRT0D4",
      "databaseArn": "arn:aws:glue:eu-west-2:123456789012:database/
E1WG1ZNPRT0D4",
      "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/E1WG1ZNPRT0D4"
    },
    "provider": {
      "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
      "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-Provider-
testCustom2-eu-west-2"
    },
    "sourceName": "testCustom2"
    "sourceVersion": "2.0"
  }
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[カスタムソースの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomLogsource](#)」の「」を参照してください。

create-data-lake-exception-subscription

次の例は、create-data-lake-exception-subscription を使用する方法を説明しています。

AWS CLI

Security Lake 例外の通知を送信するには

次のcreate-data-lake-exception-subscription例では、Security Lake 例外の通知を指定されたアカウントにSMS配信経由で送信します。例外メッセージは、指定された期間保持されます。

```
aws securitylake create-data-lake-exception-subscription \
  --notification-endpoint "123456789012" \
  --exception-time-to-live 30 \
  --subscription-protocol "sms"
```


このコマンドでは何も出力されません。

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake のトラブルシューティング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDataLakeExceptionSubscription](#)」の「」を参照してください。

create-data-lake-organization-configuration

次のコード例は、create-data-lake-organization-configuration を使用する方法を示しています。

AWS CLI

新しい組織アカウントで Security Lake を設定するには

次のcreate-data-lake-organization-configuration例では、Security Lake を有効にし、指定されたソースイベントとログを新しい組織アカウントで収集します。

```
aws securitylake create-data-lake-organization-configuration \  
  --auto-enable-new-account '[{"region": "us-east-1", "sources":  
  [{"sourceName": "SH_FINDINGS", "sourceVersion": "1.0"}]}'
```

このコマンドでは何も出力されません。

詳細については、「Amazon Security Lake ユーザーガイド」の[AWS 「Organizations で複数のアカウントを管理する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDataLakeOrganizationConfiguration](#)」の「」を参照してください。

create-data-lake

次の例は、create-data-lake を使用する方法を説明しています。

AWS CLI

例 1: 複数のリージョンでデータレイクを設定するには

次のcreate-data-lake例では、複数の AWS リージョンで Amazon Security Lake を有効にし、データレイクを設定します。

```
aws securitylake create-data-lake \
  --configurations '[{"encryptionConfiguration":
{"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
{"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
{"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-2","lifecycleConfiguration":
{"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}]}' \
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

出力:

```
{
  "dataLakes": [
    {
      "createStatus": "COMPLETED",
      "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
      "encryptionConfiguration": {
        "kmsKeyId": "S3_MANAGED_KEY"
      },
      "lifecycleConfiguration": {
        "expiration": {
          "days": 365
        },
        "transitions": [
          {
            "days": 60,
            "storageClass": "ONEZONE_IA"
          }
        ]
      },
      "region": "us-east-1",
      "replicationConfiguration": {
        "regions": [
          "ap-northeast-3"
        ],
        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
      },
      "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnev76s8z7bzby8oi3uiaysbr8v2ml",
    }
  ]
}
```

```
    "updateStatus": {
      "exception": {},
      "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
      "status": "INITIALIZED"
    }
  },
  {
    "createStatus": "COMPLETED",
    "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
    "encryptionConfiguration": {
      "kmsKeyId": "S3_MANAGED_KEY"
    },
    "lifecycleConfiguration": {
      "expiration": {
        "days": 365
      },
      "transitions": [
        {
          "days": 60,
          "storageClass": "ONEZONE_IA"
        }
      ]
    },
    "region": "us-east-2",
    "replicationConfiguration": {
      "regions": [
        "ap-northeast-3"
      ],
      "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
    },
    "s3BucketArn": "arn:aws:s3::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
    "updateStatus": {
      "exception": {},
      "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
      "status": "INITIALIZED"
    }
  }
]
}
```

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake の開始方法」](#)を参照してください。

例 2: データレイクを 1 つのリージョンに設定するには

次のcreate-data-lake例では、単一の AWS リージョンで Amazon Security Lake を有効にし、データレイクを設定します。

```
aws securitylake create-data-lake \
  --configurations '[{"encryptionConfiguration":
{"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}]' \
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

出力:

```
{
  "dataLakes": [
    {
      "createStatus": "COMPLETED",
      "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
      "encryptionConfiguration": {
        "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      "lifecycleConfiguration": {
        "expiration": {
          "days": 500
        },
        "transitions": [
          {
            "days": 30,
            "storageClass": "GLACIER"
          }
        ]
      },
      "region": "us-east-2",
      "replicationConfiguration": {
        "regions": [
          "ap-northeast-3"
        ]
      }
    }
  ]
}
```

```

        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
      },
      "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifz15rwmhm6m62h7zhvtseogr9",
      "updateStatus": {
        "exception": {},
        "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
        "status": "INITIALIZED"
      }
    }
  ]
}

```

詳細については、[「Amazon Security Lake ユーザーガイド」](#)の「[Amazon Security Lake の開始方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateDataLake](#)」の「」を参照してください。

create-subscriber-data-access

次の例は、create-subscriber-data-access を使用する方法を説明しています。

AWS CLI

データアクセスを持つサブスクライバーを作成するには

次のcreate-subscriber例では、AWS ソースの指定されたサブスクライバー ID の現在の AWS リージョンのデータにアクセスできる Security Lake にサブスクライバーを作成します。

```

aws securitylake create-subscriber \
  --access-types "S3" \
  --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW", "sourceVersion":
"2.0"}}]' \
  --subscriber-name "opensearch-s3" \
  --subscriber-identity '{"principal": "029189416600", "externalId":
"123456789012"}'

```

出力:

```
{
```

```
"subscriber": {
  "accessTypes": [
    "S3"
  ],
  "createdAt": "2024-07-17T19:08:26.787000+00:00",
  "roleArn": "arn:aws:iam::773172568199:role/AmazonSecurityLake-896f218b-
cfba-40be-a255-8b49a65d0407",
  "s3BucketArn": "arn:aws:s3::aws-security-data-lake-us-east-1-
um632ufwpvxkyz0bc5hkb64atycnf3",
  "sources": [
    {
      "awsLogSource": {
        "sourceName": "VPC_FLOW",
        "sourceVersion": "2.0"
      }
    }
  ],
  "subscriberArn": "arn:aws:securitylake:us-
east-1:773172568199:subscriber/896f218b-cfba-40be-a255-8b49a65d0407",
  "subscriberId": "896f218b-cfba-40be-a255-8b49a65d0407",
  "subscriberIdentity": {
    "externalId": "123456789012",
    "principal": "029189416600"
  },
  "subscriberName": "opensearch-s3",
  "subscriberStatus": "ACTIVE",
  "updatedAt": "2024-07-17T19:08:27.133000+00:00"
}
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[データアクセスを持つサブスクライバーの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubscriberDataAccess](#)」の「」を参照してください。

create-subscriber-notification

次の例は、create-subscriber-notification を使用する方法を説明しています。

AWS CLI

サブスクライバー通知を作成するには

次のcreate-subscriber-notification例は、サブスクライバー通知を指定して、新しいデータがデータレイクに書き込まれたときに通知を作成する方法を示しています。

```
aws securitylake create-subscriber-notification \  
  --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \  
  --configuration '{"httpsNotificationConfiguration":  
{"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",  
"endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

出力:

```
{  
  "subscriberEndpoint": [  
    "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"  
  ]  
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[サブスクライバー管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubscriberNotification](#)」の「」を参照してください。

create-subscriber-query-access

次の例は、create-subscriber-query-access を使用する方法を説明しています。

AWS CLI

クエリアクセスを持つサブスクライバーを作成するには

次のcreate-subscriber例では、Security Lake に、指定されたサブスクライバー ID の現在のAWS リージョンにクエリアクセスするサブスクライバーを作成します。

```
aws securitylake create-subscriber \  
  --access-types "LAKEFORMATION" \  
  --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW", "sourceVersion":  
"2.0"}}]' \  
  --subscriber-name "opensearch-s3" \  
  --subscriber-identity '{"principal": "029189416600", "externalId":  
"123456789012"}'
```

出力:

```
{
  "subscriber": {
    "accessTypes": [
      "LAKEFORMATION"
    ],
    "createdAt": "2024-07-18T01:05:55.853000+00:00",
    "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/8c31da49-c224-4f1e-bb12-37ab756d6d8a",
    "resourceShareName": "LakeFormation-V2-NAMENAMENA-123456789012",
    "sources": [
      {
        "awsLogSource": {
          "sourceName": "VPC_FLOW",
          "sourceVersion": "2.0"
        }
      }
    ],
    "subscriberArn": "arn:aws:securitylake:us-east-1:123456789012:subscriber/
e762aabb-ce3d-4585-beab-63474597845d",
    "subscriberId": "e762aabb-ce3d-4585-beab-63474597845d",
    "subscriberIdentity": {
      "externalId": "123456789012",
      "principal": "029189416600"
    },
    "subscriberName": "opensearch-s3",
    "subscriberStatus": "ACTIVE",
    "updatedAt": "2024-07-18T01:05:58.393000+00:00"
  }
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の [「クエリアクセスを持つサブスクライバーの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSubscriberQueryAccess](#)」の「」を参照してください。

delete-aws-logsource

次のコード例は、delete-aws-logsource を使用する方法を示しています。

AWS CLI

ネイティブにサポートされる AWS サービスを削除するには。

次のdelete-aws-logsource例では、指定されたアカウントとリージョンの Security Lake ソースとしてVPCフローログを削除します。

```
aws securitylake delete-aws-log-source \  
  --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],  
  "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

出力:

```
{  
  "failed": [  
    "123456789012"  
  ]  
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の [「ソースとしての AWS サービスの削除」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAwsLogsource](#)」の「」を参照してください。

delete-custom-logsource

次のコード例は、delete-custom-logsource を使用する方法を示しています。

AWS CLI

カスタムソースを削除するには。

次のdelete-custom-logsource例では、指定されたリージョンの指定されたログプロバイダーアカウントのカスタムソースを削除します。

```
aws securitylake delete-custom-log-source \  
  --source-name "CustomSourceName"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Security Lake ユーザーガイド」の[「カスタムソースの削除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCustomLogsource](#)」の「」を参照してください。

delete-data-lake-organization-configuration

次のコード例は、delete-data-lake-organization-configuration を使用する方法を示しています。

AWS CLI

メンバーアカウントのソースの自動収集を停止するには

次のdelete-data-lake-organization-configuration例では、組織に参加する新しいメンバーアカウントからの AWS Security Hub の検出結果の自動収集を停止します。このコマンドを実行できるのは、委任された Security Lake 管理者のみです。これにより、新しいメンバーアカウントがデータレイクにデータを自動的に提供できなくなります。

```
aws securitylake delete-data-lake-organization-configuration \  
  --auto-enable-new-account '[{"region": "us-east-1", "sources":  
  [{"sourceName": "SH_FINDINGS"}]}'
```

このコマンドでは何も出力されません。

詳細については、「Amazon Security Lake ユーザーガイド」の[AWS 「Organizations で複数のアカウントを管理する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDataLakeOrganizationConfiguration](#)」の「」を参照してください。

delete-data-lake

次の例は、delete-data-lake を使用する方法を説明しています。

AWS CLI

データレイクを無効にするには

次のdelete-data-lake例では、指定された AWS リージョンのデータレイクを無効にします。指定されたリージョンでは、ソースはデータレイクにデータを提供しなくなります。AWS

Organizations を使用した Security Lake デプロイでは、組織の委任された Security Lake 管理者のみが、組織内のアカウントの Security Lake を無効にすることができます。

```
aws securitylake delete-data-lake \  
  --regions "ap-northeast-1" "eu-central-1"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake の無効化」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDataLake](#)」の「」を参照してください。

delete-subscriber-notification

次のコード例は、delete-subscriber-notification を使用する方法を示しています。

AWS CLI

サブスクライバー通知を削除するには

次のdelete-subscriber-notification例は、特定の Security Lake サブスクライバーのサブスクライバー通知を削除する方法を示しています。

```
aws securitylake delete-subscriber-notification \  
  --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Security Lake ユーザーガイド」の[「サブスクライバー管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSubscriberNotification](#)」の「」を参照してください。

delete-subscriber

次の例は、delete-subscriber を使用する方法を説明しています。

AWS CLI

サブスクライバーを削除するには

次のdelete-subscriber例は、サブスクライバーが Security Lake からデータを消費しなくなった場合にサブスクライバーを削除する方法を示しています。

```
aws securitylake delete-subscriber \  
  --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Security Lake ユーザーガイド」の「[サブスクライバー管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSubscriber](#)」の「」を参照してください。

get-data-lake-exception-subscription

次の例は、get-data-lake-exception-subscription を使用する方法を説明しています。

AWS CLI

例外サブスクリプションの詳細を取得するには

次のget-data-lake-exception-subscription例は、Security Lake 例外サブスクリプションの詳細を示しています。この例では、指定された AWS アカウントのユーザーにSMS配信によるエラーが通知されます。例外メッセージは、指定された期間アカウント内に残ります。例外サブスクリプションは、リクエストの優先プロトコルを通じて Security Lake ユーザーにエラーを通知します。

```
aws securitylake get-data-lake-exception-subscription
```

出力:

```
{  
  "exceptionTimeToLive": 30,  
  "notificationEndpoint": "123456789012",  
  "subscriptionProtocol": "sms"  
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[データレイクステータスのトラブルシューティング](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDataLakeExceptionSubscription](#)」の「」を参照してください。

get-data-lake-organization-configuration

次の例は、get-data-lake-organization-configuration を使用する方法を説明しています。

AWS CLI

新しい組織アカウントの設定の詳細を取得するには

次のget-data-lake-organization-configuration例では、Amazon Security Lake へのオンボーディング後に新しい組織アカウントが送信するソースログの詳細を取得します。

```
aws securitylake get-data-lake-organization-configuration
```

出力:

```
{
  "autoEnableNewAccount": [
    {
      "region": "us-east-1",
      "sources": [
        {
          "sourceName": "VPC_FLOW",
          "sourceVersion": "1.0"
        },
        {
          "sourceName": "ROUTE53",
          "sourceVersion": "1.0"
        },
        {
          "sourceName": "SH_FINDINGS",
          "sourceVersion": "1.0"
        }
      ]
    }
  ]
}
```

```
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の[AWS 「Organizations で複数のアカウントを管理する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDataLakeOrganizationConfiguration](#)」の「」を参照してください。

get-data-lake-sources

次のコード例は、get-data-lake-sources を使用する方法を示しています。

AWS CLI

ログ収集のステータスを取得するには

次のget-data-lake-sources例では、現在の AWS リージョンで指定されたアカウントのログコレクションのスナップショットを取得します。アカウントでは Amazon Security Lake が有効になっています。

```
aws securitylake get-data-lake-sources \  
  --accounts "123456789012"
```

出力:

```
{  
  "dataLakeSources": [  
    {  
      "account": "123456789012",  
      "sourceName": "SH_FINDINGS",  
      "sourceStatuses": [  
        {  
          "resource": "vpc-1234567890abcdef0",  
          "status": "COLLECTING"  
        }  
      ]  
    },  
    {  
      "account": "123456789012",  
      "sourceName": "VPC_FLOW",  
      "sourceStatuses": [  
        {
```

```
        "resource": "vpc-1234567890abcdef0",
        "status": "NOT_COLLECTING"
      }
    ]
  },
  {
    "account": "123456789012",
    "sourceName": "LAMBDA_EXECUTION",
    "sourceStatuses": [
      {
        "resource": "vpc-1234567890abcdef0",
        "status": "COLLECTING"
      }
    ]
  },
  {
    "account": "123456789012",
    "sourceName": "ROUTE53",
    "sourceStatuses": [
      {
        "resource": "vpc-1234567890abcdef0",
        "status": "COLLECTING"
      }
    ]
  },
  {
    "account": "123456789012",
    "sourceName": "CLOUD_TRAIL_MGMT",
    "sourceStatuses": [
      {
        "resource": "vpc-1234567890abcdef0",
        "status": "COLLECTING"
      }
    ]
  }
],
"dataLakeArn": null
}
```

詳細については、「[Amazon Security Lake ユーザーガイド](#)」の [AWS 「サービスからのデータの収集」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDataLakeSources](#)」の「」を参照してください。

get-subscriber

次のコード例は、get-subscriber を使用する方法を示しています。

AWS CLI

サブスクリプション情報を取得するには

次のget-subscriber例では、指定された Security Lake サブスクライバーのサブスクリプション情報を取得します。

```
aws securitylake get-subscriber \  
--subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "subscriber": {  
    "accessTypes": [  
      "LAKEFORMATION"  
    ],  
    "createdAt": "2024-04-19T15:19:44.421803+00:00",  
    "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",  
    "sources": [  
      {  
        "awsLogSource": {  
          "sourceName": "LAMBDA_EXECUTION",  
          "sourceVersion": "1.0"  
        }  
      },  
      {  
        "awsLogSource": {  
          "sourceName": "EKS_AUDIT",  
          "sourceVersion": "2.0"  
        }  
      },  
      {  
        "awsLogSource": {  
          "sourceName": "ROUTE53",  
          "sourceVersion": "1.0"  
        }  
      }  
    ]  
  }  
}
```



```
    },
    {
      "awsLogSource": {
        "sourceName": "SH_FINDINGS",
        "sourceVersion": "1.0"
      }
    },
    {
      "awsLogSource": {
        "sourceName": "VPC_FLOW",
        "sourceVersion": "1.0"
      }
    },
    {
      "customLogSource": {
        "attributes": {
          "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
testCustom2",
          "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
          "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom2"
        },
        "provider": {
          "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfjpbldwbgf4vge98av9/ext/testCustom2/",
          "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-testCustom2-eu-west-2"
        },
        "sourceName": "testCustom2"
      }
    },
    {
      "customLogSource": {
        "attributes": {
          "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
TestCustom",
          "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
          "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom"
        },
        "provider": {
```

```
        "location": "s3://aws-security-data-lake-eu-west-2-8ugsus4ztnsfjpbldwbgf4vge98av9/ext/TestCustom/",
        "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-Provider-TestCustom-eu-west-2"
    },
    "sourceName": "TestCustom"
}
],
"subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"subscriberIdentity": {
    "externalId": "123456789012",
    "principal": "123456789012"
},
"subscriberName": "test",
"subscriberStatus": "ACTIVE",
"updatedAt": "2024-04-19T15:19:55.230588+00:00"
}
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[サブスクライバー管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSubscriber](#)」の「」を参照してください。

list-data-lake-exceptions

次の例は、list-data-lake-exceptions を使用する方法を説明しています。

AWS CLI

データレイクに影響する問題を一覧表示するには

次のlist-data-lake-exceptions例は、指定された AWS リージョンで過去 14 日間にデータレイクに影響を与えている問題を示しています。

```
aws securitylake list-data-lake-exceptions \
  --regions "us-east-1" "eu-west-3"
```

出力:

```
{
  "exceptions": [
    {
      "exception": "The account does not have the required role permissions.
Update your role permissions to use the new data source version.",
      "region": "us-east-1",
      "timestamp": "2024-02-29T12:24:15.641725+00:00"
    },
    {
      "exception": "The account does not have the required role permissions.
Update your role permissions to use the new data source version.",
      "region": "eu-west-3",
      "timestamp": "2024-02-29T12:24:15.641725+00:00"
    }
  ]
}
```

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake のトラブルシューティング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDataLakeExceptions](#)」の「」を参照してください。

list-data-lakes

次の例は、list-data-lakes を使用する方法を説明しています。

AWS CLI

Security Lake 設定オブジェクトを一覧表示するには

次のlist-data-lakes例では、指定された AWS リージョンの Amazon Security Lake 設定オブジェクトを一覧表示します。このコマンドを使用して、指定されたリージョンで Security Lake が有効になっているかどうかを判断できます。

```
aws securitylake list-data-lakes \
  --regions "us-east-1"
```

出力:

```
{
```

```
"dataLakes": [
  {
    "createStatus": "COMPLETED",
    "dataLakeArn": "arn:aws:securitylake:us-east-1:123456789012:data-lake/
default",
    "encryptionConfiguration": {
      "kmsKeyId": "S3_MANAGED_KEY"
    },
    "lifecycleConfiguration": {
      "expiration": {
        "days": 365
      },
      "transitions": [
        {
          "days": 60,
          "storageClass": "ONEZONE_IA"
        }
      ]
    },
    "region": "us-east-1",
    "replicationConfiguration": {
      "regions": [
        "ap-northeast-3"
      ],
      "roleArn": "arn:aws:securitylake:ap-northeast-3:123456789012:data-
lake/default"
    },
    "s3BucketArn": "arn:aws:s3::aws-security-data-lake-us-
east-1-1234567890abcdef0",
    "updateStatus": {
      "exception": {
        "code": "software.amazon.awssdk.services.s3.model.S3Exception",
        "reason": ""
      },
      "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "status": "FAILED"
    }
  }
]
```

詳細については、「[Amazon Security Lake ユーザーガイド](#)」の「[リージョンステータスの確認](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDataLakes](#)」の「」を参照してください。

list-log-sources

次のコード例は、list-log-sources を使用する方法を示しています。

AWS CLI

Amazon Security Lake ログソースを取得するには

次のlist-log-sources例では、指定されたアカウントの Amazon Security Lake ログソースを一覧表示します。

```
aws securitylake list-log-sources \  
  --accounts "123456789012"
```

出力:

```
{  
  "account": "123456789012",  
  "region": "xy-region-1",  
  "sources": [  
    {  
      "awsLogSource": {  
        "sourceName": "VPC_FLOW",  
        "sourceVersion": "2.0"  
      }  
    },  
    {  
      "awsLogSource": {  
        "sourceName": "SH_FINDINGS",  
        "sourceVersion": "2.0"  
      }  
    }  
  ]  
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[ソース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLogSources](#)」の「」を参照してください。

list-subscribers

次のコード例は、list-subscribers を使用する方法を示しています。

AWS CLI

Amazon Security Lake サブスクライバーを取得するには

次のlist-subscribers例では、特定のアカウントのすべての Amazon Security Lake サブスクライバーを一覧表示します。

```
aws securitylake list-subscribers
```

出力:

```
{
  "subscribers": [
    {
      "accessTypes": [
        "S3"
      ],
      "createdAt": "2024-06-04T15:02:28.921000+00:00",
      "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-
E1WG1ZNPRXT0D4",
      "s3BucketArn": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
      "sources": [
        {
          "awsLogSource": {
            "sourceName": "CLOUD_TRAIL_MGMT",
            "sourceVersion": "2.0"
          }
        },
        {
          "awsLogSource": {
            "sourceName": "LAMBDA_EXECUTION",
            "sourceVersion": "1.0"
          }
        }
      ]
    }
  ]
}
```

```
        "customLogSource": {
          "attributes": {
            "crawlerArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRT0D4",
            "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRT0D4",
            "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRT0D4"
          },
          "provider": {
            "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
            "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRT0D4"
          },
          "sourceName": "testCustom2"
        }
      ],
      "subscriberArn": "arn:aws:securitylake:eu-
west-2:123456789012:subscriber/E1WG1ZNPRT0D4",
      "subscriberEndpoint": "arn:aws:sqs:eu-
west-2:123456789012:AmazonSecurityLake-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-Main-
Queue",
      "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "subscriberIdentity": {
        "externalId": "ext123456789012",
        "principal": "123456789012"
      },
      "subscriberName": "Test",
      "subscriberStatus": "ACTIVE",
      "updatedAt": "2024-06-04T15:02:35.617000+00:00"
    }
  ]
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[サブスクライバー管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSubscribers](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

既存のリソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定された Amazon Security Lake サブスクライバーのタグを一覧表示します。この例では、所有者タグキーには関連付けられたタグ値がありません。このオペレーションを使用して、他の既存の Security Lake リソースのタグを一覧表示することもできます。

```
aws securitylake list-tags-for-resource \  
  --resource-arn "arn:aws:securitylake:us-  
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab"
```

出力:

```
{  
  "tags": [  
    {  
      "key": "Environment",  
      "value": "Cloud"  
    },  
    {  
      "key": "CostCenter",  
      "value": "12345"  
    },  
    {  
      "key": "Owner",  
      "value": ""  
    }  
  ]  
}
```

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

register-data-lake-delegated-administrator

次のコード例は、register-data-lake-delegated-administrator を使用する方法を示しています。

AWS CLI

委任された管理者を指定するには

次のregister-data-lake-delegated-administrator例では、指定された AWS アカウントを委任された Amazon Security Lake 管理者として指定します。

```
aws securitylake register-data-lake-delegated-administrator \  
  --account-id 123456789012
```

このコマンドでは何も出力されません。

詳細については、「Amazon Security Lake ユーザーガイド」の[AWS 「Organizations で複数のアカウントを管理する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterDataLakeDelegatedAdministrator](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

既存のリソースにタグを追加するには

次のtag-resource例では、既存のサブスクライバーリソースにタグを追加します。新しいリソースを作成し、1 つ以上のタグを追加するには、このオペレーションを使用しないでください。代わりに、作成するリソースのタイプに適した作成オペレーションを使用します。

```
aws securitylake tag-resource \  
  --resource-arn "arn:aws:securitylake:us-  
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags key=Environment,value=Cloud
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、`untag-resource` を使用する方法を示しています。

AWS CLI

既存のリソースからタグを削除するには

次の`untag-resource`例では、既存のサブスクライバーリソースから指定されたタグを削除します。

```
aws securitylake untag-resource \  
  --resource-arn "arn:aws:securitylake:us-  
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags Environment Owner
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-data-lake-exception-subscription

次の例は、`update-data-lake-exception-subscription` を使用する方法を説明しています。

AWS CLI

Security Lake 例外の通知サブスクリプションを更新するには

次の`update-data-lake-exception-subscription`例では、Security Lake の例外をユーザーに通知する通知サブスクリプションを更新します。

```
aws securitylake update-data-lake-exception-subscription \  
  --resource-arn "arn:aws:securitylake:us-east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --exception-name MyExceptionName
```

```
--notification-endpoint "123456789012" \  
--exception-time-to-live 30 \  
--subscription-protocol "email"
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake のトラブルシューティング」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateDataLakeExceptionSubscription](#)」の「」を参照してください。

update-data-lake

次の例は、update-data-lake を使用する方法を説明しています。

AWS CLI

例 1: データレイク設定を更新するには

次のupdate-data-lake例では、Amazon Security Lake データレイクの設定を更新します。このオペレーションを使用して、データの暗号化、ストレージ、ロールアップリージョンの設定を指定できます。

```
aws securitylake update-data-lake \  
  --configurations '[{"encryptionConfiguration":  
  {"kmsKeyId": "S3_MANAGED_KEY", "region": "us-east-1", "lifecycleConfiguration":  
  {"expiration": {"days": 365}, "transitions":  
  [{"days": 60, "storageClass": "ONEZONE_IA"}]}}, {"encryptionConfiguration":  
  {"kmsKeyId": "S3_MANAGED_KEY", "region": "us-east-2", "lifecycleConfiguration":  
  {"expiration": {"days": 365}, "transitions":  
  [{"days": 60, "storageClass": "ONEZONE_IA"}]}}]' \  
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-  
  role/AmazonSecurityLakeMetaStoreManager"
```

出力:

```
{  
  "dataLakes": [  
    {  
      "createStatus": "COMPLETED",
```

```
    "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
    "encryptionConfiguration": {
        "kmsKeyId": "S3_MANAGED_KEY"
    },
    "lifecycleConfiguration": {
        "expiration": {
            "days": 365
        },
        "transitions": [
            {
                "days": 60,
                "storageClass": "ONEZONE_IA"
            }
        ]
    },
    "region": "us-east-1",
    "replicationConfiguration": {
        "regions": [
            "ap-northeast-3"
        ],
        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
    },
    "s3BucketArn": "arn:aws:s3::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
    "updateStatus": {
        "exception": {},
        "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
        "status": "INITIALIZED"
    }
},
{
    "createStatus": "COMPLETED",
    "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
    "encryptionConfiguration": {
        "kmsKeyId": "S3_MANAGED_KEY"
    },
    "lifecycleConfiguration": {
        "expiration": {
            "days": 365
        },
        "transitions": [
```

```

        {
            "days": 60,
            "storageClass": "ONEZONE_IA"
        }
    ],
    "region": "us-east-2",
    "replicationConfiguration": {
        "regions": [
            "ap-northeast-3"
        ],
        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
    },
    "s3BucketArn": "arn:aws:s3::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
    "updateStatus": {
        "exception": {},
        "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
        "status": "INITIALIZED"
    }
}
]
}

```

詳細については、[「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake の開始方法」](#)を参照してください。

例 2: データレイクを 1 つのリージョンに設定するには

次のcreate-data-lake例では、Amazon Security Lake を 1 つの AWS リージョンで有効にし、データレイクを設定します。

```

aws securitylake create-data-lake \
  --configurations '[{"encryptionConfiguration":
  {"kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"}, "region": "us-
east-2", "lifecycleConfiguration": {"expiration": {"days": 500}, "transitions":
[{"days": 30, "storageClass": "GLACIER"}]}]' \
  --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"

```

出力:

```
{
  "dataLakes": [
    {
      "createStatus": "COMPLETED",
      "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/default",
      "encryptionConfiguration": {
        "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      "lifecycleConfiguration": {
        "expiration": {
          "days": 500
        },
        "transitions": [
          {
            "days": 30,
            "storageClass": "GLACIER"
          }
        ]
      },
      "region": "us-east-2",
      "replicationConfiguration": {
        "regions": [
          "ap-northeast-3"
        ],
        "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-lake/default"
      },
      "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-cehuifz15rwmhm6m62h7zhvtseogr9",
      "updateStatus": {
        "exception": {},
        "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
        "status": "INITIALIZED"
      }
    }
  ]
}
```

詳細については、[「Amazon Security Lake ユーザーガイド」](#)の「[Amazon Security Lake の開始方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDataLake](#)」の「」を参照してください。

update-subscriber-notification

次のコード例は、update-subscriber-notification を使用する方法を示しています。

AWS CLI

サブスクライバー通知を更新するには

次のupdate-subscriber-notification例は、サブスクライバーの通知方法を更新する方法を示しています。

```
aws securitylake update-subscriber-notification \  
  --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \  
  --configuration '{"httpsNotificationConfiguration":  
{"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",  
"endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

出力:

```
{  
  "subscriberEndpoint": [  
    "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"  
  ]  
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[サブスクライバー管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSubscriberNotification](#)」の「」を参照してください。

update-subscriber

次の例は、update-subscriber を使用する方法を説明しています。

AWS CLI

Amazon Security Lake サブスクライバーを更新するには。

次のupdate-subscriber例では、特定の Security Lake サブスクライバーのセキュリティレイクデータソースを更新します。

```
aws securitylake update-subscriber \  
--subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "subscriber": {  
    "accessTypes": [  
      "LAKEFORMATION"  
    ],  
    "createdAt": "2024-04-19T15:19:44.421803+00:00",  
    "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",  
    "sources": [  
      {  
        "awsLogSource": {  
          "sourceName": "LAMBDA_EXECUTION",  
          "sourceVersion": "1.0"  
        }  
      },  
      {  
        "awsLogSource": {  
          "sourceName": "EKS_AUDIT",  
          "sourceVersion": "2.0"  
        }  
      },  
      {  
        "awsLogSource": {  
          "sourceName": "ROUTE53",  
          "sourceVersion": "1.0"  
        }  
      },  
      {  
        "awsLogSource": {  
          "sourceName": "SH_FINDINGS",  
          "sourceVersion": "1.0"  
        }  
      }  
    ]  
  }  
}
```



```
        "awsLogSource": {
            "sourceName": "VPC_FLOW",
            "sourceVersion": "1.0"
        }
    },
    {
        "customLogSource": {
            "attributes": {
                "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRT0D4",
                "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRT0D4",
                "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRT0D4"
            },
            "provider": {
                "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
                "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRT0D4"
            },
            "sourceName": "testCustom2"
        }
    }
],
"subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"subscriberIdentity": {
    "externalId": "123456789012",
    "principal": "123456789012"
},
"subscriberName": "test",
"subscriberStatus": "ACTIVE",
"updatedAt": "2024-07-18T20:47:37.098000+00:00"
}
}
```

詳細については、「Amazon Security Lake ユーザーガイド」の「[サブスクライバー管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSubscriber](#)」の「」を参照してください。

AWS Serverless Application Repository を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Serverless Application Repository。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

put-application-policy

次の例は、put-application-policy を使用する方法を説明しています。

AWS CLI

例 1: アプリケーションを公開するには

以下put-application-policyでは、アプリケーションを公開しているため、誰でも AWS サーバーレスアプリケーションリポジトリでアプリケーションを検索してデプロイできます。

```
aws serverlessrepo put-application-policy \  
  --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-  
test-application \  
  --statements Principals='*',Actions=Deploy
```

出力:

```
{  
  "Statements": [  
    {  
      "Actions": [  
        "Deploy"  
      ]  
    }  
  ]  
}
```

```

    ],
    "Principals": [
        ""
    ],
    "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
]
}

```

例 2: アプリケーションをプライベートに共有するには

以下はアプリケーションをプライベートにput-application-policy共有するため、特定の AWS アカウントのみが AWS サーバーレスアプリケーションリポジトリでアプリケーションを検索してデプロイできます。

```

aws serverlessrepo put-application-policy \
  --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
  --statements Principals=111111111111,222222222222,Actions=Deploy

```

出力:

```

{
  "Statements": [
    {
      "Actions": [
        "Deploy"
      ],
      "Principals": [
        "111111111111",
        "222222222222"
      ],
      "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
  ]
}

```

詳細については、AWS 「[サーバーレスアプリケーションリポジトリデベロッパーガイド](#)」の「[コンソールを介したアプリケーションの共有](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスPutApplicationPolicy](#)」の「」を参照してください。

を使用した Service Catalog の例 AWS CLI

次のコード例は、Service Catalog AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

accept-portfolio-share

次の例は、accept-portfolio-share を使用方法を説明しています。

AWS CLI

ポートフォリオ共有を受け入れるには

次のaccept-portfolio-share例では、指定されたポートフォリオを共有するために別のユーザーが行ったオファーを受け入れます。

```
aws servicecatalog accept-portfolio-share \  
  --portfolio-id port-2s6wuabcdefghijk
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[AcceptPortfolioShare](#)」の「」を参照してください。

associate-principal-with-portfolio

次の例は、associate-principal-with-portfolio を使用方法を説明しています。

AWS CLI

プリンシパルをポートフォリオに関連付けるには

次のassociate-principal-with-portfolio例では、ユーザーを指定されたポートフォリオに関連付けます。

```
aws servicecatalog associate-principal-with-portfolio \  
  --portfolio-id port-2s6abcdefwdh4 \  
  --principal-arn arn:aws:iam::123456789012:user/usertest \  
  --principal-type IAM
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociatePrincipalWithPortfolio](#)」の「」を参照してください。

associate-product-with-portfolio

次の例は、associate-product-with-portfolio を使用する方法を説明しています。

AWS CLI

製品をポートフォリオに関連付けるには

次のassociate-product-with-portfolio例では、指定された製品を指定されたポートフォリオに関連付けます。

```
aws servicecatalog associate-product-with-portfolio \  
  --product-id prod-3p5abcdef3oyk \  
  --portfolio-id port-2s6abcdef5wdh4
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateProductWithPortfolio](#)」の「」を参照してください。

associate-tag-option-with-resource

次のコード例は、associate-tag-option-with-resource を使用する方法を示しています。

AWS CLI

をリソース TagOption に関連付けるには

次のassociate-tag-option-with-resource例では、指定された を指定されたリソース TagOption に関連付けます。

```
aws servicecatalog associate-tag-option-with-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id tag-p3abc2pkpz5qc
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateTagOptionWithResource](#)」の「」を参照してください。

copy-product

次のコード例は、copy-product を使用する方法を示しています。

AWS CLI

製品をコピーするには

次のcopy-product例では、JSON ファイルを使用してパラメータを渡すことで、指定された製品のコピーを作成します。

```
aws servicecatalog copy-product --cli-input-json file://copy-product-input.json
```

copy-product-input.json の内容:

```
{  
  "SourceProductArn": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
tcabcd3syn2xy",  
  "TargetProductName": "copy-of-myproduct",  
  "CopyOptions": [  
    "CopyTags"  
  ]  
}
```

出力:

```
{
  "CopyProductToken": "copyproduct-abc5defgjkdji"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CopyProduct](#)」の「」を参照してください。

create-portfolio-share

次の例は、create-portfolio-share を使用する方法を説明しています。

AWS CLI

ポートフォリオをアカウントと共有するには

次のcreate-portfolio-share例では、指定されたポートフォリオを指定されたアカウントと共有します。

```
aws servicecatalog create-portfolio-share \
  --portfolio-id port-2s6abcdef5wdh4 \
  --account-id 794123456789
```

このコマンドは出力を生成しません。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePortfolioShare](#)」の「」を参照してください。

create-portfolio

次のコード例は、create-portfolio を使用する方法を示しています。

AWS CLI

ポートフォリオを作成するには

次のcreate-portfolio例では、ポートフォリオを作成します。

```
aws servicecatalog create-portfolio \
  --provider-name my-provider \
  --display-name my-portfolio
```

出力:

```
{
  "PortfolioDetail": {
    "ProviderName": "my-provider",
    "DisplayName": "my-portfolio",
    "CreatedTime": 1571337221.555,
    "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-2s6xmplq5wdh4",
    "Id": "port-2s6xmplq5wdh4"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePortfolio](#)」の「」を参照してください。

create-product

次の例は、create-product を使用する方法を説明しています。

AWS CLI

製品を作成するには

次のcreate-product例では、JSON ファイルを使用してパラメータを渡す製品を作成します。

```
aws servicecatalog create-product \
  --cli-input-json file://create-product-input.json
```

create-product-input.json の内容:

```
{
  "AcceptLanguage": "en",
  "Name": "test-product",
  "Owner": "test-owner",
  "Description": "test-description",
  "Distributor": "test-distributor",
  "SupportDescription": "test-support",
  "SupportEmail": "test@amazon.com",
  "SupportUrl": "https://aws.amazon.com",
```



```

"ProductType": "CLOUD_FORMATION_TEMPLATE",
"Tags": [
  {
    "Key": "region",
    "Value": "us-east-1"
  }
],
"ProvisioningArtifactParameters": {
  "Name": "test-version-name",
  "Description": "test-version-description",
  "Info": {
    "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
  },
  "Type": "CLOUD_FORMATION_TEMPLATE"
}
}

```

出力:

```

{
  "Tags": [
    {
      "Key": "region",
      "Value": "us-east-1"
    }
  ],
  "ProductViewDetail": {
    "CreatedTime": 1576025036.0,
    "ProductARN": "arn:aws:catalog:us-west-2:1234568542028:product/
prod-3p5abcdef3oyk",
    "Status": "CREATED",
    "ProductViewSummary": {
      "Type": "CLOUD_FORMATION_TEMPLATE",
      "Distributor": "test-distributor",
      "SupportUrl": "https://aws.amazon.com",
      "SupportEmail": "test@amazon.com",
      "Id": "prodview-abcd42wvx45um",
      "SupportDescription": "test-support",
      "ShortDescription": "test-description",
      "Owner": "test-owner",
      "Name": "test-product2",
      "HasDefaultPath": false,

```

```
        "ProductId": "prod-3p5abcdef3oyk"
      }
    },
    "ProvisioningArtifactDetail": {
      "CreatedTime": 1576025036.0,
      "Active": true,
      "Id": "pa-pq3p5lil12a34",
      "Description": "test-version-description",
      "Name": "test-version-name",
      "Type": "CLOUD_FORMATION_TEMPLATE"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateProduct](#)」の「」を参照してください。

create-provisioning-artifact

次のコード例は、create-provisioning-artifact を使用する方法を示しています。

AWS CLI

プロビジョニングアーティファクトを作成するには

次のcreate-provisioning-artifact例では、JSON ファイルを使用してパラメータを渡すことで、プロビジョニングアーティファクトを作成します。

```
aws servicecatalog create-provisioning-artifact \
  --cli-input-json file://create-provisioning-artifact-input.json
```

create-provisioning-artifact-input.json の内容:

```
{
  "ProductId": "prod-nfi2abcdefghi",
  "Parameters": {
    "Name": "test-provisioning-artifact",
    "Description": "test description",
    "Info": {
      "LoadTemplateFromURL": "https://s3-us-west-1.amazonaws.com/
cloudformation-templates-us-west-1/my-cfn-template.template"
    },
    "Type": "CLOUD_FORMATION_TEMPLATE"
  }
}
```

```
}  
}
```

出力:

```
{  
  "Info": {  
    "TemplateUrl": "https://s3-us-west-1.amazonaws.com/cloudformation-templates-us-west-1/my-cfn-template.template"  
  },  
  "Status": "CREATING",  
  "ProvisioningArtifactDetail": {  
    "Id": "pa-bb4abcdefwnaio",  
    "Name": "test-provisioning-artifact",  
    "Description": "test description",  
    "Active": true,  
    "Type": "CLOUD_FORMATION_TEMPLATE",  
    "CreatedTime": 1576022545.0  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateProvisioningArtifact](#)」の「」を参照してください。

create-tag-option

次の例は、create-tag-option を使用する方法を説明しています。

AWS CLI

を作成するには TagOption

次のcreate-tag-option例では、を作成します TagOption。

```
aws servicecatalog create-tag-option  
  --key 1234  
  --value name
```

出力:

```
{
```

```
"TagOptionDetail": {
  "Id": "tag-iabcdn4fzjjms",
  "Value": "name",
  "Active": true,
  "Key": "1234"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTagOption](#)」の「」を参照してください。

delete-portfolio-share

次の例は、delete-portfolio-share を使用する方法を説明しています。

AWS CLI

アカウントとポートフォリオの共有を停止するには

次のdelete-portfolio-share例では、指定されたアカウントとのポートフォリオの共有を停止します。

```
aws servicecatalog delete-portfolio-share \
  --portfolio-id port-2s6abcdq5wdh4 \
  --account-id 123456789012
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePortfolioShare](#)」の「」を参照してください。

delete-portfolio

次の例は、delete-portfolio を使用する方法を説明しています。

AWS CLI

ポートフォリオを削除するには

次のdelete-portfolio例では、指定されたポートフォリオを削除します。

```
aws servicecatalog delete-portfolio \
```

```
--id port-abcd1x4gox4do
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeletePortfolio](#)」の「」を参照してください。

delete-product

次の例は、delete-product を使用する方法を説明しています。

AWS CLI

製品を削除するには

次のdelete-product例では、指定された製品を削除します。

```
aws servicecatalog delete-product \  
  --id prod-abcdcek6yhbxi
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteProduct](#)」の「」を参照してください。

delete-provisioning-artifact

次のコード例は、delete-provisioning-artifact を使用する方法を示しています。

AWS CLI

プロビジョニングアーティファクトを削除するには

次のdelete-provisioning-artifact例では、指定されたプロビジョニングアーティファクトを削除します。

```
aws servicecatalog delete-provisioning-artifact \  
  --product-id prod-abc2uebuplcpw \  
  --provisioning-artifact-id pa-pqabccddii7ouc
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProvisioningArtifact](#)」の「」を参照してください。

delete-tag-option

次の例は、delete-tag-option を使用する方法を説明しています。

AWS CLI

を削除するには TagOption

次のdelete-tag-option例では、指定された を削除します TagOption。

```
aws servicecatalog delete-tag-option \  
  --id tag-iabcdn4fzjjms
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTagOption](#)」の「」を参照してください。

describe-copy-product-status

次の例は、describe-copy-product-status を使用する方法を説明しています。

AWS CLI

コピー製品オペレーションのステータスを記述するには

次のdescribe-copy-product-status例では、指定された非同期コピー製品オペレーションの現在のステータスを表示します。

```
aws servicecatalog describe-copy-product-status \  
  --copy-product-token copyproduct-znn5tf5abcd3w
```

出力:

```
{  
  "CopyProductStatus": "SUCCEEDED",
```

```
"TargetProductId": "prod-os6hog7abcdt2"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCopyProductStatus](#)」の「」を参照してください。

describe-portfolio

次のコード例は、describe-portfolio を使用する方法を示しています。

AWS CLI

ポートフォリオを記述するには

次のdescribe-portfolio例では、指定されたポートフォリオの詳細を表示します。

```
aws servicecatalog describe-portfolio \
  --id port-2s6abcdq5wdh4
```

出力:

```
{
  "TagOptions": [],
  "PortfolioDetail": {
    "ARN": "arn:aws:catalog:us-west-2:687558541234:portfolio/
port-2s6abcdq5wdh4",
    "Id": "port-2s6wuzyzq5wdh4",
    "CreatedTime": 1571337221.555,
    "DisplayName": "my-portfolio",
    "ProviderName": "my-provider"
  },
  "Tags": []
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePortfolio](#)」の「」を参照してください。

describe-product-as-admin

次の例は、describe-product-as-admin を使用する方法を説明しています。

AWS CLI

製品を管理者として記述するには

次のdescribe-product-as-admin例では、管理者権限を使用して、指定された製品の詳細を表示します。

```
aws servicecatalog describe-product-as-admin \  
  --id prod-abcdcek6yhbx1
```

出力:

```
{  
  "TagOptions": [],  
  "ProductViewDetail": {  
    "ProductARN": "arn:aws:catalog:us-west-2:687558542028:product/prod-  
abcdcek6yhbx1",  
    "ProductViewSummary": {  
      "SupportEmail": "test@amazon.com",  
      "Type": "CLOUD_FORMATION_TEMPLATE",  
      "Distributor": "test-distributor",  
      "ShortDescription": "test-description",  
      "Owner": "test-owner",  
      "Id": "prodview-wi3l2j4abc6vc",  
      "SupportDescription": "test-support",  
      "ProductId": "prod-abcdcek6yhbx1",  
      "HasDefaultPath": false,  
      "Name": "test-product3",  
      "SupportUrl": "https://aws.amazon.com"  
    },  
    "CreatedTime": 1577136715.0,  
    "Status": "CREATED"  
  },  
  "ProvisioningArtifactSummaries": [  
    {  
      "CreatedTime": 1577136715.0,  
      "Description": "test-version-description",  
      "ProvisioningArtifactMetadata": {  
        "SourceProvisioningArtifactId": "pa-abcdxkkiv5fcm"  
      },  
      "Name": "test-version-name-3",  
      "Id": "pa-abcdxkkiv5fcm"  
    }  
  ]  
}
```



```
    ],
    "Tags": [
      {
        "Value": "iad",
        "Key": "region"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeProductAsAdmin](#)」の「」を参照してください。

describe-provisioned-product

次のコード例は、describe-provisioned-product を使用する方法を示しています。

AWS CLI

プロビジョニングされた製品を記述するには

次のdescribe-provisioned-product例では、指定されたプロビジョニング済み製品の詳細を表示します。

```
aws servicecatalog describe-provisioned-product \
  --id pp-dpom27bm4abcd
```

出力:

```
{
  "ProvisionedProductDetail": {
    "Status": "ERROR",
    "CreatedTime": 1577222793.358,
    "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/mytestppname3/pp-dpom27bm4abcd",
    "Id": "pp-dpom27bm4abcd",
    "StatusMessage": "AmazonCloudFormationException Parameters: [KeyName] must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code: ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
    "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
    "LastRecordId": "rec-tfuawdjovzxge",
    "Type": "CFN_STACK",
    "Name": "mytestppname3"
  }
}
```

```
  },  
  "CloudWatchDashboards": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeProvisionedProduct](#)」の「」を参照してください。

describe-provisioning-artifact

次の例は、describe-provisioning-artifact を使用する方法を説明しています。

AWS CLI

プロビジョニングアーティファクトを記述するには

次のdescribe-provisioning-artifact例では、指定されたプロビジョニングアーティファクトの詳細を表示します。

```
aws servicecatalog describe-provisioning-artifact \  
  --provisioning-artifact-id pa-pcz347abcdcfm \  
  --product-id prod-abcdefz3syn2rg
```

出力:

```
{  
  "Info": {  
    "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/  
myexampledevelopment-environment.template"  
  },  
  "ProvisioningArtifactDetail": {  
    "Id": "pa-pcz347abcdcfm",  
    "Active": true,  
    "Type": "CLOUD_FORMATION_TEMPLATE",  
    "Description": "updated description",  
    "CreatedTime": 1562097906.0,  
    "Name": "updated name"  
  },  
  "Status": "AVAILABLE"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeProvisioningArtifact](#)」の「」を参照してください。

describe-tag-option

次のコード例は、describe-tag-option を使用する方法を示しています。

AWS CLI

を記述するには TagOption

次のdescribe-tag-option例では、指定された の詳細を表示します TagOption。

```
aws servicecatalog describe-tag-option \  
  --id tag-p3tej2abcd5qc
```

出力:

```
{  
  "TagOptionDetail": {  
    "Active": true,  
    "Id": "tag-p3tej2abcd5qc",  
    "Value": "value-3",  
    "Key": "1234"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTagOption](#)」の「」を参照してください。

disassociate-principal-from-portfolio

次の例は、disassociate-principal-from-portfolio を使用する方法を説明しています。

AWS CLI

ポートフォリオからプリンシパルの関連付けを解除するには

次のdisassociate-principal-from-portfolio例では、指定されたプリンシパルとポートフォリオの関連付けを解除します。

```
aws servicecatalog disassociate-principal-from-portfolio \  
  --portfolio-id port-2s6abcdq5wdh4 \  
  --principal-id prnc-3t4abcdq5wdh4
```

```
--principal-arn arn:aws:iam::123456789012:group/myendusers
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociatePrincipalFromPortfolio](#)」の「」を参照してください。

disassociate-product-from-portfolio

次のコード例は、disassociate-product-from-portfolio を使用する方法を示しています。

AWS CLI

ポートフォリオから製品の関連付けを解除するには

次のdisassociate-product-from-portfolio例では、指定された製品をポートフォリオから関連付け解除します。

```
aws servicecatalog disassociate-product-from-portfolio \  
  --product-id prod-3p5abcdmu3oyk \  
  --portfolio-id port-2s6abcdq5wdh4
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateProductFromPortfolio](#)」の「」を参照してください。

disassociate-tag-option-from-resource

次の例は、disassociate-tag-option-from-resource を使用する方法を説明しています。

AWS CLI

リソース TagOption から の関連付けを解除するには

次のdisassociate-tag-option-from-resource例では、指定された をリソースTagOptionから関連付け解除します。

```
aws servicecatalog disassociate-tag-option-from-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id tag-p3abc2pkpz5qc
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateTagOptionFromResource](#)」の「」を参照してください。

list-accepted-portfolio-shares

次のコード例は、list-accepted-portfolio-shares を使用する方法を示しています。

AWS CLI

受け入れられたポートフォリオ共有を一覧表示するには

次のlist-accepted-portfolio-shares例では、デフォルトの Service Catalog ポートフォリオのみを含め、このアカウントで共有が受け入れられたすべてのポートフォリオを一覧表示します。

```
aws servicecatalog list-accepted-portfolio-shares \  
  --portfolio-share-type "AWS_SERVICECATALOG"
```

出力:

```
{  
  "PortfolioDetails": [  
    {  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-  
d2abcd5dpkuma",  
      "Description": "AWS Service Catalog Reference blueprints for often-used  
AWS services such as EC2, S3, RDS, VPC and EMR.",  
      "CreatedTime": 1574456190.687,  
      "ProviderName": "AWS Service Catalog",  
      "DisplayName": "Reference Architectures",  
      "Id": "port-d2abcd5dpkuma"  
    },  
    {  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/port-  
abcdefaua7zpu",  
      "Description": "AWS well-architected blueprints for high reliability  
applications.",  
      "CreatedTime": 1574461496.092,  
      "ProviderName": "AWS Service Catalog",  
      "DisplayName": "High Reliability Architectures",
```

```
        "Id": "port-abcdefaua7zpu"
      }
    ]
  }
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAcceptedPortfolioShares](#)」の「」を参照してください。

list-portfolio-access

次の例は、list-portfolio-access を使用する方法を説明しています。

AWS CLI

ポートフォリオにアクセスできるアカウントを一覧表示するには

次のlist-portfolio-access例では、指定されたポートフォリオにアクセスできる AWS アカウントを一覧表示します。

```
aws servicecatalog list-portfolio-access \
  --portfolio-id port-2s6abcdq5wdh4
```

出力:

```
{
  "AccountIds": [
    "123456789012"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPortfolioAccess](#)」の「」を参照してください。

list-portfolios-for-product

次のコード例は、list-portfolios-for-product を使用する方法を示しています。

AWS CLI

製品に関連付けられたポートフォリオを一覧表示するには

次のlist-portfolios-for-product例は、指定された製品に関連付けられたポートフォリオを一覧表示します。

```
aws servicecatalog list-portfolios-for-product \  
  --product-id prod-abcdefz3syn2rg
```

出力:

```
{  
  "PortfolioDetails": [  
    {  
      "CreatedTime": 1571337221.555,  
      "Id": "port-2s6abcdq5wdh4",  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/  
port-2s6abcdq5wdh4",  
      "DisplayName": "my-portfolio",  
      "ProviderName": "my-provider"  
    },  
    {  
      "CreatedTime": 1559665256.348,  
      "Id": "port-5abcd3e5st4ei",  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/  
port-5abcd3e5st4ei",  
      "DisplayName": "test",  
      "ProviderName": "provider-name"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPortfoliosForProduct](#)」の「」を参照してください。

list-portfolios

次のコード例は、list-portfolios を使用する方法を示しています。

AWS CLI

ポートフォリオを一覧表示するには

次のlist-portfolios例では、現在のリージョンの Service Catalog ポートフォリオを一覧表示します。

```
aws servicecatalog list-portfolios
```

出力:

```
{
  "PortfolioDetails": [
    {
      "CreatedTime": 1559665256.348,
      "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-5pzcxmlst4ei",
      "DisplayName": "my-portfolio",
      "Id": "port-5pzcxmlst4ei",
      "ProviderName": "my-user"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPortfolios](#)」の「」を参照してください。

list-principals-for-portfolio

次の例は、list-principals-for-portfolio を使用する方法を説明しています。

AWS CLI

ポートフォリオのすべてのプリンシパルを一覧表示するには

次のlist-principals-for-portfolio例では、指定されたポートフォリオのすべてのプリンシパルを一覧表示します。

```
aws servicecatalog list-principals-for-portfolio \
  --portfolio-id port-2s6abcdq5wdh4
```

出力:

```
{
  "Principals": [
    {
      "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
      "PrincipalType": "IAM"
    }
  ]
}
```



```
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPrincipalsForPortfolio](#)」の「」を参照してください。

list-provisioning-artifacts

次のコード例は、list-provisioning-artifacts を使用する方法を示しています。

AWS CLI

製品のすべてのプロビジョニングアーティファクトを一覧表示するには

次のlist-provisioning-artifacts例では、指定された製品のすべてのプロビジョニングアーティファクトを一覧表示します。

```
aws servicecatalog list-provisioning-artifacts \  
  --product-id prod-nfi2abcdefgcpw
```

出力:

```
{  
  "ProvisioningArtifactDetails": [  
    {  
      "Id": "pa-abcdef54ipm6z",  
      "Description": "test-version-description",  
      "Type": "CLOUD_FORMATION_TEMPLATE",  
      "CreatedTime": 1576021147.0,  
      "Active": true,  
      "Name": "test-version-name"  
    },  
    {  
      "Id": "pa-bb4zyxwwnaio",  
      "Description": "test description",  
      "Type": "CLOUD_FORMATION_TEMPLATE",  
      "CreatedTime": 1576022545.0,  
      "Active": true,  
      "Name": "test-provisioning-artifact-2"  
    }  
  ]  
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProvisioningArtifacts](#)」の「」を参照してください。

list-resources-for-tag-option

次の例は、list-resources-for-tag-option を使用する方法を説明しています。

AWS CLI

に関連付けられたリソースを一覧表示するには TagOption

次のlist-resources-for-tag-option例では、指定されたに関連付けられているリソースを一覧表示しますTagOption。

```
aws servicecatalog list-resources-for-tag-option \  
  --tag-option-id tag-p3tej2abcd5qc
```

出力:

```
{  
  "ResourceDetails": [  
    {  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
abcdfz3syn2rg",  
      "Name": "my product",  
      "Description": "description",  
      "CreatedTime": 1562097906.0,  
      "Id": "prod-abcdfz3syn2rg"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourcesForTagOption](#)」の「」を参照してください。

list-tag-options

次の例は、list-tag-options を使用する方法を説明しています。

AWS CLI

次のlist-tag-options例では、 のすべての値を一覧表示しますTagOptions。

```
aws servicecatalog list-tag-options
```

出力:

```
{
  "TagOptionDetails": [
    {
      "Value": "newvalue",
      "Active": true,
      "Id": "tag-iabcdn4fzjjms",
      "Key": "1234"
    },
    {
      "Value": "value1",
      "Active": true,
      "Id": "tag-e3abcdvmwvrzy",
      "Key": "key"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagOptions](#)」の「」を参照してください。

provision-product

次のコード例は、provision-product を使用する方法を示しています。

AWS CLI

製品をプロビジョニングするには

次のprovision-product例では、指定されたプロビジョニングアーティファクトを使用して、指定された製品をプロビジョニングします。

```
aws servicecatalog provision-product \
  --product-id prod-abcdfz3syn2rg \
```

```
--provisioning-artifact-id pa-abc347pcscfm \  
--provisioned-product-name "mytestppname3"
```

出力:

```
{  
  "RecordDetail": {  
    "RecordId": "rec-tfuawdabcdege",  
    "CreatedTime": 1577222793.362,  
    "ProvisionedProductId": "pp-abcd27bm4mldq",  
    "PathId": "lpv2-abcdg3jp6t5k6",  
    "RecordErrors": [],  
    "ProductId": "prod-abcdefz3syn2rg",  
    "UpdatedTime": 1577222793.362,  
    "RecordType": "PROVISION_PRODUCT",  
    "ProvisionedProductName": "mytestppname3",  
    "ProvisioningArtifactId": "pa-pcz347abcdcfm",  
    "RecordTags": [],  
    "Status": "CREATED",  
    "ProvisionedProductType": "CFN_STACK"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ProvisionProduct](#)」の「」を参照してください。

reject-portfolio-share

次の例は、reject-portfolio-share を使用する方法を説明しています。

AWS CLI

ポートフォリオ共有を拒否するには

次のreject-portfolio-share例では、特定のポートフォリオのポートフォリオ共有を拒否します。

```
aws servicecatalog reject-portfolio-share \  
  --portfolio-id port-2s6wuabcdefghijk
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[RejectPortfolioShare](#)」の「」を参照してください。

scan-provisioned-products

次の例は、scan-provisioned-products を使用する方法を説明しています。

AWS CLI

利用可能なプロビジョニング済み製品をすべて一覧表示するには

次のscan-provisioned-products例では、利用可能なプロビジョニング済み製品を一覧表示します。

```
aws servicecatalog scan-provisioned-products
```

出力:

```
{
  "ProvisionedProducts": [
    {
      "Status": "ERROR",
      "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/mytestppname3/pp-abcd27bm4mldq",
      "StatusMessage": "AmazonCloudFormationException Parameters: [KeyName] must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code: ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
      "Id": "pp-abcd27bm4mldq",
      "Type": "CFN_STACK",
      "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
      "CreatedTime": 1577222793.358,
      "Name": "mytestppname3",
      "LastRecordId": "rec-tfuawdabcdxge"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ScanProvisionedProducts](#)」の「」を参照してください。

search-products-as-admin

次の例は、search-products-as-admin を使用する方法を説明しています。

AWS CLI

管理者権限を持つ製品を検索するには

次のsearch-products-as-admin例では、ポートフォリオ ID をフィルターとして使用して、管理者権限を持つ製品を検索します。

```
aws servicecatalog search-products-as-admin \  
  --portfolio-id port-5abcd3e5st4ei
```

出力:

```
{  
  "ProductViewDetails": [  
    {  
      "ProductViewSummary": {  
        "Name": "my product",  
        "Owner": "owner name",  
        "Type": "CLOUD_FORMATION_TEMPLATE",  
        "ProductId": "prod-abcdefz3syn2rg",  
        "HasDefaultPath": false,  
        "Id": "prodview-abcdmyuzv2dlu",  
        "ShortDescription": "description"  
      },  
      "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
abcdefz3syn2rg",  
      "CreatedTime": 1562097906.0,  
      "Status": "CREATED"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SearchProductsAsAdmin](#)」の「」を参照してください。

search-provisioned-products

次の例は、search-provisioned-products を使用する方法を説明しています。

AWS CLI

プロビジョニングされた製品を検索するには

次のsearch-provisioned-products例では、JSON ファイルを使用してパラメータを渡すことで、指定された製品 ID に一致するプロビジョニング済み製品を検索します。

```
aws servicecatalog search-provisioned-products \  
--cli-input-json file://search-provisioned-products-input.json
```

search-provisioned-products-input.json の内容:

```
{  
  "Filters": {  
    "SearchQuery": [  
      "prod-tcjevz3syn2rg"  
    ]  
  }  
}
```

出力:

```
{  
  "ProvisionedProducts": [  
    {  
      "ProvisioningArtifactId": "pa-pcz347abcdcfm",  
      "Name": "mytestppname3",  
      "CreatedTime": 1577222793.358,  
      "Id": "pp-abcd27bm4mldq",  
      "Status": "ERROR",  
      "UserArn": "arn:aws:iam::123456789012:user/cliuser",  
      "StatusMessage": "AmazonCloudFormationException Parameters: [KeyName]  
must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:  
ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",  
      "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/  
mytestppname3/pp-abcd27bm4mldq",  
      "Tags": [  
        {  
          "Value": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
abcdfz3syn2rg",  
          "Key": "aws:servicecatalog:productArn"  
        }  
      ],  
    }  
  ],  
}
```

```

    {
      "Value": "arn:aws:iam::123456789012:user/cliuser",
      "Key": "aws:servicecatalog:provisioningPrincipalArn"
    },
    {
      "Value": "value-3",
      "Key": "1234"
    },
    {
      "Value": "pa-pcz347abcdcfm",
      "Key": "aws:servicecatalog:provisioningArtifactIdentifier"
    },
    {
      "Value": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-2s6abcdq5wdh4",
      "Key": "aws:servicecatalog:portfolioArn"
    },
    {
      "Value": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
      "Key": "aws:servicecatalog:provisionedProductArn"
    }
  ],
  "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
  "UserArnSession": "arn:aws:iam::123456789012:user/cliuser",
  "Type": "CFN_STACK",
  "LastRecordId": "rec-tfuawdabcdxge",
  "ProductId": "prod-abcdfz3syn2rg"
}
],
"TotalResultsCount": 1
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [SearchProvisionedProducts](#)」の「」を参照してください。

update-portfolio

次の例は、update-portfolio を使用する方法を説明しています。

AWS CLI

ポートフォリオを更新するには

次のupdate-portfolio例では、指定されたポートフォリオの名前を更新します。

```
aws servicecatalog update-portfolio \  
  --id port-5abcd3e5st4ei \  
  --display-name "New portfolio name"
```

出力:

```
{  
  "PortfolioDetail": {  
    "DisplayName": "New portfolio name",  
    "ProviderName": "provider",  
    "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/  
port-5abcd3e5st4ei",  
    "Id": "port-5abcd3e5st4ei",  
    "CreatedTime": 1559665256.348  
  },  
  "Tags": []  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePortfolio](#)」の「」を参照してください。

update-product

次の例は、update-product を使用する方法を説明しています。

AWS CLI

製品を更新するには

次のupdate-product例では、指定された製品の名前と所有者を更新します。

```
aws servicecatalog update-product \  
  --id prod-os6abc7drqlt2 \  
  --name "New product name" \  
  --owner "Updated product owner"
```

出力:

```
{
```

```
"Tags": [
  {
    "Value": "iad",
    "Key": "region"
  }
],
"ProductViewDetail": {
  "ProductViewSummary": {
    "Owner": "Updated product owner",
    "ProductId": "prod-os6abc7drqlt2",
    "Distributor": "test-distributor",
    "SupportUrl": "https://aws.amazon.com",
    "Name": "New product name",
    "ShortDescription": "test-description",
    "HasDefaultPath": false,
    "Id": "prodview-6abcdgrfhvidy",
    "SupportDescription": "test-support",
    "SupportEmail": "test@amazon.com",
    "Type": "CLOUD_FORMATION_TEMPLATE"
  },
  "Status": "CREATED",
  "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
os6abc7drqlt2",
  "CreatedTime": 1577136255.0
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateProduct](#)」の「」を参照してください。

update-provisioning-artifact

次のコード例は、update-provisioning-artifact を使用する方法を示しています。

AWS CLI

プロビジョニングアーティファクトを更新するには

次のupdate-provisioning-artifact例では、JSON ファイルを使用してパラメータを渡すことで、指定されたプロビジョニングアーティファクトの名前と説明を更新します。

```
aws servicecatalog update-provisioning-artifact \
```

```
--cli-input-json file://update-provisioning-artifact-input.json
```

update-provisioning-artifact-input.json の内容:

```
{
  "ProductId": "prod-abcdefz3syn2rg",
  "ProvisioningArtifactId": "pa-pcz347abcdcfm",
  "Name": "updated name",
  "Description": "updated description"
}
```

出力:

```
{
  "Info": {
    "TemplateUrl": "https://awsdocs.s3.amazonaws.com/servicecatalog/
myexampledevelopment-environment.template"
  },
  "Status": "AVAILABLE",
  "ProvisioningArtifactDetail": {
    "Active": true,
    "Description": "updated description",
    "Id": "pa-pcz347abcdcfm",
    "Name": "updated name",
    "Type": "CLOUD_FORMATION_TEMPLATE",
    "CreatedTime": 1562097906.0
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateProvisioningArtifact](#)」の「」を参照してください。

update-tag-option

次のコード例は、update-tag-option を使用する方法を示しています。

AWS CLI

を更新するには TagOption

次のupdate-tag-option例ではTagOption、指定されたJSONファイルを使用して の値を更新します。

```
aws servicecatalog update-tag-option --cli-input-json file://update-tag-option-input.json
```

update-tag-option-input.json の内容:

```
{
  "Id": "tag-iabcdn4fzjjms",
  "Value": "newvalue",
  "Active": true
}
```

出力:

```
{
  "TagOptionDetail": {
    "Value": "newvalue",
    "Key": "1234",
    "Active": true,
    "Id": "tag-iabcdn4fzjjms"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateTagOption](#)」の「」を参照してください。

を使用した Service Quotas の例 AWS CLI

次のコード例は、Service Quotas AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

get-aws-default-service-quota

次の例は、get-aws-default-service-quota を使用する方法を説明しています。

AWS CLI

デフォルトのサービスクォータを記述するには

次のget-aws-default-service-quota例では、指定されたクォータの詳細を表示します。

```
aws service-quotas get-aws-default-service-quota \  
  --service-code ec2 \  
  --quota-code L-1216C47A
```

出力:

```
{  
  "Quota": {  
    "ServiceCode": "ec2",  
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",  
    "QuotaArn": "arn:aws:servicequotas:us-east-2::ec2/L-1216C47A",  
    "QuotaCode": "L-1216C47A",  
    "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)  
instances",  
    "Value": 5.0,  
    "Unit": "None",  
    "Adjustable": true,  
    "GlobalQuota": false,  
    "UsageMetric": {  
      "MetricNamespace": "AWS/Usage",  
      "MetricName": "ResourceCount",  
      "MetricDimensions": {  
        "Class": "Standard/OnDemand",  
        "Resource": "vCPU",  
        "Service": "EC2",  
        "Type": "Resource"  
      },  
      "MetricStatisticRecommendation": "Maximum"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAwsDefaultServiceQuota](#)」の「」を参照してください。

get-requested-service-quota-change

次の例は、get-requested-service-quota-change を使用する方法を説明しています。

AWS CLI

サービスクォータ引き上げリクエストを記述するには

次のget-requested-service-quota-change例では、指定されたクォータ引き上げリクエストについて説明します。

```
aws service-quotas get-requested-service-quota-change \  
--request-id d187537d15254312a9609aa51bbf7624u7W49tP0
```

出力:

```
{  
  "RequestedQuota": {  
    "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",  
    "CaseId": "6780195351",  
    "ServiceCode": "ec2",  
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",  
    "QuotaCode": "L-20F13EBD",  
    "QuotaName": "Running Dedicated c5n Hosts",  
    "DesiredValue": 2.0,  
    "Status": "CASE_OPENED",  
    "Created": 1580446904.067,  
    "LastUpdated": 1580446953.265,  
    "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":  
\\\"arn:aws:iam::123456789012:root\\\"}\",  
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",  
    "GlobalQuota": false,  
    "Unit": "None"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRequestedServiceQuotaChange](#)」の「」を参照してください。

get-service-quota

次の例は、get-service-quota を使用する方法を説明しています。

AWS CLI

サービスクォータを記述するには

次のget-service-quota例では、指定されたクォータの詳細を表示します。

```
aws service-quotas get-service-quota \  
  --service-code ec2 \  
  --quota-code L-1216C47A
```

出力:

```
{  
  "Quota": {  
    "ServiceCode": "ec2",  
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",  
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-1216C47A",  
    "QuotaCode": "L-1216C47A",  
    "QuotaName": "Running On-Demand Standard (A, C, D, H, I, M, R, T, Z)  
instances",  
    "Value": 1920.0,  
    "Unit": "None",  
    "Adjustable": true,  
    "GlobalQuota": false,  
    "UsageMetric": {  
      "MetricNamespace": "AWS/Usage",  
      "MetricName": "ResourceCount",  
      "MetricDimensions": {  
        "Class": "Standard/OnDemand",  
        "Resource": "vCPU",  
        "Service": "EC2",  
        "Type": "Resource"  
      },  
      "MetricStatisticRecommendation": "Maximum"  
    }  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceQuota](#)」の「」を参照してください。

list-aws-default-service-quotas

次の例は、list-aws-default-service-quotas を使用する方法を説明しています。

AWS CLI

サービスのデフォルトクォータを一覧表示するには

次のlist-aws-default-service-quotas例では、指定されたサービスのクォータのデフォルト値を一覧表示します。

```
aws service-quotas list-aws-default-service-quotas \
  --service-code xray
```

出力:

```
{
  "Quotas": [
    {
      "ServiceCode": "xray",
      "ServiceName": "AWS X-Ray",
      "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-C6B6F05D",
      "QuotaCode": "L-C6B6F05D",
      "QuotaName": "Indexed annotations per trace",
      "Value": 50.0,
      "Unit": "None",
      "Adjustable": false,
      "GlobalQuota": false
    },
    {
      "ServiceCode": "xray",
      "ServiceName": "AWS X-Ray",
      "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-D781C0FD",
      "QuotaCode": "L-D781C0FD",
      "QuotaName": "Segment document size",
      "Value": 64.0,
      "Unit": "Kilobytes",
      "Adjustable": false,
      "GlobalQuota": false
    }
  ]
}
```



```
    },
    {
      "ServiceCode": "xray",
      "ServiceName": "AWS X-Ray",
      "QuotaArn": "arn:aws:servicequotas:us-west-2::xray/L-998BFF16",
      "QuotaCode": "L-998BFF16",
      "QuotaName": "Trace and service graph retention in days",
      "Value": 30.0,
      "Unit": "None",
      "Adjustable": false,
      "GlobalQuota": false
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAwsDefaultServiceQuotas](#)」の「」を参照してください。

list-requested-service-quota-change-history-by-quota

次の例は、list-requested-service-quota-change-history-by-quota を使用する方法を説明しています。

AWS CLI

クォータ引き上げリクエストを一覧表示するには

次のlist-requested-service-quota-change-history-by-quota例では、指定されたクォータのクォータ引き上げリクエストを一覧表示します。

```
aws service-quotas list-requested-service-quota-change-history-by-quota \
  --service-code ec2 \
  --quota-code L-20F13EBD
```

出力:

```
{
  "RequestedQuotas": [
    {
      "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",
      "CaseId": "6780195351",
      "ServiceCode": "ec2",
```

```

    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
    "QuotaCode": "L-20F13EBD",
    "QuotaName": "Running Dedicated c5n Hosts",
    "DesiredValue": 2.0,
    "Status": "CASE_OPENED",
    "Created": 1580446904.067,
    "LastUpdated": 1580446953.265,
    "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\\\"arn:aws:iam::123456789012:root\\\"}\",
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
    "GlobalQuota": false,
    "Unit": "None"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListRequestedServiceQuotaChangeHistoryByQuota](#)」の「」を参照してください。

list-requested-service-quota-change-history

次のコード例は、list-requested-service-quota-change-history を使用する方法を示しています。

AWS CLI

クォータ引き上げリクエストを一覧表示するには

次のlist-requested-service-quota-change-history例では、指定されたサービスのクォータ引き上げリクエストを一覧表示します。

```
aws service-quotas list-requested-service-quota-change-history \
  --service-code ec2
```

出力:

```

{
  "RequestedQuotas": [
    {
      "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",

```

```

    "CaseId": "6780195351",
    "ServiceCode": "ec2",
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
    "QuotaCode": "L-20F13EBD",
    "QuotaName": "Running Dedicated c5n Hosts",
    "DesiredValue": 2.0,
    "Status": "CASE_OPENED",
    "Created": 1580446904.067,
    "LastUpdated": 1580446953.265,
    "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":
\\\"arn:aws:iam::123456789012:root\\\"}\",
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/
L-20F13EBD",
    "GlobalQuota": false,
    "Unit": "None"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListRequestedServiceQuotaChangeHistory](#)」の「」を参照してください。

list-service-quotas

次のコード例は、list-service-quotas を使用する方法を示しています。

AWS CLI

サービスのクォータを一覧表示するには

次のlist-service-quotas例では、のクォータの詳細を表示します AWS CloudFormation。

```

aws service-quotas list-service-quotas \
  --service-code cloudformation

```

出力:

```

{
  "Quotas": [
    {
      "ServiceCode": "cloudformation",
      "ServiceName": "AWS CloudFormation",

```

```
    "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-87D14FB7",
    "QuotaCode": "L-87D14FB7",
    "QuotaName": "Output count in CloudFormation template",
    "Value": 60.0,
    "Unit": "None",
    "Adjustable": false,
    "GlobalQuota": false
  },
  {
    "ServiceCode": "cloudformation",
    "ServiceName": "AWS CloudFormation",
    "QuotaArn": "arn:aws:servicequotas:us-
east-2:123456789012:cloudformation/L-0485CB21",
    "QuotaCode": "L-0485CB21",
    "QuotaName": "Stack count",
    "Value": 200.0,
    "Unit": "None",
    "Adjustable": true,
    "GlobalQuota": false
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServiceQuotas](#)」の「」を参照してください。

list-services

次の例は、list-services を使用する方法を説明しています。

AWS CLI

利用可能なサービスを一覧表示するには

次のコマンドは、Service Quotas で利用可能なサービスを一覧表示します。

```
aws service-quotas list-services
```

出力:

```
{
```

```
"Services": [  
  {  
    "ServiceCode": "AWSCloudMap",  
    "ServiceName": "AWS Cloud Map"  
  },  
  {  
    "ServiceCode": "access-analyzer",  
    "ServiceName": "Access Analyzer"  
  },  
  {  
    "ServiceCode": "acm",  
    "ServiceName": "AWS Certificate Manager (ACM)"  
  },  
  ...truncated...  
  {  
    "ServiceCode": "xray",  
    "ServiceName": "AWS X-Ray"  
  }  
]
```

--query パラメータを追加して、関心のある情報に表示をフィルタリングできます。次の例では、サービスコードのみを表示します。

```
aws service-quotas list-services \  
  --query Services[*].ServiceCode
```

出力:

```
[  
  "AWSCloudMap",  
  "access-analyzer",  
  "acm",  
  "acm-pca",  
  "amplify",  
  "apigateway",  
  "application-autoscaling",  
  ...truncated...  
  "xray"  
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServices](#)」の「」を参照してください。

request-service-quota-increase

次の例は、request-service-quota-increase を使用する方法を説明しています。

AWS CLI

サービスクォータの引き上げをリクエストするには

次のrequest-service-quota-increase例では、指定されたサービスクォータの引き上げをリクエストします。

```
aws service-quotas request-service-quota-increase \
  --service-code ec2 \
  --quota-code L-20F13EBD \
  --desired-value 2
```

出力:

```
{
  "RequestedQuota": {
    "Id": "d187537d15254312a9609aa51bbf7624u7W49tP0",
    "ServiceCode": "ec2",
    "ServiceName": "Amazon Elastic Compute Cloud (Amazon EC2)",
    "QuotaCode": "L-20F13EBD",
    "QuotaName": "Running Dedicated c5n Hosts",
    "DesiredValue": 2.0,
    "Status": "PENDING",
    "Created": 1580446904.067,
    "Requester": "{\"accountId\":\"123456789012\",\"callerArn\":\n\n\"arn:aws:iam::123456789012:root\"}",
    "QuotaArn": "arn:aws:servicequotas:us-east-2:123456789012:ec2/L-20F13EBD",
    "GlobalQuota": false,
    "Unit": "None"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RequestServiceQuotaIncrease](#)」の「」を参照してください。

を使用した Amazon SES の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますSES。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

delete-identity

次のコード例は、delete-identity を使用する方法を示しています。

AWS CLI

ID を削除するには

次の例では、delete-identity コマンドを使用して、Amazon で検証された ID のリストから ID を削除しますSES。

```
aws ses delete-identity --identity user@example.com
```

検証された ID の詳細については、「Amazon Simple Email Service デベロッパーガイドSES」の「Amazon での E メールアドレスとドメインの検証」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIdentity](#)」の「」を参照してください。

get-identity-dkim-attributes

次の例は、get-identity-dkim-attributes を使用する方法を説明しています。

AWS CLI

ID のリストの Amazon SES Easy DKIM 属性を取得するには

次の例では、`get-identity-dkim-attributes` コマンドを使用して ID のリストの Amazon SES Easy DKIM 属性を取得します。

```
aws ses get-identity-dkim-attributes --identities "example.com" "user@example.com"
```

出力:

```
{
  "DkimAttributes": {
    "example.com": {
      "DkimTokens": [
        "EXAMPLEejcs5xoyqytjsotsijas7236gr",
        "EXAMPLEejr76cvoc6mysspnioorxsn6ep",
        "EXAMPLEekbmkkhlm2lyz77ppkulerm4k"
      ],
      "DkimEnabled": true,
      "DkimVerificationStatus": "Success"
    },
    "user@example.com": {
      "DkimEnabled": false,
      "DkimVerificationStatus": "NotStarted"
    }
  }
}
```

検証のために、送信したことがない ID を使用してこのコマンドを呼び出した場合、その ID は出力に表示されません。

Easy の詳細については DKIM、「Amazon Simple Email Service デベロッパーガイド SES」の「Amazon DKIMでの Easy」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetIdentityDkimAttributes](#)」の「」を参照してください。

`get-identity-notification-attributes`

次の例は、`get-identity-notification-attributes` を使用する方法を説明しています。

AWS CLI

ID のリストの Amazon SES通知属性を取得するには

次の例では、`get-identity-notification-attributes` コマンドを使用して ID のリストの Amazon SES通知属性を取得します。

```
aws ses get-identity-notification-attributes --
identities "user1@example.com" "user2@example.com"
```

出力:

```
{
  "NotificationAttributes": {
    "user1@example.com": {
      "ForwardingEnabled": false,
      "ComplaintTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
      "BounceTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic",
      "DeliveryTopic": "arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic"
    },
    "user2@example.com": {
      "ForwardingEnabled": true
    }
  }
}
```

このコマンドは、E メールフィードバック転送のステータスと、該当する場合は、バウンス、苦情、配信通知が送信される Amazon SNSトピックの Amazon リソースネーム (ARNs) を返します。

検証のために、送信したことがない ID を使用してこのコマンドを呼び出した場合、その ID は出力に表示されません。

通知の詳細については、Amazon Simple Email Service デベロッパーガイド SES の「Amazon での通知の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetIdentityNotificationAttributes](#)」の「」を参照してください。

get-identity-verification-attributes

次のコード例は、`get-identity-verification-attributes` を使用する方法を示しています。

AWS CLI

ID のリストの Amazon SES 検証ステータスを取得するには

次の例では、`get-identity-verification-attributes` コマンドを使用して、アイデンティティのリストの Amazon SES 検証ステータスを取得します。

```
aws ses get-identity-verification-attributes --
identities "user1@example.com" "user2@example.com"
```

出力:

```
{
  "VerificationAttributes": {
    "user1@example.com": {
      "VerificationStatus": "Success"
    },
    "user2@example.com": {
      "VerificationStatus": "Pending"
    }
  }
}
```

検証のために、送信したことがない ID を使用してこのコマンドを呼び出した場合、その ID は出力に表示されません。

検証された ID の詳細については、「Amazon Simple Email Service デベロッパーガイド SES」の「Amazon での E メールアドレスとドメインの検証」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetIdentityVerificationAttributes](#)」の「」を参照してください。

get-send-quota

次の例は、`get-send-quota` を使用方法を説明しています。

AWS CLI

Amazon SES 送信制限を取得するには

次の例では、`get-send-quota` コマンドを使用して Amazon SES 送信制限を返します。

```
aws ses get-send-quota
```

出力:

```
{
  "Max24HourSend": 200.0,
  "SentLast24Hours": 1.0,
  "MaxSendRate": 1.0
}
```

Max24HourSend は送信クォータです。これは、24 時間以内に送信できる E メール の最大数です。送信クォータには、期間の推移が反映されます。E メールを送信しようとするたびに、Amazon は過去 24 時間に送信された E メール の数 SES を確認します。送信済みのメールの合計数がクォータ未満であれば、送信リクエストは受理され、E メールが送信されます。

SentLast24Hours、過去 24 時間に送信された E メール の数です。

MaxSendRate は、1 秒あたりに送信できる E メール の最大数です。

送信制限は、メッセージ数ではなく、受取人数に基づいていることに注意してください。例えば、受取人数が 10 人である E メールは、送信クォータに対しては 10 通とカウントされます。

詳細については、「Amazon Simple Email Service デベロッパーガイド」の「Amazon SES 送信制限の管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetSendQuota](#)」の「」を参照してください。

get-send-statistics

次のコード例は、get-send-statistics を使用する方法を示しています。

AWS CLI

Amazon SES 送信統計を取得するには

次の例では、get-send-statistics コマンドを使用して Amazon SES 送信統計を返します。

```
aws ses get-send-statistics
```

出力:

```
{
  "SendDataPoints": [
    {
      "Complaints": 0,
      "Timestamp": "2013-06-12T19:32:00Z",
      "DeliveryAttempts": 2,
      "Bounces": 0,
      "Rejects": 0
    },
    {
      "Complaints": 0,
      "Timestamp": "2013-06-12T00:47:00Z",
      "DeliveryAttempts": 1,
      "Bounces": 0,
      "Rejects": 0
    }
  ]
}
```

結果は、送信アクティビティの過去 2 週間を表すデータポイントのリストです。リスト内の各データポイントには、15 分の間隔の統計が含まれます。

この例では、過去 2 週間にユーザーが送信した E メールが 15 分間隔で 2 つしかなかったため、データポイントは 2 つしかありません。

詳細については、「Amazon Simple Email Service デベロッパーガイド」の「Amazon SES 使用状況統計のモニタリング」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSendStatistics](#)」の「」を参照してください。

list-identities

次の例は、list-identities を使用する方法を説明しています。

AWS CLI

特定の AWS アカウントのすべての ID (E メールアドレスとドメイン) を一覧表示するには

次の例では、list-identities コマンドを使用して、Amazon での検証のために送信されたすべての ID を一覧表示します SES。

```
aws ses list-identities
```

出力:

```
{
  "Identities": [
    "user@example.com",
    "example.com"
  ]
}
```

返されるリストには、検証ステータス (検証済み、検証保留中、失敗など) に関係なく、すべての ID が含まれます。

この例では、`identity-type` パラメータを指定しなかったため、E メールアドレスおよびドメインが返されます。

検証の詳細については、「Amazon Simple Email Service デベロッパーガイド SES」の「Amazon での E メールアドレスとドメインの検証」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListIdentities](#)」の「」を参照してください。

send-email

次のコード例は、`send-email` を使用する方法を示しています。

AWS CLI

Amazon を使用してフォーマットされた E メールを送信するには SES

以下の例では、`send-email` コマンドを使用してフォーマットされた E メールを送信しています。

```
aws ses send-email --from sender@example.com --destination file://destination.json
--message file://message.json
```

出力:

```
{
```

```
"MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"  
}
```

送信先とメッセージは、現在のディレクトリの .json ファイルに保存されているJSONデータ構造です。これらのファイルは以下のとおりです。

destination.json:

```
{  
  "ToAddresses": ["recipient1@example.com", "recipient2@example.com"],  
  "CcAddresses": ["recipient3@example.com"],  
  "BccAddresses": []  
}
```

message.json:

```
{  
  "Subject": {  
    "Data": "Test email sent using the AWS CLI",  
    "Charset": "UTF-8"  
  },  
  "Body": {  
    "Text": {  
      "Data": "This is the message body in text format.",  
      "Charset": "UTF-8"  
    },  
    "Html": {  
      "Data": "This message body contains HTML formatting. It can, for example,  
      contain links like this one: <a class=\"ulink\" href=\"http://docs.aws.amazon.com/  
      ses/latest/DeveloperGuide\" target=\"_blank\">Amazon SES Developer Guide</a>.",  
      "Charset": "UTF-8"  
    }  
  }  
}
```

送信者と受信者の E メールアドレスを、使用したい E メールアドレスに置き換えます。送信者の E メールアドレスは Amazon で検証する必要があることに注意してください。SES。Amazon への本番アクセスが許可されるまで SES、受信者が Amazon SES メールボックスシミュレーターでない限り、各受信者の E メールアドレスも確認する必要があります。検証の詳細については、「Amazon Simple Email Service デベロッパーガイド SES」の「Amazon での E メールアドレスとドメインの検証」を参照してください。

出力のメッセージ ID は、send-email の呼び出しが成功したことを示しています。

E メールが届かない場合は、迷惑メールフォルダを確認してください。

フォーマットされた Eメールの送信の詳細については、「Amazon SES API Simple Email Service デベロッパーガイド」の「Amazon を使用したフォーマットされた Eメールの送信」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendEmail](#)」の「」を参照してください。

send-raw-email

次の例は、send-raw-email を使用する方法を説明しています。

AWS CLI

Amazon を使用して raw Eメールを送信するには SES

次の例では、send-raw-email コマンドを使用してTXT、添付ファイルを含む Eメールを送信します。

```
aws ses send-raw-email --raw-message file://message.json
```

出力:

```
{
  "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

raw メッセージは、現在のディレクトリにある という名前message.jsonのファイルに保存されたJSONデータ構造です。以下の要素が含まれます。

```
{
  "Data": "From: sender@example.com\nTo: recipient@example.com\nSubject: Test email sent using the AWS CLI (contains an attachment)\nMIME-Version: 1.0\nContent-type: Multipart/Mixed; boundary=\"NextPart\"\n\n--NextPart\nContent-Type: text/plain\n\nThis is the message body.\n\n--NextPart\nContent-Type: text/plain;\nContent-Disposition: attachment; filename=\"attachment.txt\"\n\nThis is the text in the attachment.\n\n--NextPart--"
```

```
}
```

ご覧のとおり、「Data」は、添付ファイルである attachment.txt など、未加工の E メールコンテンツ全体を MIME 形式で含む 1 つの長い文字列です。

sender@example.com と recipient@example.com は、使用するアドレスに置き換えてください。送信者の E メールアドレスは Amazon で検証する必要があることに注意してください SES。Amazon への本番稼働アクセスが許可されるまで SES、受信者が Amazon SES メールボックスシミュレーターでない限り、受信者の E メールアドレスも確認する必要があります。検証の詳細については、「Amazon Simple Email Service デベロッパーガイド SES」の「Amazon での E メールアドレスとドメインの検証」を参照してください。

出力のメッセージ ID は、への send-raw-email 呼び出しが成功したことを示します。

E メールが届かない場合は、迷惑メールフォルダを確認してください。

未加工の Eメールの送信の詳細については、Amazon Simple Email Service デベロッパーガイドの SES API 「Amazon を使用した未加工の Eメールの送信」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SendRawEmail](#)」の「」を参照してください。

set-identity-dkim-enabled

次の例は、set-identity-dkim-enabled を使用する方法を説明しています。

AWS CLI

Amazon SES 検証済み ID DKIM で Easy を有効または無効にするには

次の例では、set-identity-dkim-enabled コマンドを使用して、検証済みの E メールアドレス DKIM に対して を無効にします。

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

Easy の詳細については DKIM、「Amazon Simple Email Service デベロッパーガイド SES」の「Amazon DKIM での Easy」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetIdentityDkimEnabled](#)」の「」を参照してください。

set-identity-feedback-forwarding-enabled

次の例は、set-identity-feedback-forwarding-enabled を使用する方法を説明しています。

AWS CLI

Amazon SES 検証済み ID のバウンスと苦情の E メールフィードバック転送を有効または無効にするには

次の例では、set-identity-feedback-forwarding-enabled コマンドを使用して、検証済みの E メールアドレスが E メールでバウンスと苦情の通知を受信できるようにします。

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --forwarding-enabled
```

Amazon SNS または E メールフィードバック転送を介してバウンス通知と苦情通知を受け取る必要があるため、バウンス通知と苦情通知の両方に Amazon SNS トピックを選択した場合のみ、E メールフィードバック転送を無効にすることができます。

通知の詳細については、「Amazon Simple Email Service デベロッパーガイド SES」の「Amazon での通知の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [SetIdentityFeedbackForwardingEnabled](#)」の「」を参照してください。

set-identity-notification-topic

次のコード例は、set-identity-notification-topic を使用する方法を示しています。

AWS CLI

Amazon が検証済み ID のバウンス、苦情、配信通知 SES を発行する Amazon SNS トピックを設定するには

次の例では、set-identity-notification-topic コマンドを使用して、検証済みの E メールアドレスがバウンス通知を受信する Amazon SNS トピックを指定します。

```
aws ses set-identity-notification-topic --identity user@example.com --notification-type Bounce --sns-topic arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic
```

通知の詳細については、「Amazon Simple Email Service デベロッパーガイドSES」の「Amazonでの通知の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetIdentityNotificationTopic](#)」の「」を参照してください。

verify-domain-dkim

次の例は、verify-domain-dkim を使用する方法を説明しています。

AWS CLI

Amazon と DKIM 署名するための検証済みドメインの DKIM トークンを生成するには SES

次の例では、verify-domain-dkim コマンドを使用して、Amazon で検証されたドメインの DKIM トークンを生成しますSES。

```
aws ses verify-domain-dkim --domain example.com
```

出力:

```
{
  "DkimTokens": [
    "EXAMPLEEq76owjnks3lnluwg65scbemvw",
    "EXAMPLEi3dnsj67hstzaj673klariwx2",
    "EXAMPLEwfbtcukvimehexktdmtaz6naj"
  ]
}
```

を設定するにはDKIM、返されたDKIMトークンを使用して、Amazon がホストするDKIMパブリックキーを指すCNAMEレコードでドメインDNSの設定を更新する必要がありますSES。詳細については、「Amazon Simple Email Service デベロッパーガイドSES」の「Amazon DKIMで簡単」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[VerifyDomainDkim](#)」の「」を参照してください。

verify-domain-identity

次の例は、verify-domain-identity を使用する方法を説明しています。

AWS CLI

Amazon でドメインを検証するには SES

以下の例では、`verify-domain-identity` コマンドを使用してドメインを認証しています。

```
aws ses verify-domain-identity --domain example.com
```

出力:

```
{
  "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wwmvjuEXAMPLE"
}
```

ドメインの検証を完了するには、返された検証トークンを含むTXTレコードをドメインDNSの設定に追加する必要があります。詳細については、「Amazon Simple Email Service デベロッパーガイドSES」の「Amazon でのドメインの検証」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[VerifyDomainIdentity](#)」の「」を参照してください。

`verify-email-identity`

次のコード例は、`verify-email-identity` を使用する方法を示しています。

AWS CLI

Amazon で E メールアドレスを確認するには SES

以下の例では、`verify-email-identity` コマンドを使用して E メールアドレスを認証しています。

```
aws ses verify-email-identity --email-address user@example.com
```

Amazon を使用して E メールを送信する前に SES、E メールを送信するアドレスまたはドメインを確認して、所有していることを証明する必要があります。本番稼働用アクセスがまだない場合は、Amazon SES メールボックスシミュレーターから提供された E メールアドレスを除き、E メールを送信する E メールアドレスも確認する必要があります。

が呼び出されると `verify-email-identity`、E メールアドレスに検証 E メールが送信されます。ユーザーは、Eメールのリンクをクリックして、検証プロセスを完了する必要があります。

詳細については、「Amazon Simple Email Service デベロッパーガイドSES」の「Amazon での E メールアドレスの検証」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[VerifyEmailIdentity](#)」の「」を参照してください。

を使用したシールドの例 AWS CLI

次のコード例は、Shield AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-drt-log-bucket

次のコード例は、associate-drt-log-bucket を使用する方法を示しています。

AWS CLI

DRTに Amazon S3 バケットへのアクセスを許可するには

次のassociate-drt-log-bucket例では、DRTと指定された S3 バケットの間に関連付けを作成します。これにより、DRTは アカウントに代わってバケットにアクセスできます。

```
aws shield associate-drt-log-bucket \  
  --log-bucket flow-logs-for-website-lb
```

このコマンドでは何も出力されません。

詳細については、AWS Shield Advanced デベロッパーガイドの[DDoS 「レスポンスチームの承認」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateDrtLogBucket](#)」の「」を参照してください。

associate-drt-role

次の例は、associate-drt-role を使用する方法を説明しています。

AWS CLI

にユーザーに代わって潜在的な攻撃DRTを軽減する権限を付与するには

次のassociate-drt-role例では、DRTと指定されたロールの間に関連付けを作成します。DRTは、ロールを使用してアカウントにアクセスして管理できます。

```
aws shield associate-drt-role \  
  --role-arn arn:aws:iam::123456789012:role/service-role/DrtRole
```

このコマンドでは何も出力されません。

詳細については、AWS Shield Advanced デベロッパーガイドの[DDoS 「レスポンスチームの承認」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateDrtRole](#)」の「」を参照してください。

create-protection

次のコード例は、create-protection を使用する方法を示しています。

AWS CLI

1つのAWSリソースに対してAWS Shield Advanced 保護を有効にするには

次のcreate-protection例では、指定されたAWS CloudFront ディストリビューションのShield Advanced 保護を有効にします。

```
aws shield create-protection \  
  --name "Protection for CloudFront distribution" \  
  --resource-arn arn:aws:cloudfront::123456789012:distribution/E198WC25FX0WY8
```

出力:

```
{
  "ProtectionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

詳細については、AWS Shield Advanced デベロッパーガイドの「[保護するリソースを指定する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateProtection](#)」の「」を参照してください。

create-subscription

次のコード例は、create-subscription を使用する方法を示しています。

AWS CLI

アカウントの AWS Shield Advanced 保護を有効にするには

次のcreate-subscription例では、アカウントの Shield Advanced 保護を有効にします。

```
aws shield create-subscription
```

このコマンドでは何も出力されません。

詳細については、[AWS Shield Advanced デベロッパーガイドの「Shield Advanced の開始方法」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateSubscription](#)」の「」を参照してください。

delete-protection

次の例は、delete-protection を使用する方法を説明しています。

AWS CLI

AWS リソースから AWS Shield Advanced 保護を削除するには

次のdelete-protection例では、指定された AWS Shield Advanced 保護を削除します。

```
aws shield delete-protection \  
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Shield Advanced デベロッパーガイド」の AWS 「リソースから Shield Advanced を削除する」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteProtection](#)」の「」を参照してください。

describe-attack

次のコード例は、describe-attack を使用する方法を示しています。

AWS CLI

攻撃の詳細な説明を取得するには

次のdescribe-attack例では、指定されたDDoS攻撃 ID を持つ攻撃の詳細を表示します。list-attacks コマンドを実行するIDsことで攻撃を受けることができます。

```
aws shield describe-attack --attack-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

出力:

```
{
  "Attack": {
    "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "ResourceArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/testElb",
    "SubResources": [
      {
        "Type": "IP",
        "Id": "192.0.2.2",
        "AttackVectors": [
          {
            "VectorType": "SYN_FLOOD",
            "VectorCounters": [
              {
                "Name": "SYN_FLOOD_BPS",
                "Max": 982184.0,
                "Average": 982184.0,
                "Sum": 11786208.0,
                "N": 12,
                "Unit": "BPS"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
    }
  ]
}
],
"Counters": []
},
{
  "Type": "IP",
  "Id": "192.0.2.3",
  "AttackVectors": [
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 9821840.0,
          "N": 10,
          "Unit": "BPS"
        }
      ]
    }
  ]
},
"Counters": []
},
{
  "Type": "IP",
  "Id": "192.0.2.4",
  "AttackVectors": [
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 7857472.0,
          "N": 8,
          "Unit": "BPS"
        }
      ]
    }
  ]
},
],
```



```
    "Counters": [],
  },
  {
    "Type": "IP",
    "Id": "192.0.2.5",
    "AttackVectors": [
      {
        "VectorType": "SYN_FLOOD",
        "VectorCounters": [
          {
            "Name": "SYN_FLOOD_BPS",
            "Max": 982184.0,
            "Average": 982184.0,
            "Sum": 1964368.0,
            "N": 2,
            "Unit": "BPS"
          }
        ]
      }
    ]
  },
  "Counters": [],
},
{
  "Type": "IP",
  "Id": "2001:DB8::bcde:4321:8765:0:0",
  "AttackVectors": [
    {
      "VectorType": "SYN_FLOOD",
      "VectorCounters": [
        {
          "Name": "SYN_FLOOD_BPS",
          "Max": 982184.0,
          "Average": 982184.0,
          "Sum": 1964368.0,
          "N": 2,
          "Unit": "BPS"
        }
      ]
    }
  ],
  "Counters": []
},
{
  "Type": "IP",
```

```
"Id": "192.0.2.6",
"AttackVectors": [
  {
    "VectorType": "SYN_FLOOD",
    "VectorCounters": [
      {
        "Name": "SYN_FLOOD_BPS",
        "Max": 982184.0,
        "Average": 982184.0,
        "Sum": 1964368.0,
        "N": 2,
        "Unit": "BPS"
      }
    ]
  }
],
"Counters": []
},
"StartTime": 1576024927.457,
"EndTime": 1576025647.457,
"AttackCounters": [],
"AttackProperties": [
  {
    "AttackLayer": "NETWORK",
    "AttackPropertyIdentifier": "SOURCE_IP_ADDRESS",
    "TopContributors": [
      {
        "Name": "198.51.100.5",
        "Value": 2024475682
      },
      {
        "Name": "198.51.100.8",
        "Value": 1311380863
      },
      {
        "Name": "203.0.113.4",
        "Value": 900599855
      },
      {
        "Name": "198.51.100.4",
        "Value": 769417366
      }
    ]
  }
]
```

```
        "Name": "203.1.113.13",
        "Value": 757992847
    }
],
"Unit": "BYTES",
"Total": 92773354841
},
{
    "AttackLayer": "NETWORK",
    "AttackPropertyIdentifier": "SOURCE_COUNTRY",
    "TopContributors": [
        {
            "Name": "United States",
            "Value": 80938161764
        },
        {
            "Name": "Brazil",
            "Value": 9929864330
        },
        {
            "Name": "Netherlands",
            "Value": 1635009446
        },
        {
            "Name": "Mexico",
            "Value": 144832971
        },
        {
            "Name": "Japan",
            "Value": 45369000
        }
    ],
    "Unit": "BYTES",
    "Total": 92773354841
},
{
    "AttackLayer": "NETWORK",
    "AttackPropertyIdentifier": "SOURCE_ASN",
    "TopContributors": [
        {
            "Name": "12345",
            "Value": 74953625841
        },
        {
```

```
        "Name": "12346",
        "Value": 4440087595
      },
      {
        "Name": "12347",
        "Value": 1635009446
      },
      {
        "Name": "12348",
        "Value": 1221230000
      },
      {
        "Name": "12349",
        "Value": 1199425294
      }
    ],
    "Unit": "BYTES",
    "Total": 92755479921
  }
],
"Mitigations": []
}
}
```

詳細については、AWS Shield Advanced デベロッパーガイドの[DDoS 「インシデントの確認」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAttack](#)」の「」を参照してください。

describe-drt-access

次の例は、describe-drt-access を使用する方法を説明しています。

AWS CLI

がユーザーに代わって攻撃を軽減するためにDRT持つ承認の説明を取得するには

次のdescribe-drt-access例では、 がDRT持つロールと S3 バケット認証を取得し、ユーザーに代わって潜在的な攻撃に応答できるようにします。

```
aws shield describe-drt-access
```

出力:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/service-role/DrtRole",
  "LogBucketList": [
    "flow-logs-for-website-lb"
  ]
}
```

詳細については、AWS Shield Advanced デベロッパーガイドの [DDoS 「レスポンスチームの承認」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDrtAccess](#)」の「」を参照してください。

describe-emergency-contact-settings

次のコード例は、describe-emergency-contact-settings を使用する方法を示しています。

AWS CLI

に登録されている緊急 E メールアドレスを取得するには DRT

次のdescribe-emergency-contact-settings例では、DRTアカウントのに登録されている E メールアドレスを取得します。これらは、疑わしい攻撃に応答しているときに が問い合わせ DRTるアドレスです。

```
aws shield describe-emergency-contact-settings
```

出力:

```
{
  "EmergencyContactList": [
    {
      "EmailAddress": "ops@example.com"
    },
    {
      "EmailAddress": "ddos-notifications@example.com"
    }
  ]
}
```

詳細については、AWS Shield AWS Advanced デベロッパーガイドの「Shield の仕組み <<https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html>>」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEmergencyContactSettings](#)」の「」を参照してください。

describe-protection

次の例は、describe-protection を使用する方法を説明しています。

AWS CLI

AWS Shield Advanced 保護の詳細を取得するには

次のdescribe-protection例では、指定された ID を持つ Shield Advanced 保護の詳細を表示します。list-protections コマンドを実行するIDsことで、保護を取得できます。

```
aws shield describe-protection \  
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "Protection": {  
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Name": "1.2.3.4",  
    "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:eip-allocation/  
eipalloc-0ac1537af40742a6d"  
  }  
}
```

詳細については、AWS Shield Advanced デベロッパーガイドの「[保護するリソースを指定する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeProtection](#)」の「」を参照してください。

describe-subscription

次の例は、describe-subscription を使用する方法を説明しています。

AWS CLI

アカウントの AWS Shield Advanced 保護の詳細を取得するには

次のdescribe-subscription例では、アカウントに提供される Shield Advanced 保護の詳細を表示します。

```
aws shield describe-subscription
```

出力:

```
{
  "Subscription": {
    "StartTime": 1534368978.0,
    "EndTime": 1597613778.0,
    "TimeCommitmentInSeconds": 63244800,
    "AutoRenew": "ENABLED",
    "Limits": [
      {
        "Type": "GLOBAL_ACCELERATOR",
        "Max": 1000
      },
      {
        "Type": "ROUTE53_HOSTED_ZONE",
        "Max": 1000
      },
      {
        "Type": "CF_DISTRIBUTION",
        "Max": 1000
      },
      {
        "Type": "ELB_LOAD_BALANCER",
        "Max": 1000
      },
      {
        "Type": "EC2_ELASTIC_IP_ALLOCATION",
        "Max": 1000
      }
    ]
  }
}
```

詳細については、[AWS Shield Advanced デベロッパーガイドの「Shield の仕組み」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSubscription](#)」の「」を参照してください。

disassociate-drt-log-bucket

次の例は、disassociate-drt-log-bucket を使用する方法を説明しています。

AWS CLI

お客様に代わって Amazon S3 バケットにアクセスDRTするための の承認を削除するには

次のdisassociate-drt-log-bucket例では、DRTと指定された S3 バケット間の関連付けを削除します。このコマンドが完了すると、DRT はアカウントに代わってバケットにアクセスできなくなります。

```
aws shield disassociate-drt-log-bucket \  
  --log-bucket flow-logs-for-website-lb
```

このコマンドでは何も出力されません。

詳細については、AWS Shield Advanced デベロッパーガイドの[DDoS 「レスポンスチームの承認」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateDrtLogBucket](#)」の「」を参照してください。

disassociate-drt-role

次の例は、disassociate-drt-role を使用する方法を説明しています。

AWS CLI

お客様に代わって潜在的な攻撃を軽減DRTするための の承認を削除するには

次のdisassociate-drt-role例では、DRTと アカウント間の関連付けを削除します。この呼び出し後、DRT はアカウントにアクセスまたは管理できなくなります。

```
aws shield disassociate-drt-role
```


このコマンドでは何も出力されません。

詳細については、AWS Shield Advanced デベロッパーガイドの[DDoS「レスポンスチームの承認」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DisassociateDrtRole](#)」の「」を参照してください。

get-subscription-state

次の例は、get-subscription-state を使用する方法を説明しています。

AWS CLI

アカウントの AWS Shield Advanced サブスクリプションの現在の状態を取得するには

次のget-subscription-state例では、アカウントの Shield Advanced 保護の状態を取得します。

```
aws shield get-subscription-state
```

出力:

```
{
  "SubscriptionState": "ACTIVE"
}
```

詳細については、[AWS Shield Advanced デベロッパーガイドの「Shield の仕組み」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[GetSubscriptionState](#)」の「」を参照してください。

list-attacks

次のコード例は、list-attacks を使用する方法を示しています。

AWS CLI

AWS Shield Advanced から攻撃の概要を取得するには

次のlist-attacks例では、指定された期間中の指定された AWS CloudFront デイストリビューションの攻撃の概要を取得します。レスポンスにはIDs、攻撃に関する詳細情報を得るためにdescribe-attack コマンドに提供できる攻撃が含まれます。

```
aws shield list-attacks \  
  --resource-arns arn:aws:cloudfront::12345678910:distribution/E1PXMP22ZVFAOR \  
  --start-time FromInclusive=1529280000,ToExclusive=1529300000
```

出力:

```
{  
  "AttackSummaries": [  
    {  
      "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/  
E1PXMP22ZVFAOR",  
      "StartTime": 1529280000.0,  
      "EndTime": 1529449200.0,  
      "AttackVectors": [  
        {  
          "VectorType": "SYN_FLOOD"  
        }  
      ]  
    }  
  ]  
}
```

詳細については、AWS Shield Advanced デベロッパーガイドの[DDoS 「インシデントの確認」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAttacks](#)」の「」を参照してください。

list-protections

次のコード例は、list-protections を使用する方法を示しています。

AWS CLI

AWS Shield Advanced から保護の概要を取得するには

次のlist-protections例では、アカウントで有効になっている保護の概要を取得します。

aws shield list-protections

出力:

```
{
  "Protections": [
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Name": "Protection for CloudFront distribution",
      "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E198WC25FX0WY8"
    }
  ]
}
```

詳細については、AWS Shield Advanced デベロッパーガイドの「[保護するリソースを指定する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListProtections](#)」の「」を参照してください。

update-emergency-contact-settings

次の例は、update-emergency-contact-settings を使用する方法を説明しています。

AWS CLI

に登録されている緊急 E メールアドレスを定義するには DRT

次のupdate-emergency-contact-settings例では、疑わしい攻撃に応答しているときに が 問い合わせDRTる必要がある 2 つの E メールアドレスを定義します。

```
aws shield update-emergency-contact-settings \
  --emergency-contact-list EmailAddress=ops@example.com EmailAddress=ddos-
notifications@example.com
```

このコマンドでは何も出力されません。

詳細については、[AWS Shield Advanced デベロッパーガイドの「Shield の仕組み](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateEmergencyContactSettings](#)」の「」を参照してください。

update-subscription

次のコード例は、update-subscription を使用する方法を示しています。

AWS CLI

アカウントの AWS Shield Advanced サブスクリプションを変更するには

次のupdate-subscription例では、アカウントの AWS Shield Advanced サブスクリプションの自動更新を有効にします。

```
aws shield update-subscription \  
  --auto-renew ENABLED
```

このコマンドでは何も出力されません。

詳細については、[AWS Shield Advanced デベロッパーガイドの「Shield の仕組み」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSubscription](#)」の「」を参照してください。

を使用した署名者の例 AWS CLI

次のコード例は、Signer AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

cancel-signing-profile

次の例は、cancel-signing-profile を使用する方法を説明しています。

AWS CLI

署名プロファイルを削除するには

次のcancel-signing-profile例では、AWS Signer から既存の署名プロファイルを削除します。

```
aws signer cancel-signing-profile \  
  --profile-name MyProfile1
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelSigningProfile](#)」の「」を参照してください。

describe-signing-job

次の例は、describe-signing-job を使用する方法を説明しています。

AWS CLI

署名ジョブの詳細を表示するには

次のdescribe-signing-job例では、指定された署名ジョブの詳細を表示します。

```
aws signer describe-signing-job \  
  --job-id 2065c468-73e2-4385-a6c9-0123456789abc
```

出力:

```
{  
  "status": "Succeeded",  
  "completedAt": 1568412037,  
  "platformId": "AmazonFreeRTOS-Default",  
  "signingMaterial": {  
    "certificateArn": "arn:aws:acm:us-  
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
```

```

    },
    "statusReason": "Signing Succeeded",
    "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",
    "source": {
      "s3": {
        "version": "PNyFaUTgsQh5ZdMccoCe6pT1g0pgB_M4",
        "bucketName": "signer-source",
        "key": "MyCode.rb"
      }
    },
    "profileName": "MyProfile2",
    "signedObject": {
      "s3": {
        "bucketName": "signer-destination",
        "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
      }
    },
    "requestedBy": "arn:aws:iam::123456789012:user/maria",
    "createdAt": 1568412036
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSigningJob](#)」の「」を参照してください。

get-signing-platform

次のコード例は、get-signing-platform を使用する方法を示しています。

AWS CLI

署名プラットフォームの詳細を表示するには

次のget-signing-platform例では、指定された署名プラットフォームの詳細を表示します。

```

aws signer get-signing-platform \
  --platform-id AmazonFreeRTOS-TI-CC3220SF

```

出力:

```

{
  "category": "AWS",

```

```
"displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
"target": "SHA1-RSA-TISHA1",
"platformId": "AmazonFreeRTOS-TI-CC3220SF",
"signingConfiguration": {
  "encryptionAlgorithmOptions": {
    "defaultValue": "RSA",
    "allowedValues": [
      "RSA"
    ]
  },
  "hashAlgorithmOptions": {
    "defaultValue": "SHA1",
    "allowedValues": [
      "SHA1"
    ]
  }
},
"maxSizeInMB": 16,
"partner": "AmazonFreeRTOS",
"signingImageFormat": {
  "defaultFormat": "JSONEmbedded",
  "supportedFormats": [
    "JSONEmbedded"
  ]
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSigningPlatform](#)」の「」を参照してください。

get-signing-profile

次の例は、get-signing-profile を使用する方法を説明しています。

AWS CLI

署名プロファイルの詳細を表示するには

次のget-signing-profile例では、指定された署名プロファイルの詳細を表示します。

```
aws signer get-signing-profile \
  --profile-name MyProfile3
```

出力:

```
{
  "platformId": "AmazonFreeRTOS-TI-CC3220SF",
  "profileName": "MyProfile3",
  "status": "Active",
  "signingMaterial": {
    "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSigningProfile](#)」の「」を参照してください。

list-signing-jobs

次の例は、list-signing-jobs を使用する方法を説明しています。

AWS CLI

すべての署名ジョブを一覧表示するには

次のlist-signing-jobs例では、アカウントのすべての署名ジョブの詳細を表示します。

```
aws signer list-signing-jobs
```

この例では、2つのジョブが返され、1つは成功、もう1つは失敗します。

```
{
  "jobs": [
    {
      "status": "Succeeded",
      "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
      },
      "jobId": "2065c468-73e2-4385-a6c9-0123456789abc",
      "source": {
        "s3": {
          "version": "PNyFaUTgsQh5ZdMCcoCe6pT1g0pgB_M4",
          "bucketName": "signer-source",

```



```
        "key": "MyCode.rb"
      }
    },
    "signedObject": {
      "s3": {
        "bucketName": "signer-destination",
        "key": "signed-2065c468-73e2-4385-a6c9-0123456789abc"
      }
    },
    "createdAt": 1568412036
  },
  {
    "status": "Failed",
    "source": {
      "s3": {
        "version": "PNyFaUTgsQh5ZdMccoCe6pT1g0pgB_M4",
        "bucketName": "signer-source",
        "key": "MyOtherCode.rb"
      }
    },
    "signingMaterial": {
      "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
    },
    "createdAt": 1568402690,
    "jobId": "74d9825e-22fc-4a0d-b962-0123456789abc"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSigningJobs](#)」の「」を参照してください。

list-signing-platforms

次のコード例は、list-signing-platforms を使用する方法を示しています。

AWS CLI

すべての署名プラットフォームを一覧表示するには

次のlist-signing-platforms例では、使用可能なすべての署名プラットフォームの詳細を表示します。

aws signer list-signing-platforms

出力:

```
{
  "platforms": [
    {
      "category": "AWS",
      "displayName": "AWS IoT Device Management SHA256-ECDSA ",
      "target": "SHA256-ECDSA",
      "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
      "signingConfiguration": {
        "encryptionAlgorithmOptions": {
          "defaultValue": "ECDSA",
          "allowedValues": [
            "ECDSA"
          ]
        },
        "hashAlgorithmOptions": {
          "defaultValue": "SHA256",
          "allowedValues": [
            "SHA256"
          ]
        }
      },
      "maxSizeInMB": 2048,
      "partner": "AWSIoTDeviceManagement",
      "signingImageFormat": {
        "defaultFormat": "JSONDetached",
        "supportedFormats": [
          "JSONDetached"
        ]
      }
    },
    {
      "category": "AWS",
      "displayName": "Amazon FreeRTOS SHA1-RSA CC3220SF-Format",
      "target": "SHA1-RSA-TISHA1",
      "platformId": "AmazonFreeRTOS-TI-CC3220SF",
      "signingConfiguration": {
        "encryptionAlgorithmOptions": {
          "defaultValue": "RSA",
          "allowedValues": [
```

```
        "RSA"
      ]
    },
    "hashAlgorithmOptions": {
      "defaultValue": "SHA1",
      "allowedValues": [
        "SHA1"
      ]
    }
  },
  "maxSizeInMB": 16,
  "partner": "AmazonFreeRTOS",
  "signingImageFormat": {
    "defaultFormat": "JSONEmbedded",
    "supportedFormats": [
      "JSONEmbedded"
    ]
  }
},
{
  "category": "AWS",
  "displayName": "Amazon FreeRTOS SHA256-ECDSA",
  "target": "SHA256-ECDSA",
  "platformId": "AmazonFreeRTOS-Default",
  "signingConfiguration": {
    "encryptionAlgorithmOptions": {
      "defaultValue": "ECDSA",
      "allowedValues": [
        "ECDSA"
      ]
    },
    "hashAlgorithmOptions": {
      "defaultValue": "SHA256",
      "allowedValues": [
        "SHA256"
      ]
    }
  },
  "maxSizeInMB": 16,
  "partner": "AmazonFreeRTOS",
  "signingImageFormat": {
    "defaultFormat": "JSONEmbedded",
    "supportedFormats": [
      "JSONEmbedded"
    ]
  }
}
```

```
    ]
  }
}
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSigningPlatforms](#)」の「」を参照してください。

list-signing-profiles

次の例は、list-signing-profiles を使用する方法を説明しています。

AWS CLI

すべての署名プロファイルを一覧表示するには

次のlist-signing-profiles例では、アカウントのすべての署名プロファイルの詳細を表示します。

```
aws signer list-signing-profiles
```

出力:

```
{
  "profiles": [
    {
      "platformId": "AmazonFreeRTOS-TI-CC3220SF",
      "profileName": "MyProfile4",
      "status": "Active",
      "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
      }
    },
    {
      "platformId": "AWSIoTDeviceManagement-SHA256-ECDSA",
      "profileName": "MyProfile5",
      "status": "Active",
      "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
      }
    }
  ]
}
```

```
    }  
  }  
]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSigningProfiles](#)」の「」を参照してください。

put-signing-profile

次のコード例は、put-signing-profile を使用する方法を示しています。

AWS CLI

署名プロファイルを作成するには

次のput-signing-profile例では、指定された証明書とプラットフォームを使用して署名プロファイルを作成します。

```
aws signer put-signing-profile \  
  --profile-name MyProfile6 \  
  --signing-material certificateArn=arn:aws:acm:us-west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc \  
  --platform AmazonFreeRTOS-TI-CC3220SF
```

出力:

```
{  
  "arn": "arn:aws:signer:us-west-2:123456789012:/signing-profiles/MyProfile6"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutSigningProfile](#)」の「」を参照してください。

start-signing-job

次のコード例は、start-signing-job を使用する方法を示しています。

AWS CLI

署名ジョブを開始するには

次のstart-signing-job例では、指定されたソースにあるコードの署名ジョブを開始します。指定されたプロファイルを使用して署名を行い、署名されたコードを指定された送信先に配置します。

```
aws signer start-signing-job \  
  --source 's3={bucketName=signer-  
source,key=MyCode.rb,version=PNyFaUTgsQh5ZdMCcoCe6pT1g0pgB_M4}' \  
  --destination 's3={bucketName=signer-destination,prefix=signed-}' \  
  --profile-name MyProfile7
```

出力は署名ジョブの ID です。

```
{  
  "jobId": "2065c468-73e2-4385-a6c9-0123456789abc"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartSigningJob](#)」の「」を参照してください。

を使用した Snowball の例 AWS CLI

次のコード例は、Snowball AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

get-snowball-usage

次の例は、get-snowball-usage を使用する方法を説明しています。

AWS CLI

アカウントの Snowball サービス制限に関する情報を取得するには

次の`get-snowball-usage`例は、アカウントの Snowball サービス制限に関する情報と、アカウントが使用している Snowball の数を示しています。

```
aws snowball get-snowball-usage
```

出力:

```
{
  "SnowballLimit": 1,
  "SnowballsInUse": 0
}
```

FOR 詳細については、[AWS Snowball デベロッパーガイドの「Snowball Edge Limits」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSnowballUsage](#)」の「」を参照してください。

list-jobs

次のコード例は、`list-jobs` を使用する方法を示しています。

AWS CLI

アカウント内の現在の Snowball ジョブを一覧表示するには

次の`list-jobs`例は、`JobListEntry`オブジェクトの配列を示しています。この例では、1 つのジョブが一覧表示されます。

```
aws snowball list-jobs
```

出力:

```
{
  "JobListEntries": [
    {
      "CreationDate": 2016-09-27T14:50Z,
```

```
    "Description": "Important Photos 2016-08-11",
    "IsMaster": TRUE,
    "JobId": "ABCd1e324fe-022f-488e-a98b-3b0566063db1",
    "JobState": "Complete",
    "JobType": "IMPORT",
    "SnowballType": "EDGE"
  }
]
}
```

詳細については、[AWS Snowball デベロッパーガイドの「Jobs for Snowball Edge devices」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListJobs](#)」の「」を参照してください。

を使用した Amazon SNS の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますSNS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

「シナリオ」は、1つのサービス内から、または他の AWS のサービスと組み合わせて複数の関数を呼び出し、特定のタスクを実行する方法を示すコード例です。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)
- [シナリオ](#)

アクション

add-permission

次の例は、add-permission を使用する方法を説明しています。

AWS CLI

トピックにアクセス許可を追加するには

次のadd-permission例では、AWS アカウント で指定されたトピックで Publishアクションを使用する987654321098アクセス許可を AWS アカウント に追加します123456789012。

```
aws sns add-permission \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --label Publish-Permission \  
  --aws-account-id 987654321098 \  
  --action-name Publish
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddPermission](#)」の「」を参照してください。

check-if-phone-number-is-opted-out

次のコード例は、check-if-phone-number-is-opted-out を使用する方法を示しています。

AWS CLI

電話番号のSMSメッセージのオプトアウトを確認するには

次のcheck-if-phone-number-is-opted-out例では、指定された電話番号が現在の AWS アカウントからのSMSメッセージの受信をオプトアウトされているかどうかを確認します。

```
aws sns check-if-phone-number-is-opted-out \  
  --phone-number +1555550100
```

出力:

```
{  
  "isOptedOut": false  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CheckIfPhoneNumberIsOptedOut](#)」の「」を参照してください。

confirm-subscription

次の例は、confirm-subscription を使用する方法を説明しています。

AWS CLI

サブスクリプションを確認するには

次のconfirm-subscriptionコマンドは、 という名前のSNSトピックをサブスクライブしたときに開始された確認プロセスを完了しますmy-topic。 --token パラメータは、subscribe の呼び出しで指定した通知エンドポイントに送信される確認メッセージから取得されます。

```
aws sns confirm-subscription \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \  
  --  
  token 2336412f37fb687f5d51e6e241d7700ae02f7124d8268910b858cb4db727ceeb2474bb937929d3bdd7ce5a
```

出力:

```
{  
  "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-  
  topic:8a21d249-4329-4871-acc6-7be709c6ea7f"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ConfirmSubscription](#)」の「」を参照してください。

create-platform-application

次の例は、create-platform-application を使用する方法を説明しています。

AWS CLI

プラットフォームアプリケーションを作成するには

次のcreate-platform-application例では、指定されたプラットフォーム認証情報を使用して Google Firebase プラットフォームアプリケーションを作成します。

```
aws sns create-platform-application \  
  --name MyApplication \  
  --
```

```
--platform GCM \  
--attributes PlatformCredential=EXAMPLEabcd12345jklm67890stuv12345bcdef
```

出力:

```
{  
  "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/GCM/  
MyApplication"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePlatformApplication](#)」の「」を参照してください。

create-topic

次のコード例は、create-topic を使用する方法を示しています。

AWS CLI

トピック SNS を作成するには

次のcreate-topic例では、 という名前のSNSトピックを作成しますmy-topic。

```
aws sns create-topic \  
--name my-topic
```

出力:

```
{  
  "ResponseMetadata": {  
    "RequestId": "1469e8d7-1642-564e-b85d-a19b4b341f83"  
  },  
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"  
}
```

詳細については、[AWS 「コマンドラインインターフェイスユーザーガイド」の「Amazon SQS および Amazon でのSNSコマンドラインインターフェイスの使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTopic](#)」の「」を参照してください。

delete-endpoint

次のコード例は、delete-endpoint を使用する方法を示しています。

AWS CLI

プラットフォームアプリケーションエンドポイントを削除するには

次のdelete-endpoint例では、指定されたプラットフォームアプリケーションエンドポイントを削除します。

```
aws sns delete-endpoint \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteEndpoint](#)」の「」を参照してください。

delete-platform-application

次の例は、delete-platform-application を使用する方法を説明しています。

AWS CLI

プラットフォームアプリケーションを削除するには

次のdelete-platform-application例では、指定されたプラットフォームアプリケーションを削除します。

```
aws sns delete-platform-application \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/ADM/  
MyApplication
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePlatformApplication](#)」の「」を参照してください。

delete-topic

次のコード例は、delete-topic を使用する方法を示しています。

AWS CLI

SNSトピックを削除するには

次のdelete-topic例では、指定されたSNSトピックを削除します。

```
aws sns delete-topic \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTopic](#)」の「」を参照してください。

get-endpoint-attributes

次のコード例は、get-endpoint-attributes を使用する方法を示しています。

AWS CLI

プラットフォームアプリケーションのエンドポイント属性を一覧表示するには

次のget-endpoint-attributes例では、指定されたプラットフォームアプリケーションエンドポイントの属性を一覧表示します。

```
aws sns get-endpoint-attributes \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/  
  MyApplication/12345678-abcd-9012-efgh-345678901234
```

出力:

```
{  
  "Attributes": {  
    "Enabled": "true",  
    "Token": "EXAMPLE12345..."  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEndpointAttributes](#)」の「」を参照してください。

get-platform-application-attributes

次のコード例は、`get-platform-application-attributes` を使用する方法を示しています。

AWS CLI

プラットフォームアプリケーション属性を一覧表示するには

次の`get-platform-application-attributes`例では、指定されたプラットフォームアプリケーションの属性を一覧表示します。

```
aws sns get-platform-application-attributes \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/MPNS/  
  MyApplication
```

出力:

```
{  
  "Attributes": {  
    "Enabled": "true",  
    "SuccessFeedbackSampleRate": "100"  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPlatformApplicationAttributes](#)」の「」を参照してください。

get-sms-attributes

次の例は、`get-sms-attributes` を使用する方法を説明しています。

AWS CLI

デフォルトのSMSメッセージ属性を一覧表示するには

次の`get-sms-attributes`例では、SMSメッセージを送信するためのデフォルトの属性を一覧表示します。

```
aws sns get-sms-attributes
```

出力:

```
{
  "attributes": {
    "DefaultSenderId": "MyName"
  }
}
```

- API 詳細については、「コマンドリファレンス」の「[GetSMSAttributes](#)」を参照してください。AWS CLI

get-subscription-attributes

次の例は、get-subscription-attributes を使用する方法を説明しています。

AWS CLI

トピックのサブスクリプション属性を取得するには

以下に、指定されたサブスクリプションの属性get-subscription-attributesを示します。list-subscriptions コマンドの出力subscription-arnから を取得できます。

```
aws sns get-subscription-attributes \
  --subscription-arn "arn:aws:sns:us-west-2:123456789012:my-
  topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

出力:

```
{
  "Attributes": {
    "Endpoint": "my-email@example.com",
    "Protocol": "email",
    "RawMessageDelivery": "false",
    "ConfirmationWasAuthenticated": "false",
    "Owner": "123456789012",
    "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
  topic:8a21d249-4329-4871-acc6-7be709c6ea7f",
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
  }
}
```

```
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSubscriptionAttributes](#)」の「」を参照してください。

get-topic-attributes

次のコード例は、get-topic-attributes を使用する方法を示しています。

AWS CLI

トピックの属性を取得するには

次の get-topic-attributes の例では、指定したトピックの属性を表示します。

```
aws sns get-topic-attributes \
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

出力:

```
{
  "Attributes": {
    "SubscriptionsConfirmed": "1",
    "DisplayName": "my-topic",
    "SubscriptionsDeleted": "0",
    "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":{\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries\":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear\"},\"disableSubscriptionOverrides\":false}}",
    "Owner": "123456789012",
    "Policy": "{\"Version\":\"2008-10-17\",\"Id\":\"__default_policy_ID\", \"Statement\": [{\"Sid\":\"__default_statement_ID\", \"Effect\":\"Allow\", \"Principal\": {\"AWS\":\"*\"}, \"Action\": [\"SNS:Subscribe\", \"SNS:ListSubscriptionsByTopic\", \"SNS:DeleteTopic\", \"SNS:GetTopicAttributes\", \"SNS:Publish\", \"SNS:RemovePermission\", \"SNS:AddPermission\", \"SNS:SetTopicAttributes\"], \"Resource\": \"arn:aws:sns:us-west-2:123456789012:my-topic\", \"Condition\": {\"StringEquals\": {\"AWS:SourceOwner\": \"0123456789012\"}}}]}",
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
    "SubscriptionsPending": "0"
  }
}
```



```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTopicAttributes](#)」の「」を参照してください。

list-endpoints-by-platform-application

次の例は、list-endpoints-by-platform-application を使用する方法を説明しています。

AWS CLI

プラットフォームアプリケーションのエンドポイントを一覧表示するには

次のlist-endpoints-by-platform-application例では、指定されたプラットフォームアプリケーションのエンドポイントとエンドポイント属性を一覧表示します。

```
aws sns list-endpoints-by-platform-application \
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication
```

出力:

```
{
  "Endpoints": [
    {
      "Attributes": {
        "Token": "EXAMPLE12345...",
        "Enabled": "true"
      },
      "EndpointArn": "arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListEndpointsByPlatformApplication](#)」の「」を参照してください。

list-phone-numbers-opted-out

次の例は、list-phone-numbers-opted-out を使用する方法を説明しています。

AWS CLI

SMS メッセージオプトアウトを一覧表示するには

次のlist-phone-numbers-opted-out例では、SMSメッセージの受信をオプトアウトされた電話番号を一覧表示します。

```
aws sns list-phone-numbers-opted-out
```

出力:

```
{
  "phoneNumbers": [
    "+15555550100"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPhoneNumbersOptedOut](#)」の「」を参照してください。

list-platform-applications

次の例は、list-platform-applications を使用する方法を説明しています。

AWS CLI

プラットフォームアプリケーションを一覧表示するには

次のlist-platform-applications例は、ADMと のプラットフォームアプリケーションを示していますMPNS。

```
aws sns list-platform-applications
```

出力:

```
{
  "PlatformApplications": [
    {
      "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/ADM/MyApplication",
      "Attributes": {
        "SuccessFeedbackSampleRate": "100",
```

```
        "Enabled": "true"
      }
    },
    {
      "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/MPNS/MyOtherApplication",
      "Attributes": {
        "SuccessFeedbackSampleRate": "100",
        "Enabled": "true"
      }
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPlatformApplications](#)」の「」を参照してください。

list-subscriptions-by-topic

次の例は、list-subscriptions-by-topic を使用する方法を説明しています。

AWS CLI

トピックに関連付けられたサブスクリプションを一覧表示するには

以下は、指定されたトピックに関連付けられたSNSサブスクリプションのリストlist-subscriptions-by-topicを取得します。

```
aws sns list-subscriptions-by-topic \
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

出力:

```
{
  "Subscriptions": [
    {
      "Owner": "123456789012",
      "Endpoint": "my-email@example.com",
      "Protocol": "email",
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",
      "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
    }
  ]
}
```

```
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSubscriptionsByTopic](#)」の「」を参照してください。

list-subscriptions

次の例は、list-subscriptions を使用する方法を説明しています。

AWS CLI

SNSサブスクリプションを一覧表示するには

次のlist-subscriptions例では、AWS アカウントのSNSサブスクリプションのリストを表示します。

```
aws sns list-subscriptions
```

出力:

```
{  
  "Subscriptions": [  
    {  
      "Owner": "123456789012",  
      "Endpoint": "my-email@example.com",  
      "Protocol": "email",  
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic",  
      "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListSubscriptions](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

トピックのタグを一覧表示するには

次のlist-tags-for-resource例では、指定された Amazon SNS トピックのタグを一覧表示します。

```
aws sns list-tags-for-resource \  
--resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic
```

出力:

```
{  
  "Tags": [  
    {  
      "Key": "Team",  
      "Value": "Alpha"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-topics

次の例は、list-topics を使用する方法を説明しています。

AWS CLI

SNS トピックを一覧表示するには

次のlist-topics例では、AWS アカウント内のすべてのSNS トピックを一覧表示します。

```
aws sns list-topics
```

出力:

```
{  
  "Topics": [  
    {
```

```
        "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
    }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTopics](#)」の「」を参照してください。

opt-in-phone-number

次のコード例は、opt-in-phone-number を使用する方法を示しています。

AWS CLI

SMS メッセージにオプトインするには

次のopt-in-phone-number例では、指定された電話番号をSMSメッセージの受信にオプトします。

```
aws sns opt-in-phone-number \  
  --phone-number +15555550100
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[OptInPhoneNumber](#)」の「」を参照してください。

publish

次のコード例は、publish を使用する方法を示しています。

AWS CLI

例 1: トピックにメッセージを発行するには

次のpublish例では、指定されたメッセージを指定されたSNSトピックに発行します。メッセージはテキストファイルから取得されたもので、改行を含めることができます。

```
aws sns publish \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic" \  
  --message file://message.txt
```

message.txt の内容:

```
Hello World
Second Line
```

出力:

```
{
  "MessageId": "123a45b6-7890-12c3-45d6-111122223333"
}
```

例 2: メッセージを電話番号SMSに発行するには

次の publish の例では、Hello world! メッセージを電話番号 +1-555-555-0100 に公開します。

```
aws sns publish \
  --message "Hello world!" \
  --phone-number +1-555-555-0100
```

出力:

```
{
  "MessageId": "123a45b6-7890-12c3-45d6-333322221111"
}
```

• API 詳細については、「コマンドリファレンス」の [「発行」](#) を参照してください。AWS CLI

put-data-protection-policy

次の例は、put-data-protection-policy を使用する方法を説明しています。

AWS CLI

データ保護ポリシーを設定するには

例 1: パブリッシャーが を使用してメッセージを発行することを拒否するには CreditCardNumber

次のput-data-protection-policy例では、パブリッシャーが を使用してメッセージを公開することを拒否しています CreditCardNumber。

```
aws sns put-data-protection-policy \  
  --resource-arn arn:aws:sns:us-east-1:123456789012:mytopic \  
  --data-protection-policy '{"Name":"data_protection_policy","Description\  
":"Example data protection policy","Version":"2021-06-01","Statement\  
":[{"DataDirection":"Inbound","Principal":["*"],"DataIdentifier":\  
["arn:aws:dataprotection::aws:data-identifier/CreditCardNumber"],"Operation":\  
{"Deny":{}}]}'
```

このコマンドでは何も出力されません。

例 2: ファイルからパラメータをロードするには

次の は、ファイルからパラメータをput-data-protection-policyロードします。

```
aws sns put-data-protection-policy \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --data-protection-policy file://policy.json
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutDataProtectionPolicy](#)」の「」を参照してください。

remove-permission

次のコード例は、remove-permission を使用する方法を示しています。

AWS CLI

トピックからアクセス許可を削除するには

次のremove-permission例では、指定されたトピックPublish-Permissionからアクセス許可を削除します。

```
aws sns remove-permission \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --label Publish-Permission
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[RemovePermission](#)」の「」を参照してください。

set-endpoint-attributes

次のコード例は、set-endpoint-attributes を使用する方法を示しています。

AWS CLI

エンドポイント属性を設定するには

次のset-endpoint-attributes例では、指定されたプラットフォームアプリケーションエンドポイントを無効にします。

```
aws sns set-endpoint-attributes \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234 \  
  --attributes Enabled=false
```

出力:

```
{  
  "Attributes": {  
    "Enabled": "false",  
    "Token": "EXAMPLE12345..."  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetEndpointAttributes](#)」の「」を参照してください。

set-platform-application-attributes

次のコード例は、set-platform-application-attributes を使用する方法を示しています。

AWS CLI

プラットフォームアプリケーション属性を設定するには

次のset-platform-application-attributes例では、指定されたプラットフォームアプリケーションの EventDeliveryFailure 属性を、指定された Amazon SNS トピックARNの に設定します。

```
aws sns set-platform-application-attributes \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --platform-application-attributes EventDeliveryFailure=true
```

```
--platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/MyApplication \  
--attributes EventDeliveryFailure=arn:aws:sns:us-west-2:123456789012:AnotherTopic
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetPlatformApplicationAttributes](#)」の「」を参照してください。

set-sms-attributes

次の例は、set-sms-attributes を使用する方法を説明しています。

AWS CLI

SMS メッセージ属性を設定するには

次のset-sms-attributes例では、SMSメッセージのデフォルトの送信者 ID を に設定しますMyName。

```
aws sns set-sms-attributes \  
--attributes DefaultSenderId=MyName
```

このコマンドでは何も出力されません。

- API 詳細については、「コマンドリファレンス」の「[SetSMSAttributes](#)」を参照してください。AWS CLI

set-subscription-attributes

次の例は、set-subscription-attributes を使用する方法を説明しています。

AWS CLI

サブスクリプション属性を設定するには

次のset-subscription-attributes例では、RawMessageDelivery 属性をSQSサブスクリプションに設定します。

```
aws sns set-subscription-attributes \  

```

```
--subscription-arn arn:aws:sns:us-east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \  
--attribute-name RawMessageDelivery \  
--attribute-value true
```

このコマンドでは何も出力されません。

次のset-subscription-attributes例では、FilterPolicy 属性をSQSサブスクリプションに設定します。

```
aws sns set-subscription-attributes \  
  --subscription-arn arn:aws:sns:us-east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \  
  --attribute-name FilterPolicy \  
  --attribute-value "{ \"anyMandatoryKey\": [\"any\", \"of\", \"these\"] }"
```

このコマンドでは何も出力されません。

次のset-subscription-attributes例では、SQSサブスクリプションから FilterPolicy 属性を削除します。

```
aws sns set-subscription-attributes \  
  --subscription-arn arn:aws:sns:us-east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \  
  --attribute-name FilterPolicy \  
  --attribute-value "{}"
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetSubscriptionAttributes](#)」の「」を参照してください。

set-topic-attributes

次の例は、set-topic-attributes を使用する方法を説明しています。

AWS CLI

トピックの属性を設定するには

次の set-topic-attributes の例では、指定したトピックの DisplayName 属性を設定します。

```
aws sns set-topic-attributes \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --attribute-name DisplayName \  
  --attribute-value MyTopicDisplayName
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[SetTopicAttributes](#)」の「」を参照してください。

subscribe

次の例は、subscribe を使用する方法を説明しています。

AWS CLI

トピックにサブスクライブするには

次の subscribe コマンドは、指定したトピックに E メールアドレスをサブスクライブします。

```
aws sns subscribe \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \  
  --protocol email \  
  --notification-endpoint my-email@example.com
```

出力:

```
{  
  "SubscriptionArn": "pending confirmation"  
}
```

- API 詳細については、「コマンドリファレンス」の「[サブスクライブ](#)」を参照してください。

AWS CLI

tag-resource

次の例は、tag-resource を使用する方法を説明しています。

AWS CLI

トピックにタグを追加するには

次のtag-resource例では、指定された Amazon SNSトピックにメタデータタグを追加します。

```
aws sns tag-resource \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --tags Key=Team,Value=Alpha
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

unsubscribe

次の例は、unsubscribe を使用する方法を説明しています。

AWS CLI

トピックからサブスクライブを解除するには

次の unsubscribe の例では、指定したサブスクリプションをトピックから削除します。

```
aws sns unsubscribe \  
  --subscription-arn arn:aws:sns:us-west-2:0123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス」の「[サブスクリプション解除](#)」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

トピックからタグを削除するには

次のuntag-resource例では、指定された Amazon SNSトピックから指定されたキーを持つタグを削除します。

```
aws sns untag-resource \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --tag-keys Team
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [UntagResource](#)」の「」を参照してください。

シナリオ

プッシュ通知のプラットフォームエンドポイントを作成します

次のコード例は、Amazon SNSプッシュ通知用のプラットフォームエンドポイントを作成する方法を示しています。

AWS CLI

プラットフォームアプリケーションのエンドポイントを作成するには

次の `create-platform-endpoint` の例では、指定したトークンを使用して、指定したプラットフォームアプリケーションのエンドポイントを作成します。

```
aws sns create-platform-endpoint \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/MyApplication \  
  --token EXAMPLE12345...
```

出力:

```
{  
  "EndpointArn": "arn:aws:sns:us-west-2:1234567890:endpoint/GCM/MyApplication/12345678-abcd-9012-efgh-345678901234"  
}
```

を使用した Amazon SQS の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますSQS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-permission

次の例は、`add-permission` を使用する方法を説明しています。

AWS CLI

キューにアクセス許可を追加するには

この例では、指定された AWS アカウントが指定されたキューにメッセージを送信できるようにします。

コマンド:

```
aws sqs add-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue --aws-account-ids 12345EXAMPLE --actions SendMessage
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AddPermission](#)」の「」を参照してください。

cancel-message-move-task

次の例は、`cancel-message-move-task` を使用する方法を説明しています。

AWS CLI

メッセージ移動タスクをキャンセルするには

次のcancel-message-move-task例では、指定されたメッセージ移動タスクをキャンセルします。

```
aws sqs cancel-message-move-task \  
  --task-handle AQEB6nR4...HzLvZQ==
```

出力:

```
{  
  "ApproximateNumberOfMessagesMoved": 102  
}
```

詳細については、「デベロッパーガイド」の「[Amazon アクセスSQSAPI許可: アクションとリソースリファレンス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelMessageMoveTask](#)」の「」を参照してください。

change-message-visibility-batch

次の例は、change-message-visibility-batch を使用する方法を説明しています。

AWS CLI

複数のメッセージのタイムアウトビジビリティをバッチとして変更するには

この例では、2つの指定されたメッセージのタイムアウトの可視性を 10 時間 (10 時間 x 60 分 x 60 秒) に変更します。

コマンド:

```
aws sqs change-message-visibility-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://change-message-visibility-batch.json
```

入力ファイル (change-message-visibility-batch.json):


```
[
  {
    "Id": "FirstMessage",
    "ReceiptHandle": "AQEBhz2q...Jf3kaw==",
    "VisibilityTimeout": 36000
  },
  {
    "Id": "SecondMessage",
    "ReceiptHandle": "AQEBkTUH...HifSnw==",
    "VisibilityTimeout": 36000
  }
]
```

出力:

```
{
  "Successful": [
    {
      "Id": "SecondMessage"
    },
    {
      "Id": "FirstMessage"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ChangeMessageVisibilityBatch](#)」の「」を参照してください。

change-message-visibility

次のコード例は、change-message-visibility を使用する方法を示しています。

AWS CLI

メッセージのタイムアウトの可視性を変更するには

この例は、指定されたメッセージのタイムアウトの可視性を 10 時間 (10 時間 × 60 分 × 60 秒) に変更します。

コマンド:

```
aws sqs change-message-visibility --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBTpyI...t6HyQg== --visibility-timeout 36000
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ChangeMessageVisibility](#)」の「」を参照してください。

create-queue

次のコード例は、create-queue を使用する方法を示しています。

AWS CLI

キューを作成するには

この例は、指定された名前のキューを作成して、メッセージの保持期間を 3 日間 (3 日 × 24 時間 × 60 分 × 60 秒) に設定します。またキューのデッドレターキューを、最大受信数 1,000 件のメッセージを含む指定されたキューに設定します。

コマンド:

```
aws sqs create-queue --queue-name MyQueue --attributes file://create-queue.json
```

入力ファイル (create-queue.json):

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\", \"maxReceiveCount\": \"1000\"}\",
  "MessageRetentionPeriod": "259200"
}
```

出力:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateQueue](#)」の「」を参照してください。

delete-message-batch

次の例は、delete-message-batch を使用する方法を説明しています。

AWS CLI

複数のメッセージを一括削除するには

この例は、指定されたメッセージを削除します。

コマンド:

```
aws sqs delete-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://delete-message-batch.json
```

入力ファイル (delete-message-batch.json):

```
[
  {
    "Id": "FirstMessage",
    "ReceiptHandle": "AQEB1mg1...Z4GuLw=="
  },
  {
    "Id": "SecondMessage",
    "ReceiptHandle": "AQEBLsYM...VQubAA=="
  }
]
```

出力:

```
{
  "Successful": [
    {
      "Id": "FirstMessage"
    },
    {
      "Id": "SecondMessage"
    }
  ]
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMessageBatch](#)」の「」を参照してください。

delete-message

次のコード例は、delete-message を使用する方法を示しています。

AWS CLI

単一のメッセージを削除するには

この例は、指定された単一のメッセージを削除します。

コマンド:

```
aws sqs delete-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBRXTo...q2doVA==
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMessage](#)」の「」を参照してください。

delete-queue

次のコード例は、delete-queue を使用する方法を示しています。

AWS CLI

キューを削除するには

この例は、指定されたキューを削除します。

コマンド:

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteQueue](#)」の「」を参照してください。

get-queue-attributes

次のコード例は、`get-queue-attributes` を使用する方法を示しています。

AWS CLI

キューの属性を取得するには

この例では、指定されたキューの属性をすべて取得します。

コマンド:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All
```

出力:

```
{
  "Attributes": {
    "ApproximateNumberOfMessagesNotVisible": "0",
    "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":1000}",
    "MessageRetentionPeriod": "345600",
    "ApproximateNumberOfMessagesDelayed": "0",
    "MaximumMessageSize": "262144",
    "CreatedTimestamp": "1442426968",
    "ApproximateNumberOfMessages": "0",
    "ReceiveMessageWaitTimeSeconds": "0",
    "DelaySeconds": "0",
    "VisibilityTimeout": "30",
    "LastModifiedTimestamp": "1442426968",
    "QueueArn": "arn:aws:sqs:us-east-1:80398EXAMPLE:MyNewQueue"
  }
}
```

この例は、指定されたキューの最大メッセージサイズと可視性タイムアウト属性のみを取得します。

コマンド:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attribute-names MaximumMessageSize VisibilityTimeout
```

出力:

```
{
  "Attributes": {
    "VisibilityTimeout": "30",
    "MaximumMessageSize": "262144"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetQueueAttributes](#)」の「」を参照してください。

get-queue-url

次のコード例は、get-queue-url を使用する方法を示しています。

AWS CLI

キューを取得するには URL

この例では、指定されたキューの を取得しますURL。

コマンド:

```
aws sqs get-queue-url --queue-name MyQueue
```

出力:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetQueueUrl](#)」の「」を参照してください。

list-dead-letter-source-queues

次の例は、list-dead-letter-source-queues を使用する方法を説明しています。

AWS CLI

デッドレターソースキューを一覧表示するには

この例では、指定されたデッドレターソースキューに関連付けられているキューを一覧表示します。

コマンド:

```
aws sqs list-dead-letter-source-queues --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue
```

出力:

```
{
  "queueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDeadLetterSourceQueues](#)」の「」を参照してください。

list-message-move-tasks

次の例は、list-message-move-tasks を使用する方法を説明しています。

AWS CLI

メッセージ移動タスクを一覧表示するには

次のlist-message-move-tasks例では、指定されたキュー内の最新のメッセージ移動タスクを2つ一覧表示します。

```
aws sqs list-message-move-tasks \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue \  
  --max-results 2
```

出力:

```
{  
  "Results": [  
    {  
      "TaskHandle": "AQEB6nR4...Hz1vZQ==",  
      "Status": "RUNNING",  
      "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",  
      "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",  
      "MaxNumberOfMessagesPerSecond": 50,  
      "ApproximateNumberOfMessagesMoved": 203,  
      "ApproximateNumberOfMessagesToMove": 30,  
      "StartedTimestamp": 1442428276921  
    },  
    {  
      "Status": "COMPLETED",  
      "SourceArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1",  
      "DestinationArn": "arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2",  
      "ApproximateNumberOfMessagesMoved": 29,  
      "ApproximateNumberOfMessagesToMove": 0,  
      "StartedTimestamp": 1342428272093  
    }  
  ]  
}
```

詳細については、「デベロッパーガイド」の「[Amazon アクセスSQSAPI許可: アクションとリソースリファレンス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMessageMoveTasks](#)」の「」を参照してください。

list-queue-tags

次のコード例は、list-queue-tags を使用する方法を示しています。

AWS CLI

キューのすべてのコスト配分タグを一覧表示するには

次のlist-queue-tags例では、指定されたキューに関連付けられているすべてのコスト配分タグを表示します。

```
aws sqs list-queue-tags \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue
```

出力:

```
{  
  "Tags": {  
    "Team": "Alpha"  
  }  
}
```

詳細については、「Amazon Simple Queue Service デベロッパーガイド」の「[コスト配分タグの一覧表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListQueueTags](#)」の「」を参照してください。

list-queues

次のコード例は、list-queues を使用する方法を示しています。

AWS CLI

キューを一覧表示するには

この例は、すべてのキューを一覧表示します。

コマンド:

```
aws sqs list-queues
```

出力:

```
{  
  "QueueUrls": [  
    "
```

```
"https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
"https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
"https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue",
"https://queue.amazonaws.com/80398EXAMPLE/TestQueue1",
  "https://queue.amazonaws.com/80398EXAMPLE/TestQueue2"
]
}
```

この例は、「My」で始まるキューのみを一覧表示します。

コマンド:

```
aws sqs list-queues --queue-name-prefix My
```

出力:

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListQueues](#)」の「」を参照してください。

purge-queue

次のコード例は、purge-queue を使用する方法を示しています。

AWS CLI

キューを消去するには

この例では、指定されたキュー内のすべてのメッセージを削除します。

コマンド:

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PurgeQueue](#)」の「」を参照してください。

receive-message

次のコード例は、receive-message を使用する方法を示しています。

AWS CLI

メッセージを受信するには

この例は、最大 10 件のメッセージを受信し、使用可能なすべての属性を返します。

コマンド:

```
aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All --message-attribute-names All --max-number-of-messages 10
```

出力:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEBzbVv...fqNzFw==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "9424c491...26bc3ae7",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "ApproximateFirstReceiveTimestamp": "1442428276921",
        "SenderId": "AIDAIAZKMSNQ7TEXAMPLE",
        "ApproximateReceiveCount": "5",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
```

```
    "DataType": "String",
    "StringValue": "ABC123"
  },
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  }
}
]
}
```

この例では、次に使用可能なメッセージを受信し、`SenderId` および `SentTimestamp` 属性と `PostalCode` メッセージ属性のみを返します。

コマンド:

```
aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names SenderId SentTimestamp --message-attribute-names PostalCode
```

出力:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEB6nR4...HzlvZQ==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "b8e89563...e088e74f",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "SenderId": "AIDAIKMSNQ7EXAMPLE",
        "SentTimestamp": "1442428276921"
      }
    },
    "MessageAttributes": {
      "PostalCode": {
        "DataType": "String",
        "StringValue": "ABC123"
      }
    }
  ]
}
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReceiveMessage](#)」の「」を参照してください。

remove-permission

次のコード例は、remove-permission を使用する方法を示しています。

AWS CLI

アクセス許可を削除するには

この例では、指定されたキューから、指定されたラベルを持つアクセス許可を削除します。

コマンド:

```
aws sqs remove-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemovePermission](#)」の「」を参照してください。

send-message-batch

次のコード例は、send-message-batch を使用する方法を示しています。

AWS CLI

複数のメッセージを一括送信するには

この例は、メッセージ本文、遅延期間、およびメッセージ属性を指定した 2 つのメッセージを指定されたキューに送信します。

コマンド:

```
aws sqs send-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://send-message-batch.json
```

入力ファイル (send-message-batch.json):

```
[
  {
    "Id": "FuelReport-0001-2015-09-16T140731Z",
    "MessageBody": "Fuel report for account 0001 on 2015-09-16 at 02:07:31 PM.",
    "DelaySeconds": 10,
    "MessageAttributes": {
      "SellerName": {
        "DataType": "String",
        "StringValue": "Example Store"
      },
      "City": {
        "DataType": "String",
        "StringValue": "Any City"
      },
      "Region": {
        "DataType": "String",
        "StringValue": "WA"
      },
      "PostalCode": {
        "DataType": "String",
        "StringValue": "99065"
      },
      "PricePerGallon": {
        "DataType": "Number",
        "StringValue": "1.99"
      }
    }
  },
  {
    "Id": "FuelReport-0002-2015-09-16T140930Z",
    "MessageBody": "Fuel report for account 0002 on 2015-09-16 at 02:09:30 PM.",
    "DelaySeconds": 10,
    "MessageAttributes": {
      "SellerName": {
        "DataType": "String",
        "StringValue": "Example Fuels"
      },
      "City": {
```

```
    "DataType": "String",
    "StringValue": "North Town"
  },
  "Region": {
    "DataType": "String",
    "StringValue": "WA"
  },
  "PostalCode": {
    "DataType": "String",
    "StringValue": "99123"
  },
  "PricePerGallon": {
    "DataType": "Number",
    "StringValue": "1.87"
  }
}
]
```

出力:

```
{
  "Successful": [
    {
      "MD5ofMessageBody": "203c4a38...7943237e",
      "MD5ofMessageAttributes": "10809b55...baf283ef",
      "Id": "FuelReport-0001-2015-09-16T140731Z",
      "MessageId": "d175070c-d6b8-4101-861d-adeb3EXAMPLE"
    },
    {
      "MD5ofMessageBody": "2cf0159a...c1980595",
      "MD5ofMessageAttributes": "55623928...ae354a25",
      "Id": "FuelReport-0002-2015-09-16T140930Z",
      "MessageId": "f9b7d55d-0570-413e-b9c5-a9264EXAMPLE"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [SendMessageBatch](#)」の「」を参照してください。

send-message

次の例は、send-message を使用する方法を説明しています。

AWS CLI

単一のメッセージを送信するには

この例は、指定された単一のメッセージ本文、遅延期間、メッセージ属性を含むメッセージを指定されたキューに送信します。

コマンド:

```
aws sqs send-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --message-body "Information about the largest city in Any Region." --delay-seconds 10 --message-attributes file://send-message.json
```

入力ファイル (send-message.json):

```
{
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  },
  "Greeting": {
    "DataType": "Binary",
    "BinaryValue": "Hello, World!"
  },
  "Population": {
    "DataType": "Number",
    "StringValue": "1250800"
  }
}
```

出力:

```
{
  "MD5ofMessageBody": "51b0a325...39163aa0",
  "MD5ofMessageAttributes": "00484c68...59e48f06",
  "MessageId": "da68f62c-0c07-4bee-bf5f-7e856EXAMPLE"
}
```


- API 詳細については、AWS CLI 「コマンドリファレンス[SendMessage](#)」の「」を参照してください。

set-queue-attributes

次の例は、set-queue-attributes を使用する方法を説明しています。

AWS CLI

キューの属性を設定するには

この例は、指定されたキューの配信遅延時間を 10 秒、最大メッセージサイズを 128 KB (128 KB × 1,024 バイト)、メッセージ保持期間を 3 日間 (3 日 × 24 時間 × 60 分 × 60 秒)、受信メッセージ待機時間を 20 秒、デフォルトの可視性タイムアウトを 60 秒に設定します。また、この例では、指定されたデッドレターキューの最大受信数を 1,000 メッセージと関連付けます。

コマンド:

```
aws sqs set-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attributes file://set-queue-attributes.json
```

入力ファイル (set-queue-attributes.json):

```
{
  "DelaySeconds": "10",
  "MaximumMessageSize": "131072",
  "MessageRetentionPeriod": "259200",
  "ReceiveMessageWaitTimeSeconds": "20",
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":\"1000\"}",
  "VisibilityTimeout": "60"
}
```

出力:

```
None.
```

- API 詳細については、AWS CLI 「コマンドリファレンス[SetQueueAttributes](#)」の「」を参照してください。

start-message-move-task

次のコード例は、start-message-move-task を使用する方法を示しています。

AWS CLI

例 1: *メッセージ移動タスクを開始するには*

次のstart-message-move-task例では、メッセージ移動タスクを開始して、指定されたデッドレターキューからソースキューにメッセージをリドライブします。

```
aws sqs start-message-move-task \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

出力:

```
{  
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="  
}
```

詳細については、「ガイドの名前」の [「これはトピックタイトルです」](#) を参照してください。

例 2: *最大レートでメッセージ移動タスクを開始するには*

次のstart-message-move-task例では、メッセージ移動タスクを開始して、指定されたデッドレターキューから指定された送信先キューに最大 1 秒あたり 50 件のメッセージでメッセージをリドライブします。

```
aws sqs start-message-move-task \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue1 \  
  --destination-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue2 \  
  --max-number-of-messages-per-second 50
```

出力:

```
{  
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="  
}
```

詳細については、「デベロッパーガイド」の [「Amazon アクセスSQSAPI許可: アクションとリソースリファレンス」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartMessageMoveTask](#)」の「」を参照してください。

tag-queue

次の例は、tag-queue を使用する方法を説明しています。

AWS CLI

コスト配分タグをキューに追加するには

次のtag-queue例では、指定された Amazon SQSキューにコスト配分タグを追加します。

```
aws sqs tag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tags Priority=Highest
```

このコマンドでは何も出力されません。

詳細については、「Amazon Simple Queue Service デベロッパーガイド」の「[コスト配分タグの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagQueue](#)」の「」を参照してください。

untag-queue

次の例は、untag-queue を使用する方法を説明しています。

AWS CLI

キューからコスト配分タグを削除するには

次のuntag-queue例では、指定された Amazon SQSキューからコスト配分タグを削除します。

```
aws sqs untag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tag-keys "Priority"
```

このコマンドでは何も出力されません。

詳細については、「Amazon Simple Queue Service デベロッパーガイド」の「[コスト配分タグの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagQueue](#)」の「」を参照してください。

を使用した Storage Gateway の例 AWS CLI

次のコード例は、Storage Gateway AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

describe-gateway-information

次の例は、describe-gateway-information を使用する方法を説明しています。

AWS CLI

ゲートウェイを記述するには

次のdescribe-gateway-informationコマンドは、指定されたゲートウェイに関するメタデータを返します。記述するゲートウェイを指定するには、コマンドでゲートウェイの Amazon リソースネーム (ARN) を使用します。

この例では、アカウント sgw-12A3456B の ID を持つゲートウェイを指定します123456789012。

```
aws storagegateway describe-gateway-information --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

このコマンドは、名前、ネットワークインターフェイス、設定されたタイムゾーン、状態 (ゲートウェイが実行されているかどうか) など、ゲートウェイに関するメタデータを含むJSONブロックを出力します。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGatewayInformation](#)」の「」を参照してください。

list-file-shares

次のコード例は、list-file-shares を使用する方法を示しています。

AWS CLI

ファイル共有を一覧表示するには

次のcommand-name例では、AWS アカウントで使用可能なウィジェットを一覧表示します。

```
aws storagegateway list-file-shares \  
--gateway-arn arn:aws:storagegateway:us-east-1:209870788375:gateway/sgw-FB02E292
```

出力:

```
{  
  "FileShareInfoList": [  
    {  
      "FileShareType": "NFS",  
      "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/  
share-2FA12345",  
      "FileShareId": "share-2FA12345",  
      "FileShareStatus": "AVAILABLE",  
      "GatewayARN": "arn:aws:storagegateway:us-east-1:111122223333:gateway/  
sgw-FB0AAAAA"  
    }  
  ],  
  "Marker": null  
}
```

詳細については、Storage Gateway サービスリファレンス[ListFileShares](#)の「」を参照してください。AWS Storage Gateway API

- API 詳細については、AWS CLI 「コマンドリファレンス[ListFileShares](#)」の「」を参照してください。

list-gateways

次のコード例は、list-gateways を使用する方法を示しています。

AWS CLI

アカウントのゲートウェイを一覧表示するには

次のlist-gatewaysコマンドは、アカウントに定義されているすべてのゲートウェイを一覧表示します。

```
aws storagegateway list-gateways
```

このコマンドは、ゲートウェイ Amazon リソースネーム () のリストを含むJSONブロックを出力しますARNs。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGateways](#)」の「」を参照してください。

list-volumes

次の例は、list-volumes を使用する方法を説明しています。

AWS CLI

ゲートウェイ用に設定されたボリュームを一覧表示するには

次のlist-volumesコマンドは、指定されたゲートウェイ用に設定されたボリュームのリストを返します。記述するゲートウェイを指定するには、コマンドでゲートウェイの Amazon リソースネーム (ARN) を使用します。

この例では、アカウント sgw-12A3456Bの ID を持つゲートウェイを指定します123456789012。

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

このコマンドは、タイプと各ボリュームの を含むボリュームのリストであるJSONブロックを出力ARNします。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVolumes](#)」の「」を参照してください。

refresh-cache

次のコード例は、refresh-cache を使用する方法を示しています。

AWS CLI

ファイル共有キャッシュを更新するには

次のrefresh-cache例では、指定されたファイル共有のキャッシュを更新します。

```
aws storagegateway refresh-cache \  
  --file-share-arn arn:aws:storagegateway:us-east-1:111122223333:share/  
share-2FA12345
```

出力:

```
{  
  "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/  
share-2FA12345",  
  "NotificationId": "4954d4b1-abcd-ef01-1234-97950a7d3483"  
}
```

詳細については、AWS Storage Gateway サービスAPIリファレンス[ListFileShares](#)の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RefreshCache](#)」の「」を参照してください。

AWS STS を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS STS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

assume-role-with-saml

次の例は、assume-role-with-saml を使用する方法を説明しています。

AWS CLI

で認証されたロールの短期認証情報を取得するには SAML

次のassume-role-with-samlコマンドは、IAMロールの短期認証情報のセットを取得しますTestSaml。この例のリクエストは、認証時に ID プロバイダーから提供されたSAMLアサーションを使用して認証されます。

```
aws sts assume-role-with-saml \  
  --role-arn arn:aws:iam::123456789012:role/TestSaml \  
  --principal-arn arn:aws:iam::123456789012:saml-provider/SAML-test \  
  --saml-  
  assertion "VERYLONGCODEDASSERTIONEXAMPLExyzYW1s0kF1ZG1lbnNlPmJsYW5rPC9zYW1s0kF1ZG1lbnNlPjwv  
+PHNhbw6TmFtZULEIEZvcm1hdD0idXJu0m9hc2lz0m5hbWVz0nRj0lNBTUw6Mi4w0m5hbWVpZC1mb3JtYXQ6dHJhbnN  
+PHNhbw6U3ViamVjdENvbmZpcm1hdGlvbiBNZXRob2Q9InVybJpvYXNpczpuYW1lczp0YzptQU1M0jIuMDpjbTpiZWwv"
```

出力:

```
{  
  "Issuer": "https://integ.example.com/idp/shibboleth</Issuer",  
  "AssumedRoleUser": {  
    "Arn": "arn:aws:sts::123456789012:assumed-role/TestSaml",  
    "AssumedRoleId": "AR0456EXAMPLE789:TestSaml"  
  },  
  "Credentials": {  
    "AccessKeyId": "ASIAV3ZUEFP6EXAMPLE",  
    "SecretAccessKey": "8P+SQvWIuLnKhh8d++jpw0nNmQRBZvNEXAMPLEKEY",  
    "SessionToken": "IQoJb3JpZ2luX2VjE0z/////////////////////////////////  
wEXAMPLEtMSJHMEUCIDoKK3JH9uGQE1z0sINr5M4jk  
+Na8KHDcCYRVjJCZEv0AiEA30vJGtw1EcVi0leS2vhs8VdCKFJQPQrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburED  
+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz  
+scqKmlzm8FDrypNC9Yjc8fP0Ln9FX9KSYvKTr4rvx3iSI1TJabIQwj2ICCR/oLxBA==",  
    "Expiration": "2019-11-01T20:26:47Z"  
  },  
}
```



```

    "Audience": "https://signin.aws.amazon.com/saml",
    "SubjectType": "transient",
    "PackedPolicySize": "6",
    "NameQualifier": "SbdG0nUkh1i4+EXAMPLExL/jEvs=",
    "Subject": "SamlExample"
  }

```

詳細については、AWS IAM「[ユーザーガイド](#)」の「[一時的なセキュリティ認証情報のリクエスト](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス AssumeRoleWithSaml](#)」の「」を参照してください。

assume-role-with-web-identity

次の例は、assume-role-with-web-identity を使用する方法を説明しています。

AWS CLI

Web Identity で認証されたロールの短期認証情報を取得するには (OAuth 2.0)

次のassume-role-with-web-identityコマンドは、IAMロールの短期認証情報のセットを取得しますapp1。リクエストは、指定されたウェブ ID プロバイダーから提供されたウェブ ID トークンを使用して認証されます。ユーザーが実行できることをさらに制限するために、2つの追加ポリシーがセッションに適用されます。返される認証情報は、生成されてから 1 時間後に失効します。

```

aws sts assume-role-with-web-identity \
  --duration-seconds 3600 \
  --role-session-name "app1" \
  --provider-id "www.amazon.com" \
  --policy-arns "arn:aws:iam::123456789012:policy/
q=webidentitydemopolicy1","arn:aws:iam::123456789012:policy/webidentitydemopolicy2"
\
  --role-arn arn:aws:iam::123456789012:role/FederatedWebIdentityRole \
  --web-identity-token "Atza
%7CIQEBljAsAhRFiXuWpUXuRvQ9PZL3GMFcYevydwIUFAHZwXZXXXXXXXXXJnruLxKDHwy87oGKPznh0D6bEQZTSCzyoC
CrKqjG7nPBjNIL016GGvuS5gSvPRUxWES3VYfm1wL7WTI7jn-Pcb6M-
buCgHhF0zTQxod27L9Cqn0Lio7N3gZAGpsp6n1-
AJB0CJckcyXe2c6uD0sr0JeZlKUm2eTDVMf8IehDVI0r1Q0nTV6KzzAI30Y87Vd_cVMQ"

```

出力:

```
{
  "SubjectFromWebIdentityToken": "amzn1.account.AF6RH07KZU5XRVQJGXXK6HB56KR2A"
  "Audience": "client.5498841531868486423.1548@apps.example.com",
  "AssumedRoleUser": {
    "Arn": "arn:aws:sts::123456789012:assumed-role/FederatedWebIdentityRole/app1",
    "AssumedRoleId": "AROACLKWSQRAOEXAMPLE:app1"
  }
  "Credentials": {
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
    "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4B1CFFxWNE1OPTgk5TthT+FvwqnKwRc0If1rRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgkBN9bkUDNCJiBeb/AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mRl/+0tkIKG07fAE",
    "Expiration": "2020-05-19T18:06:10+00:00"
  },
  "Provider": "www.amazon.com"
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」の「[一時的なセキュリティ認証情報のリクエスト](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスAssumeRoleWithWebIdentity](#)」の「」を参照してください。

assume-role

次の例は、assume-role を使用する方法を説明しています。

AWS CLI

ロールを引き受けるには

次のassume-roleコマンドは、IAMロールの短期認証情報のセットを取得しますs3-access-example。

```
aws sts assume-role \
  --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
  --role-session-name s3-access-example
```

出力:

```
{
  "AssumedRoleUser": {
    "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
    "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-example"
  },
  "Credentials": {
    "SecretAccessKey": "9drTJvcXLB89EXAMPLEELB8923FB892xMFI",
    "SessionToken": "AQoXzdELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtY1FVgAUj
+7Indz3LU0aTwk1WKIjHmMCIoTkyYp/k7kUG7moeEYKSitwQi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6D19zR0tXoybnlrZIwML1Mi1Kcgo50ytwU=",
    "Expiration": "2016-03-15T00:05:07Z",
    "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
  }
}
```

コマンドの出力には、AWSに対する認証に使用できるアクセスキー、シークレットキー、およびセッショントークンが含まれています。

AWS CLI 使用するために、ロールに関連付けられた名前付きプロファイルを設定できます。プロファイルを使用すると、AWS CLIは `assume-role` を呼び出し、認証情報を管理します。詳細については、AWS CLI「ユーザーガイド」の「[でIAMロールを使用する AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[AssumeRole](#)」の「」を参照してください。

decode-authorization-message

次のコード例は、`decode-authorization-message` を使用する方法を示しています。

AWS CLI

リクエストへの応答として返されるエンコードされた認可メッセージをデコードするには

次の `decode-authorization-message` の例では、Amazon Web Services リクエストへの応答として返されるエンコードされたメッセージから、リクエストの承認ステータスに関する追加情報をデコードします。

```
aws sts decode-authorization-message \
  --encoded-message EXAMPLEWodyRNrtLQARDip-
eTA6i6DrLUhHhPQrLWB_LAb15pAKx19mPDLexYcGBreyIKQC1BGBIpBKr3dFDkqwe07e2NMk5j_hmzAiChJN-8oy3Ewi
Ojau7BMjOTWw0tHPHv_Zaz87yENDipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDGvh1gjqZJFUiQoubv78V1RbHNYnK44E
p0u3FZjwYStfvTb3GHs3-6rLribG09jZ0tkkfE6vqx1FzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVPXioqgoqCz
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHLI-
tIiNd7tgNTYzDuPQYD2FMDbnp82V9eVmYgtPp5NIeSpuf3f0HanFuBZgENxZQZ2dLH3xJGMTtYayzZrRXjiq_SfX9zeB
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067P0FrZECLjj0tNPBOZCcuEKEXAM
```

出力:

```
{
  "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,\"matchedStatements\
\":{\n\"items\":[\n\"statementId\":\n\"VisualEditor0\", \"effect\":\n\"DENY\", \"principals\
\":{\n\"items\":[\n\"value\":\n\"AROA123456789EXAMPLE\" ]}], \"principalGroups\
\":{\n\"items\":[\n ]}], \"actions\":{\n\"items\":[\n\"value\":\n\"ec2:RunInstances\
\" ]}], \"resources\":{\n\"items\":[\n\"value\":\n\"*\" ]}], \"conditions\":{\n\"items\
\":[\n ]}], \"failures\":{\n\"items\":[\n ]}], \"context\":{\n\"principal\":{\n\"id\":\
\n\"AROA123456789EXAMPLE:Ana\", \"arn\":\n\"arn:aws:sts:111122223333:assumed-role/\
Developer/Ana\" }, \"action\":\n\"RunInstances\", \"resource\":\n\"arn:aws:ec2:us-
east-1:111122223333:instance/*\", \"conditions\":{\n\"items\":[\n\"key\":\
\n\"ec2:MetadataHttpPutResponseHopLimit\", \"values\":{\n\"items\":[\n\"value\":\
\n\"2\" ]}], {\n\"key\":\n\"ec2:InstanceMarketType\", \"values\":{\n\"items\":[\n\"value\
\":\n\"on-demand\" ]}], {\n\"key\":\n\"aws:Resource\", \"values\":{\n\"items\":[\n\"value\
\":\n\"instance/*\" ]}], {\n\"key\":\n\"aws:Account\", \"values\":{\n\"items\":[\n\"value\
\":\n\"111122223333\" ]}], {\n\"key\":\n\"ec2:AvailabilityZone\", \"values\":{\n\"items\":\
[\n\"value\":\n\"us-east-1f\" ]}], {\n\"key\":\n\"ec2:ebsOptimized\", \"values\":{\n\"items\
\":[\n\"value\":\n\"false\" ]}], {\n\"key\":\n\"ec2:IsLaunchTemplateResource\", \"values\
\":{\n\"items\":[\n\"value\":\n\"false\" ]}], {\n\"key\":\n\"ec2:InstanceType\", \"values\":\
{\n\"items\":[\n\"value\":\n\"t2.micro\" ]}], {\n\"key\":\n\"ec2:RootDeviceType\", \"values\
\":{\n\"items\":[\n\"value\":\n\"ebs\" ]}], {\n\"key\":\n\"aws:Region\", \"values\":{\n\"items\
\":[\n\"value\":\n\"us-east-1\" ]}], {\n\"key\":\n\"ec2:MetadataHttpEndpoint\", \"values\
\":{\n\"items\":[\n\"value\":\n\"enabled\" ]}], {\n\"key\":\n\"aws:Service\", \"values\":\
{\n\"items\":[\n\"value\":\n\"ec2\" ]}], {\n\"key\":\n\"ec2:InstanceID\", \"values\":{\n\"items\
\":[\n\"value\":\n\"*\" ]}], {\n\"key\":\n\"ec2:MetadataHttpTokens\", \"values\":{\n\"items\
\":[\n\"value\":\n\"required\" ]}], {\n\"key\":\n\"aws:Type\", \"values\":{\n\"items\":\
[\n\"value\":\n\"instance\" ]}], {\n\"key\":\n\"ec2:Tenancy\", \"values\":{\n\"items\":\
[\n\"value\":\n\"default\" ]}], {\n\"key\":\n\"ec2:Region\", \"values\":{\n\"items\":[\n\"value\
\":\n\"us-east-1\" ]}], {\n\"key\":\n\"aws:ARN\", \"values\":{\n\"items\":[\n\"value\":\
\n\"arn:aws:ec2:us-east-1:111122223333:instance/*\" ]}}}}}}"}
}
```

詳細については、「ユーザーガイド」の[「ポリシー評価ロジック」](#)を参照してください。AWS IAM

- API 詳細については、AWS CLI 「コマンドリファレンス[DecodeAuthorizationMessage](#)」の「」を参照してください。

get-caller-identity

次の例は、get-caller-identity を使用する方法を説明しています。

AWS CLI

現在の IAM ID の詳細を取得するには

次のget-caller-identityコマンドは、リクエストの認証に使用される IAM ID に関する情報を表示します。発信者はIAMユーザーです。

```
aws sts get-caller-identity
```

出力:

```
{
  "UserId": "AIDASAMPLEUSERID",
  "Account": "123456789012",
  "Arn": "arn:aws:iam::123456789012:user/DevAdmin"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCallerIdentity](#)」の「」を参照してください。

get-federation-token

次の例は、get-federation-token を使用する方法を説明しています。

AWS CLI

IAM ユーザーアクセスキー認証情報を使用して一時的なセキュリティ認証情報のセットを返すには

次の get-federation-token の例では、ユーザーの一時的なセキュリティ認証情報一式 (アクセスキー ID、シークレットアクセスキー、セキュリティトークンで構成) を返します。IAM ユー

ザーの長期的なセキュリティ認証情報を使用して GetFederationToken オペレーションを呼び出す必要があります。

```
aws sts get-federation-token \  
  --name Bob \  
  --policy file://myfile.json \  
  --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \  
  --duration-seconds 900
```

myfile.json の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "ec2:Describe*",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "elasticloadbalancing:Describe*",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "cloudwatch:ListMetrics",  
        "cloudwatch:GetMetricStatistics",  
        "cloudwatch:Describe*"  
      ],  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "autoscaling:Describe*",  
      "Resource": "*"  
    }  
  ]  
}
```

出力:

```
{
  "Credentials": {
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "SessionToken": "EXAMPLEpZ21uX2VjEGoaCXVzLXd1c3QtMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42QqunWMTfKq0DCOP////////
wEQAxoMNDUy0TI1MTcwNTA3Igxuy3A0puuoLsk3MJwqgQPg8Q0d9HuoClUxq26wnc/nm
+eZLjHDyGf2KUAHK2DuaS/nrGSEXAMPLE",
    "Expiration": "2023-12-20T02:06:07+00:00"
  },
  "FederatedUser": {
    "FederatedUserId": "111122223333:Bob",
    "Arn": "arn:aws:sts::111122223333:federated-user/Bob"
  },
  "PackedPolicySize": 36
}
```

詳細については、AWS IAM「[ユーザーガイド](#)」の「[一時的なセキュリティ認証情報のリクエスト](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスGetFederationToken](#)」の「」を参照してください。

get-session-token

次の例は、get-session-token を使用する方法を説明しています。

AWS CLI

IAM ID の一連の短期認証情報を取得するには

次のget-session-tokenコマンドは、呼び出しを行う IAM ID の短期認証情報のセットを取得します。生成された認証情報は、ポリシーで多要素認証 (MFA) が必要なリクエストに使用できません。認証情報は生成されてから 15 分後に失効します。

```
aws sts get-session-token \
  --duration-seconds 900 \
  --serial-number "YourMFADeviceSerialNumber" \
  --token-code 123456
```

出力:

```
{
  "Credentials": {
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYzEXAMPLEKEY",
    "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4B1CFFxWNE1OPTgk5TthT
+FvwqnKwRc0IfrrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mRl/+0tkIKG07fAE",
    "Expiration": "2020-05-19T18:06:10+00:00"
  }
}
```

詳細については、AWS IAM 「ユーザーガイド」の「[一時的なセキュリティ認証情報のリクエスト](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSessionToken](#)」の「」を参照してください。

AWS Support を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS Support。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-attachments-to-set

次のコード例は、add-attachments-to-set を使用する方法を示しています。

AWS CLI

セットにアタッチメントを追加するには

次のadd-attachments-to-set例では、AWS アカウントでサポートケースに指定できるイメージをセットに追加します。

```
aws support add-attachments-to-set \  
  --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE" \  
  --attachments fileName=troubleshoot-screenshot.png,data=base64-encoded-string
```

出力:

```
{  
  "attachmentSetId": "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE",  
  "expiryTime": "2020-05-14T17:04:40.790+0000"  
}
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddAttachmentsToSet](#)」の「」を参照してください。

add-communication-to-case

次のコード例は、add-communication-to-case を使用する方法を示しています。

AWS CLI

ケースに通信を追加するには

次のadd-communication-to-case例では、AWS アカウントのサポートケースに通信を追加します。

```
aws support add-communication-to-case \  
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \  
  --communication-body "I'm attaching a set of images to this case." \  
  --cc-email-addresses "myemail@example.com" \  
  --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE"
```

出力:

```
{
  "result": true
}
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddCommunicationToCase](#)」の「」を参照してください。

create-case

次の例は、create-case を使用する方法を説明しています。

AWS CLI

ケースを作成する

次のcreate-case例では、AWS アカウントのサポートケースを作成します。

```
aws support create-case \
  --category-code "using-aws" \
  --cc-email-addresses "myemail@example.com" \
  --communication-body "I want to learn more about an AWS service." \
  --issue-type "technical" \
  --language "en" \
  --service-code "general-info" \
  --severity-code "low" \
  --subject "Question about my account"
```

出力:

```
{
  "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47"
}
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCase](#)」の「」を参照してください。

describe-attachment

次のコード例は、describe-attachment を使用する方法を示しています。

AWS CLI

添付ファイルについて説明する

次の describe-attachment の例では、指定された ID の添付ファイルに関する情報を返します。

```
aws support describe-attachment \  
  --attachment-id "attachment-KBnjRNrePd9D6Jx0-Mm00xZuDEaL2JAj_0-  
gJv9qqDooTipsz3V1Nb19rCfkZneeQeDPgp8X1iVJyHH7UuhZDdNeqGoduZsPrAhyMakq1c60-  
iJjL5HqyYGiT1FG8EXAMPLE"
```

出力:

```
{  
  "attachment": {  
    "fileName": "troubleshoot-screenshot.png",  
    "data": "base64-blob"  
  }  
}
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAttachment](#)」の「」を参照してください。

describe-cases

次の例は、describe-cases を使用する方法を説明しています。

AWS CLI

ケースについて説明する

次の describe-cases 例では、AWS アカウントで指定されたサポートケースに関する情報を返します。

```
aws support describe-cases \  
  --cases
```

```
--display-id "1234567890" \  
--after-time "2020-03-23T21:31:47.774Z" \  
--include-resolved-cases \  
--language "en" \  
--no-include-communications \  
--max-item 1
```

出力:

```
{  
  "cases": [  
    {  
      "status": "resolved",  
      "ccEmailAddresses": [],  
      "timeCreated": "2020-03-23T21:31:47.774Z",  
      "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",  
      "severityCode": "low",  
      "language": "en",  
      "categoryCode": "using-aws",  
      "serviceCode": "general-info",  
      "submittedBy": "myemail@example.com",  
      "displayId": "1234567890",  
      "subject": "Question about my account"  
    }  
  ]  
}
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCases](#)」の「」を参照してください。

describe-communications

次の例は、describe-communications を使用する方法を説明しています。

AWS CLI

ケースの最新のコミュニケーションについて説明する

次のdescribe-communications例では、AWS アカウントで指定されたサポートケースの最新通信を返します。

```
aws support describe-communications \  
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \  
  --after-time "2020-03-23T21:31:47.774Z" \  
  --max-item 1
```

出力:

```
{  
  "communications": [  
    {  
      "body": "I want to learn more about an AWS service.",  
      "attachmentSet": [],  
      "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",  
      "timeCreated": "2020-05-12T23:12:35.000Z",  
      "submittedBy": "Amazon Web Services"  
    }  
  ],  
  "NextToken": "eyJmZXRlc291dGVzZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQEXAMPLE=="  
}
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeCommunications](#)」の「」を参照してください。

describe-services

次のコード例は、describe-services を使用する方法を示しています。

AWS CLI

AWS サービスとサービスカテゴリを一覧表示するには

次の describe-services の例では、一般的な情報をリクエストするためのサービスカテゴリを一覧表示します。

```
aws support describe-services \  
  --service-code-list "general-info"
```

出力:

```
{
```

```
"services": [
  {
    "code": "general-info",
    "name": "General Info and Getting Started",
    "categories": [
      {
        "code": "charges",
        "name": "How Will I Be Charged?"
      },
      {
        "code": "gdpr-queries",
        "name": "Data Privacy Query"
      },
      {
        "code": "reserved-instances",
        "name": "Reserved Instances"
      },
      {
        "code": "resource",
        "name": "Where is my Resource?"
      },
      {
        "code": "using-aws",
        "name": "Using AWS & Services"
      },
      {
        "code": "free-tier",
        "name": "Free Tier"
      },
      {
        "code": "security-and-compliance",
        "name": "Security & Compliance"
      },
      {
        "code": "account-structure",
        "name": "Account Structure"
      }
    ]
  }
]
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeServices](#)」の「」を参照してください。

describe-severity-levels

次のコード例は、describe-severity-levels を使用する方法を示しています。

AWS CLI

利用可能な重要度レベルを一覧表示する

次の describe-severity-levels の例では、サポートケースの重要度レベルを一覧表示します。

```
aws support describe-severity-levels
```

出力:

```
{
  "severityLevels": [
    {
      "code": "low",
      "name": "Low"
    },
    {
      "code": "normal",
      "name": "Normal"
    },
    {
      "code": "high",
      "name": "High"
    },
    {
      "code": "urgent",
      "name": "Urgent"
    },
    {
      "code": "critical",
      "name": "Critical"
    }
  ]
}
```

詳細については、「AWS サポートユーザーガイド」の「[緊急度の選択](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeSeverityLevels](#)」の「」を参照してください。

describe-trusted-advisor-check-refresh-statuses

次のコード例は、describe-trusted-advisor-check-refresh-statuses を使用する方法を示しています。

AWS CLI

AWS Trusted Advisor チェックの更新ステータスを一覧表示するには

次のdescribe-trusted-advisor-check-refresh-statuses例では、Amazon S3 バケットのアクセス許可とIAM使用という 2 つの Trusted Advisor チェックの更新ステータスを一覧表示します。

```
aws support describe-trusted-advisor-check-refresh-statuses \  
--check-id "Pfx0RwqBli" "zXCkfM1nI3"
```

出力:

```
{  
  "statuses": [  
    {  
      "checkId": "Pfx0RwqBli",  
      "status": "none",  
      "millisUntilNextRefreshable": 0  
    },  
    {  
      "checkId": "zXCkfM1nI3",  
      "status": "none",  
      "millisUntilNextRefreshable": 0  
    }  
  ]  
}
```

詳細については、AWS サポートユーザーガイドの[AWS Trusted Advisor](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrustedAdvisorCheckRefreshStatuses](#)」の「」を参照してください。

describe-trusted-advisor-check-result

次の例は、describe-trusted-advisor-check-result を使用する方法を説明しています。

AWS CLI

AWS Trusted Advisor チェックの結果を一覧表示するには

次のdescribe-trusted-advisor-check-result例では、IAM使用チェックの結果を一覧表示します。

```
aws support describe-trusted-advisor-check-result \
  --check-id "zXCkFM1nI3"
```

出力:

```
{
  "result": {
    "checkId": "zXCkFM1nI3",
    "timestamp": "2020-05-13T21:38:05Z",
    "status": "ok",
    "resourcesSummary": {
      "resourcesProcessed": 1,
      "resourcesFlagged": 0,
      "resourcesIgnored": 0,
      "resourcesSuppressed": 0
    },
    "categorySpecificSummary": {
      "costOptimizing": {
        "estimatedMonthlySavings": 0.0,
        "estimatedPercentMonthlySavings": 0.0
      }
    },
    "flaggedResources": [
      {
        "status": "ok",
        "resourceId": "47DEQpj8HBSa-_TImW-5JCeuQeRkm5NMpJWZEXAMPLE",
        "isSuppressed": false
      }
    ]
  }
}
```

詳細については、AWS サポートユーザーガイドの[AWS Trusted Advisor](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrustedAdvisorCheckResult](#)」の「」を参照してください。

describe-trusted-advisor-check-summaries

次の例は、describe-trusted-advisor-check-summaries を使用する方法を説明しています。

AWS CLI

AWS Trusted Advisor チェックの概要を一覧表示するには

次のdescribe-trusted-advisor-check-summaries例では、Amazon S3 バケットのアクセス許可とIAM使用という 2 つの Trusted Advisor チェックの結果を一覧表示します。

```
aws support describe-trusted-advisor-check-summaries \
  --check-ids "Pfx0RwqBli" "zXCkFM1nI3"
```

出力:

```
{
  "summaries": [
    {
      "checkId": "Pfx0RwqBli",
      "timestamp": "2020-05-13T21:38:12Z",
      "status": "ok",
      "hasFlaggedResources": true,
      "resourcesSummary": {
        "resourcesProcessed": 44,
        "resourcesFlagged": 0,
        "resourcesIgnored": 0,
        "resourcesSuppressed": 0
      },
      "categorySpecificSummary": {
        "costOptimizing": {
          "estimatedMonthlySavings": 0.0,
          "estimatedPercentMonthlySavings": 0.0
        }
      }
    }
  ],
}
```

```
{
  "checkId": "zXCkfM1nI3",
  "timestamp": "2020-05-13T21:38:05Z",
  "status": "ok",
  "hasFlaggedResources": true,
  "resourcesSummary": {
    "resourcesProcessed": 1,
    "resourcesFlagged": 0,
    "resourcesIgnored": 0,
    "resourcesSuppressed": 0
  },
  "categorySpecificSummary": {
    "costOptimizing": {
      "estimatedMonthlySavings": 0.0,
      "estimatedPercentMonthlySavings": 0.0
    }
  }
}
]
```

詳細については、AWS サポートユーザーガイドの[AWS Trusted Advisor](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrustedAdvisorCheckSummaries](#)」の「」を参照してください。

describe-trusted-advisor-checks

次のコード例は、describe-trusted-advisor-checks を使用する方法を示しています。

AWS CLI

利用可能な AWS Trusted Advisor チェックを一覧表示するには

次のdescribe-trusted-advisor-checks例では、AWS アカウントで利用可能な Trusted Advisor チェックを一覧表示します。この情報には、チェック名、ID、説明、カテゴリ、メタデータが含まれます。読みやすくするために、出力が短縮されることに注意してください。

```
aws support describe-trusted-advisor-checks \
  --language "en"
```

出力:

```
{
  "checks": [
    {
      "id": "zXCkfM1nI3",
      "name": "IAM Use",
      "description": "Checks for your use of AWS Identity and Access Management (IAM). You can use IAM to create users, groups, and roles in AWS, and you can use permissions to control access to AWS resources. \n<br>\n<br>\n<b>Alert Criteria</b><br>\nYellow: No IAM users have been created for this account.\n<br>\n<br>\n<b>Recommended Action</b><br>\nCreate one or more IAM users and groups in your account. You can then create additional users whose permissions are limited to perform specific tasks in your AWS environment. For more information, see <a href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAMGettingStarted.html\" target=\"_blank\">Getting Started</a>. \n<br><br>\n<b>Additional Resources</b><br>\n<a href=\"https://docs.aws.amazon.com/IAM/latest/UserGuide/IAM_Introduction.html\" target=\"_blank\">What Is IAM?</a>",
      "category": "security",
      "metadata": []
    }
  ]
}
```

詳細については、AWS サポートユーザーガイドの[AWS Trusted Advisor](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTrustedAdvisorChecks](#)」の「」を参照してください。

refresh-trusted-advisor-check

次のコード例は、refresh-trusted-advisor-check を使用する方法を示しています。

AWS CLI

AWS Trusted Advisor チェックを更新するには

次のrefresh-trusted-advisor-check例では、AWS アカウントの Amazon S3 Bucket Permissions Trusted Advisor チェックを更新します。

```
aws support refresh-trusted-advisor-check \
  --check-id "Pfx0RwqBLi"
```

出力:

```
{
  "status": {
    "checkId": "Pfx0RwqBli",
    "status": "enqueued",
    "millisUntilNextRefreshable": 3599992
  }
}
```

詳細については、AWS 「サポートユーザーガイド」の[AWS Trusted Advisor](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RefreshTrustedAdvisorCheck](#)」の「」を参照してください。

resolve-case

次のコード例は、resolve-case を使用する方法を示しています。

AWS CLI

サポートケースを解決する

次のresolve-case例では、AWS アカウントのサポートケースを解決します。

```
aws support resolve-case \
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47"
```

出力:

```
{
  "finalCaseStatus": "resolved",
  "initialCaseStatus": "work-in-progress"
}
```

詳細については、「AWS サポートユーザーガイド」の「[ケース管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResolveCase](#)」の「」を参照してください。

を使用した Amazon SWF の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示していますSWF。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

count-closed-workflow-executions

次の例は、count-closed-workflow-executions を使用する方法を説明しています。

AWS CLI

クローズドワークフロー実行のカウント

swf count-closed-workflow-executions を使用して、特定のドメインのクローズドワークフロー実行数を取得できます。フィルターを指定して、特定の実行クラスをカウントできます。

--domain および --close-time-filterまたは 引--start-time-filter数は必須です。その他の引数はオプションです。

```
aws swf count-closed-workflow-executions \  
  --domain DataFrobtzz \  
  --close-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :  
  1370044800 }"
```

出力:

```
{
```

```
"count": 2,  
"truncated": false  
}
```

「truncated」が の場合true、「count」は Amazon が返すことができる最大数を表します SWF。その他の結果は切り捨てられます。

返される結果の数を減らすには、次の操作を行います。

--close-time-filter または --start-time-filter 値を変更して、検索する時間範囲を絞り込みます。これらはそれぞれ相互に排他的です。リクエストで指定できるのはこれらのうちの 1 つだけです。--close-status-filter、--execution-filter、--tag-filter または --type-filter 引数を使用して結果をさらにフィルタリングします。ただし、これらの引数も相互に排他的です。

Amazon Simple Workflow Service API リファレンスの も [CountClosedWorkflowExecutions](#) 参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CountClosedWorkflowExecutions](#)」の「」を参照してください。

count-open-workflow-executions

次の例は、count-open-workflow-executions を使用する方法を説明しています。

AWS CLI

オープンワークフロー実行のカウント

swf count-open-workflow-executions を使用して、特定のドメインのオープンワークフロー実行数を取得できます。フィルターを指定して、特定の実行クラスをカウントできます。

引数--domainと引--start-time-filter数は必須です。その他の引数はオプションです。

```
aws swf count-open-workflow-executions \  
  --domain DataFrobtzz \  
  --start-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :  
  1370044800 }"
```

出力:

```
{
```

```
"count": 4,  
"truncated": false  
}
```

「truncated」が `true` の場合、`count` は Amazon が返すことができる最大数を表します。その他の結果は切り捨てられます。

返される結果の数を減らすには、次の操作を行います。

`--start-time-filter` 値を変更して、検索する時間範囲を絞り込みます。`--close-status-filter`、`--execution-filter`、`--tag-filter` または `--type-filter` 引数を使用して、結果をさらにフィルタリングします。これらはそれぞれ相互に排他的です。リクエストで指定できるのは、いずれか 1 つだけです。

詳細については、「Amazon Simple Workflow Service API リファレンス `CountOpenWorkflowExecutions`」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CountOpenWorkflowExecutions](#)」の「」を参照してください。

deprecate-domain

次の例は、`deprecate-domain` を使用する方法を説明しています。

AWS CLI

ドメインの非推奨

ドメインを廃止するには、`swf deprecate-domain` を使用します(まだ表示できますが、新しいワークフロー実行の作成、またはタイプを登録することはできません)。これには、唯一の必須パラメータ `--name` があり、廃止するドメイン名を取得します。

```
aws swf deprecate-domain \  
  --name MyNeatNewDomain ""
```

`register-domain` と同様に、出力は返されません。ただし、`list-domains` を使用して登録済みドメインを表示すると、ドメインが非推奨になり、返されたデータに表示されなくなります。

```
aws swf list-domains \  
  --registration-status REGISTERED
```



```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "DataFrobotz"
    },
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

`--registration-status DEPRECATED` を使用すると `list-domains`、廃止されたドメインが表示されます。

```
aws swf list-domains \
  --registration-status DEPRECATED
{
  "domainInfos": [
    {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    }
  ]
}
```

`describe-domain` を使用して、非推奨ドメインに関する情報を取得することもできます。

```
aws swf describe-domain \
  --name MyNeatNewDomain
{
  "domainInfo": {
    "status": "DEPRECATED",
    "name": "MyNeatNewDomain"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "0"
  }
}
```

Amazon Simple Workflow Service APIリファレンスの [DeprecateDomain](#) 参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeprecateDomain](#)」の「」を参照してください。

describe-domain

次の例は、describe-domain を使用する方法を説明しています。

AWS CLI

ドメインに関する情報の取得

特定のドメインに関する詳細情報を取得するには、swf describe-domain コマンドを使用します。--name という必須パラメータが 1 つあり、これは情報が必要なドメインの名前を取得します。

```
aws swf describe-domain \  
  --name DataFrobotz \  
  {  
    "domainInfo": {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    },  
    "configuration": {  
      "workflowExecutionRetentionPeriodInDays": "1"  
    }  
  }
```

describe-domain を使用して、廃止されたドメインに関する情報を取得することもできます。

```
aws swf describe-domain \  
  --name MyNeatNewDomain \  
  {  
    "domainInfo": {  
      "status": "DEPRECATED",  
      "name": "MyNeatNewDomain"  
    },  
    "configuration": {  
      "workflowExecutionRetentionPeriodInDays": "0"  
    }  
  }
```

Amazon Simple Workflow Service APIリファレンスの [DescribeDomain](#)参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDomain](#)」の「」を参照してください。

list-activity-types

次のコード例は、list-activity-types を使用する方法を示しています。

AWS CLI

アクティビティタイプの一覧表示

ドメインのアクティビティタイプのリストを取得するには、を使用しますswf list-activity-types。引数--domainと引--registration-status数は必須です。

```
aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED
```

出力:

```
{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.451,
      "activityType": {
        "version": "1",
        "name": "confirm-user-email"
      },
      "description": "subscribe confirm-user-email activity"
    },
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.709,
      "activityType": {
        "version": "1",
        "name": "confirm-user-phone"
      },
      "description": "subscribe confirm-user-phone activity"
    },
    {
      "status": "REGISTERED",
```

```

    "creationDate": 1371454149.871,
    "activityType": {
      "version": "1",
      "name": "get-subscription-info"
    },
    "description": "subscribe get-subscription-info activity"
  },
  {
    "status": "REGISTERED",
    "creationDate": 1371454150.909,
    "activityType": {
      "version": "1",
      "name": "send-subscription-success"
    },
    "description": "subscribe send-subscription-success activity"
  },
  {
    "status": "REGISTERED",
    "creationDate": 1371454150.085,
    "activityType": {
      "version": "1",
      "name": "subscribe-user-sns"
    },
    "description": "subscribe subscribe-user-sns activity"
  }
]
}

```

--name 引数を使用して、特定の名前のアクティビティタイプのみを選択できます。

```

aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --name "send-subscription-success"

```

出力:

```

{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.909,
      "activityType": {

```

```

        "version": "1",
        "name": "send-subscription-success"
    },
    "description": "subscribe send-subscription-success activity"
}
]
}

```

ページで結果を取得するには、`--maximum-page-size` 引数を設定できます。結果のページに収まるよりも多くの結果が返された場合、`nextPageToken` 「」が結果セットに返されます。

```

aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --maximum-page-size 2

```

出力:

```

{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLDL17QNY7C30pHo9Sz06D/GuFz10yC73umBQ1t0PJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX40C/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarki qpSY1ZVveBasBv1vyU
WGAaqehiDz7/JzLT/wWNNUM0d+Nhe",
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.451,
      "activityType": {
        "version": "1",
        "name": "confirm-user-email"
      },
      "description": "subscribe confirm-user-email activity"
    },
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.709,
      "activityType": {
        "version": "1",
        "name": "confirm-user-phone"
      },
      "description": "subscribe confirm-user-phone activity"
    }
  ]
}

```

```
]
}
```

--next-page-token 引数の list-activity-types への次の呼び出しに nextPageToken 値を渡すと、結果の次のページを取得できます。

```
aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --maximum-page-size 2 \
  --next-page-token "AAAAKgAAAAEAAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCclgCuq3J+h/
+PmHvAnDxJYbup8+0R4LVtbXLDL7QNY7C30pHo9Sz06D/GuFz10yC73umBQ1t0PJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX40C/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMLtarki qpSY1ZVveBasBvLvU
WGAaqehiDz7/JzLT/wWNUMOd+Nhe"
```

出力:

```
{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCclgCuq3J+h/
m3+v0fFqtkcjLwV5cc40jNAzTCuq/
XcylPumGwkjbajtqpZpbq0cVnFjFxGoi0LB20lbvv0krbUISBvlpFPmSWpDSZJsxg5UxCcweteS1Fn1PNSZ/
MoinBZo80TkjMuzcsTuK0zH9wCaR8ITcALJ3SaqHU3pyIRS5hPmFA30LIc8zaAepjlaujo6hntNSCruB4"
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454149.871,
      "activityType": {
        "version": "1",
        "name": "get-subscription-info"
      },
      "description": "subscribe get-subscription-info activity"
    },
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.909,
      "activityType": {
        "version": "1",
        "name": "send-subscription-success"
      },
      "description": "subscribe send-subscription-success activity"
    }
  ]
}
```

```
}
```

返される結果がまだある場合は、nextPageToken「」が結果とともに返されます。返す結果のページがこれ以上ない場合、nextPageToken「」は結果セットには返されません。

--reverse-order 引数を使用して、返された結果の順序を逆にすることができます。これはページ分割された結果にも影響します。

```
aws swf list-activity-types \  
  --domain DataFrobtzz \  
  --registration-status REGISTERED \  
  --maximum-page-size 2 \  
  --reverse-order
```

出力:

```
{  
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAwXcpu5ePSyQkrC  
+8WMbmSrenuZC2ZkIXQYBPB/b9xIOVkj+bMEFhGj0KmmJ4rF7iddhjf7UMYCSfGkEn7mk  
+yMCgVc1JxDWmB0EH46bhcmLmYNQihMDmUWocpr7To6/R7CLu0St1gkFayx0idJXErQW0zdNfQaIWAnF/  
cwioBbXlkz1fQzmDeU3M5oYGMPQIrUqkPq7pMEW0q0lK5eDN97NzFYdZZ/rlcLDWPZhUjY",  
  "typeInfos": [  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.085,  
      "activityType": {  
        "version": "1",  
        "name": "subscribe-user-sns"  
      },  
      "description": "subscribe subscribe-user-sns activity"  
    },  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.909,  
      "activityType": {  
        "version": "1",  
        "name": "send-subscription-success"  
      },  
      "description": "subscribe send-subscription-success activity"  
    }  
  ]  
}
```

Amazon Simple Workflow Service APIリファレンスの [もListActivityTypes](#)参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListActivityTypes](#)」の「」を参照してください。

list-domains

次の例は、list-domains を使用する方法を説明しています。

AWS CLI

例 1: 登録済みドメインを一覧表示するには

次のlist-domainsコマンド例では、アカウントに登録したREGISTEREDSWFドメインを一覧表示します。

```
aws swf list-domains \  
  --registration-status REGISTERED
```

出力:

```
{  
  "domainInfos": [  
    {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    },  
    {  
      "status": "REGISTERED",  
      "name": "erontest"  
    }  
  ]  
}
```

詳細については、「Amazon Simple Workflow Service APIリファレンス[ListDomains](#)」の「」を参照してください。

例 2: 非推奨ドメインを一覧表示するには

次のlist-domainsコマンド例は、アカウントに登録したDEPRECATEDSWFドメインを一覧表示します。非推奨ドメインは、新しいワークフローやアクティビティを登録することはできませんが、クエリすることはできます。


```
aws swf list-domains \  
  --registration-status DEPRECATED
```

出力:

```
{  
  "domainInfos": [  
    {  
      "status": "DEPRECATED",  
      "name": "MyNeatNewDomain"  
    }  
  ]  
}
```

詳細については、「Amazon Simple Workflow Service APIリファレンス[ListDomains](#)」の「」を参照してください。

例 3: 登録済みドメインの最初のページを一覧表示するには

次のlist-domainsコマンド例では、--maximum-page-sizeオプションを使用してアカウントに登録した最初のページREGISTEREDSWFドメインを一覧表示します。

```
aws swf list-domains \  
  --registration-status REGISTERED \  
  --maximum-page-size 1
```

出力:

```
{  
  "domainInfos": [  
    {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    }  
  ],  
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD  
+QKT2ynuEbibcQWe2QKrs1MGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it  
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKE0Pm3k2r40jAGaFsG0uVbrK1jv1a7wdU7FYH301kNCP8b7PBj9SBkUyGoiAg  
}
```

詳細については、「Amazon Simple Workflow Service APIリファレンス[ListDomains](#)」の「」を参照してください。

例 4: 登録済みドメインの指定された 1 ページを一覧表示するには

次のlist-domainsコマンド例では、--maximum-page-sizeオプションを使用してアカウントに登録した最初のページREGISTEREDSWFドメインを一覧表示します。

再度呼び出しを行うと、引--next-page-token数nextPageTokenに の値を指定すると、結果の別のページが表示されます。

```
aws swf list-domains \
  --registration-status REGISTERED \
  --maximum-page-size 1 \
  --next-page-token "AAAAKgAAAAEAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrsLMGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r40jAGaFsG0uVbrKlJvLa7wdU7FYH30lkNCP8b7PBj9SBkUyGoiAg
```

出力:

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

取得する結果のページがそれ以上存在しない場合、nextPageToken が、結果に返されることはありません。

詳細については、「Amazon Simple Workflow Service APIリファレンス[ListDomains](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDomains](#)」の「」を参照してください。

list-workflow-types

次の例は、list-workflow-types を使用する方法を説明しています。

AWS CLI

ワークフロータイプの一覧表示

ドメインのワークフロータイプのリストを取得するには、`aws swf list-workflow-types` を使用します。引数 `--domain` と引数 `--registration-status` は必須です。以下に簡単な例を示します。

```
aws swf list-workflow-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED
```

出力:

```
{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454149.598,
      "description": "DataFrobtzz subscribe workflow",
      "workflowType": {
        "version": "v3",
        "name": "subscribe"
      }
    }
  ]
}
```

と同様に `list-activity-types`、`--name` 引数を使用して特定の名称のワークフロータイプのみを選択し、`--maximum-page-size` と連携して `--next-page-token` を使用して結果をページングできます。結果が返される順序を逆にするには、`--reverse-order` を使用します。

Amazon Simple Workflow Service API リファレンスの [ListWorkflowTypes](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListWorkflowTypes](#)」の「」を参照してください。

register-domain

次の例は、`register-domain` を使用する方法を説明しています。

AWS CLI

ドメインの登録

を使用して AWS CLI 新しいドメインを登録できます。swf register-domain コマンドを実行します。ドメイン名 --name を取得すると、このドメインのワークフロー実行データを保持する日数を指定するには整数 --workflow-execution-retention-period-in-days の 2 つの必須パラメータがあり、最大期間は 90 日間です (詳細については、SWFFAQ 「<https://aws.amazon.com/swf/faqs/#retain_limit>」を参照してください)。ワークフロー実行データは、指定された日数が経過した後は保持されません。

```
aws swf register-domain \  
  --name MyNeatNewDomain \  
  --workflow-execution-retention-period-in-days 0  
  ""
```

ドメインを登録すると何も返されませんが (""), swf list-domains または swf describe-domain を使用して、新しいドメインを表示できます。

```
aws swf list-domains \  
  --registration-status REGISTERED  
  {  
    "domainInfos": [  
      {  
        "status": "REGISTERED",  
        "name": "DataFrobotz"  
      },  
      {  
        "status": "REGISTERED",  
        "name": "MyNeatNewDomain"  
      },  
      {  
        "status": "REGISTERED",  
        "name": "erontest"  
      }  
    ]  
  }
```

swf describe-domain を使用する:

```
aws swf describe-domain --
name MyNeatNewDomain
{
  "domainInfo": {
    "status": "REGISTERED",
    "name": "MyNeatNewDomain"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "0"
  }
}
```

Amazon Simple Workflow Service APIリファレンスの も[RegisterDomain](#)参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterDomain](#)」の「」を参照してください。

register-workflow-type

次の例は、register-workflow-type を使用する方法を説明しています。

AWS CLI

ワークフロータイプの登録

ワークフロータイプを に登録するには AWS CLI、swf register-workflow-type コマンドを使用します。

```
aws swf register-workflow-type \
  --domain DataFrobtzz \
  --name "MySimpleWorkflow" \
  --workflow-version "v1"
```

成功すると、コマンドは出力を生成しません。

エラーの場合 (例えば、同じワークフローを 2 回登録しようとしたり、存在しないドメインを指定したりした場合)、レスポンスが表示されます JSON。

```
{
  "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",
  "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"
```

```
}
```

--domain、 --name --workflow-versionは必須です。ワークフローの説明、タイムアウト、子ワークフローポリシーを設定することもできます。

詳細については、「Amazon Simple Workflow Service APIリファレンス[RegisterWorkflowType](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterWorkflowType](#)」の「」を参照してください。

を使用した Systems Manager の例 AWS CLI

次のコード例は、Systems Manager AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

add-tags-to-resource

次の例は、add-tags-to-resource を使用する方法を説明しています。

AWS CLI

例 1: メンテナンスウィンドウにタグを追加するには

次の add-tags-to-resource の例では、指定されたメンテナンスウィンドウにタグを追加します。

```
aws ssm add-tags-to-resource \
```

```
--resource-type "MaintenanceWindow" \  
--resource-id "mw-03eb9db428EXAMPLE" \  
--tags "Key=Stack,Value=Production"
```

このコマンドでは何も出力されません。

例 2: パラメータにタグを追加するには

次の `add-tags-to-resource` の例では、指定されたパラメータに 2 つのタグを追加します。

```
aws ssm add-tags-to-resource \  
  --resource-type "Parameter" \  
  --resource-id "My-Parameter" \  
  --tags '[{"Key":"Region","Value":"East"}, {"Key":"Environment",  
"Value":"Production"}]'
```

このコマンドでは何も出力されません。

例 3: SSMドキュメントにタグを追加するには

次の `add-tags-to-resource` の例では、指定されたドキュメントにタグを追加します。

```
aws ssm add-tags-to-resource \  
  --resource-type "Document" \  
  --resource-id "My-Document" \  
  --tags "Key=Quarter,Value=Q322"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Systems Manager リソースにタグを付ける](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[AddTagsToResource](#)」の「」を参照してください。

associate-ops-item-related-item

次の例は、`associate-ops-item-related-item` を使用する方法を説明しています。

AWS CLI

関連項目を関連付けるには

次のassociate-ops-item-related-item例では、関連項目を `に` 関連付けます OpsItem。

```
aws ssm associate-ops-item-related-item \  
  --ops-item-id "oi-649fExample" \  
  --association-type "RelatesTo" \  
  --resource-type "AWS::SSMIncidents::IncidentRecord" \  
  --resource-uri "arn:aws:ssm-incidents::111122223333:incident-record/Example-  
Response-Plan/c2bde883-f7d5-343a-b13a-bf5fe9ea689f"
```

出力:

```
{  
  "AssociationId": "61d7178d-a30d-4bc5-9b4e-a9e74EXAMPLE"  
}
```

詳細については、AWS 「Systems [Manager ユーザーガイド](#)」の「[Incident Manager インシデントの使用 OpsCenter](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスAssociateOpsItemRelatedItem](#)」の「」を参照してください。

cancel-command

次の例は、cancel-command を使用する方法を説明しています。

AWS CLI

例 1: すべてのインスタンスのコマンドをキャンセルするには

次の cancel-command の例では、すべてのインスタンスで既に実行されている、指定されたコマンドのキャンセルを試みます。

```
aws ssm cancel-command \  
  --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

このコマンドでは何も出力されません。

例 2: 特定のインスタンスのコマンドをキャンセルするには

次の cancel-command の例では、指定されたインスタンスに対してのみ実行するコマンドのキャンセルを試みます。


```
aws ssm cancel-command \  
  --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE" \  
  --instance-ids "i-02573cafcfEXAMPLE"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Systems Manager パラメータにタグをつける](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelCommand](#)」の「」を参照してください。

cancel-maintenance-window-execution

次の例は、cancel-maintenance-window-execution を使用する方法を説明しています。

AWS CLI

メンテナンスウィンドウの実行をキャンセルするには

この例では、既に進行中の指定されたメンテナンスウィンドウの実行をcancel-maintenance-window-execution停止します。

```
aws ssm cancel-maintenance-window-execution \  
  --window-execution-id j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE
```

出力:

```
{  
  "WindowExecutionId": "j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"  
}
```

詳細については、「[Systems Manager ユーザーガイド](#)」の「[Systems Manager メンテナンス Windows チュートリアル \(AWS CLI\)](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CancelMaintenanceWindowExecution](#)」の「」を参照してください。

create-activation

次のコード例は、create-activation を使用する方法を示しています。

AWS CLI

マネージドインスタンスのアクティベーションを作成するには

次の `create-activation` の例では、マネージドインスタンスのアクティベーションを作成します。

```
aws ssm create-activation \  
  --default-instance-name "HybridWebServers" \  
  --iam-role "HybridWebServersRole" \  
  --registration-limit 5
```

出力:

```
{  
  "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",  
  "ActivationCode": "dRmgnYaFv567vEXAMPLE"  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Step 4: Create a Managed-Instance Activation for a Hybrid Environment](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateActivation](#)」の「」を参照してください。

`create-association-batch`

次のコード例は、`create-association-batch` を使用する方法を示しています。

AWS CLI

複数の関連付けを削除するには

この例では、設定ドキュメントを複数のインスタンスに関連付けます。出力では、成功したオペレーションと失敗したオペレーションのリストが返されます (該当する場合)。

コマンド:

```
aws ssm create-association-batch --entries "Name=AWS-  
UpdateSSMAgent, InstanceId=i-1234567890abcdef0" "Name=AWS-  
UpdateSSMAgent, InstanceId=i-9876543210abcdef0"
```

出力:

```
{
  "Successful": [
    {
      "Name": "AWS-UpdateSSMAgent",
      "InstanceId": "i-1234567890abcdef0",
      "AssociationVersion": "1",
      "Date": 1550504725.007,
      "LastUpdateAssociationDate": 1550504725.007,
      "Status": {
        "Date": 1550504725.007,
        "Name": "Associated",
        "Message": "Associated with AWS-UpdateSSMAgent"
      },
      "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Creating"
      },
      "DocumentVersion": "$DEFAULT",
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "Targets": [
        {
          "Key": "InstanceIds",
          "Values": [
            "i-1234567890abcdef0"
          ]
        }
      ]
    },
    {
      "Name": "AWS-UpdateSSMAgent",
      "InstanceId": "i-9876543210abcdef0",
      "AssociationVersion": "1",
      "Date": 1550504725.057,
      "LastUpdateAssociationDate": 1550504725.057,
      "Status": {
        "Date": 1550504725.057,
        "Name": "Associated",
        "Message": "Associated with AWS-UpdateSSMAgent"
      },
      "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Creating"
      }
    }
  ]
}
```

```

    },
    "DocumentVersion": "$DEFAULT",
    "AssociationId": "9c9f7f20-5154-4fed-a83e-0123456789ab",
    "Targets": [
      {
        "Key": "InstanceIds",
        "Values": [
          "i-9876543210abcdef0"
        ]
      }
    ]
  }
],
"Failed": []
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateAssociationBatch](#)」の「」を参照してください。

create-association

次の例は、create-association を使用する方法を説明しています。

AWS CLI

例 1: インスタンスを使用してドキュメントを関連付けるには IDs

この例では、インスタンスを使用して、設定ドキュメントをインスタンスに関連付けます IDs。

```

aws ssm create-association \
  --instance-id "i-0cb2b964d3e14fd9f" \
  --name "AWS-UpdateSSMAgent"

```

出力:

```

{
  "AssociationDescription": {
    "Status": {
      "Date": 1487875500.33,
      "Message": "Associated with AWS-UpdateSSMAgent",
      "Name": "Associated"
    }
  },

```

```

    "Name": "AWS-UpdateSSMAgent",
    "InstanceId": "i-0cb2b964d3e14fd9f",
    "Overview": {
      "Status": "Pending",
      "DetailedStatus": "Creating"
    },
    "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
    "DocumentVersion": "$DEFAULT",
    "LastUpdateAssociationDate": 1487875500.33,
    "Date": 1487875500.33,
    "Targets": [
      {
        "Values": [
          "i-0cb2b964d3e14fd9f"
        ],
        "Key": "InstanceIds"
      }
    ]
  }
}

```

詳細については、AWS「Systems Manager APIリファレンス[CreateAssociation](#)」の「」を参照してください。

例 2: ターゲットを使用してドキュメントを関連付けるには

この例では、ターゲットを使用して、設定ドキュメントをインスタンスに関連付けます。

```

aws ssm create-association \
  --name "AWS-UpdateSSMAgent" \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"

```

出力:

```

{
  "AssociationDescription": {
    "Status": {
      "Date": 1487875500.33,
      "Message": "Associated with AWS-UpdateSSMAgent",
      "Name": "Associated"
    },
    "Name": "AWS-UpdateSSMAgent",
    "InstanceId": "i-0cb2b964d3e14fd9f",

```

```

    "Overview": {
      "Status": "Pending",
      "DetailedStatus": "Creating"
    },
    "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
    "DocumentVersion": "$DEFAULT",
    "LastUpdateAssociationDate": 1487875500.33,
    "Date": 1487875500.33,
    "Targets": [
      {
        "Values": [
          "i-0cb2b964d3e14fd9f"
        ],
        "Key": "InstanceIds"
      }
    ]
  }
}

```

詳細については、AWS 「Systems Manager APIリファレンス[CreateAssociation](#)」の「」を参照してください。

例 3: 1 回だけ実行される関連付けを作成するには

この例では、指定された日付および時刻に 1 回だけ実行される新しい関連付けを作成します。過去または現在の日付で作成された関連付け (処理されるまで日付は過去のものです) は、ただちに実行されません。

```

aws ssm create-association \
  --name "AWS-UpdateSSMAgent" \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
  --schedule-expression "at(2020-05-14T15:55:00)" \
  --apply-only-at-cron-interval

```

出力:

```

{
  "AssociationDescription": {
    "Status": {
      "Date": 1487875500.33,
      "Message": "Associated with AWS-UpdateSSMAgent",
      "Name": "Associated"
    }
  }
}

```

```
    },
    "Name": "AWS-UpdateSSMAgent",
    "InstanceId": "i-0cb2b964d3e14fd9f",
    "Overview": {
      "Status": "Pending",
      "DetailedStatus": "Creating"
    },
    "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
    "DocumentVersion": "$DEFAULT",
    "LastUpdateAssociationDate": 1487875500.33,
    "Date": 1487875500.33,
    "Targets": [
      {
        "Values": [
          "i-0cb2b964d3e14fd9f"
        ],
        "Key": "InstanceIds"
      }
    ]
  }
}
```

詳細については、「Systems Manager ユーザーガイド [CreateAssociation](#)」の「[Systems Manager のリファレンス: Cron とレート式](#)」のAWS「」を参照してください。AWS API

- API 詳細については、AWS CLI「[コマンドリファレンスCreateAssociation](#)」の「」を参照してください。

create-document

次の例は、create-document を使用する方法を説明しています。

AWS CLI

ドキュメントを作成するには

次の create-document の例では、新しい Systems Manager ドキュメントを作成します。

```
aws ssm create-document \  
  --content file://exampleDocument.yml \  
  --name "Example" \  
  --document-type "Automation" \  
  --document-format YAML
```

出力:

```
{
  "DocumentDescription": {
    "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
    "HashType": "Sha256",
    "Name": "Example",
    "Owner": "29884EXAMPLE",
    "CreateDate": 1583256349.452,
    "Status": "Creating",
    "DocumentVersion": "1",
    "Description": "Document Example",
    "Parameters": [
      {
        "Name": "AutomationAssumeRole",
        "Type": "String",
        "Description": "(Required) The ARN of the role that allows Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses your IAM permissions to execute this document.",
        "DefaultValue": ""
      },
      {
        "Name": "InstanceId",
        "Type": "String",
        "Description": "(Required) The ID of the Amazon EC2 instance.",
        "DefaultValue": ""
      }
    ],
    "PlatformTypes": [
      "Windows",
      "Linux"
    ],
    "DocumentType": "Automation",
    "SchemaVersion": "0.3",
    "LatestVersion": "1",
    "DefaultVersion": "1",
    "DocumentFormat": "YAML",
    "Tags": []
  }
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[SSM ドキュメントコンテンツを作成する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDocument](#)」の「」を参照してください。

create-maintenance-window

次の例は、create-maintenance-window を使用する方法を説明しています。

AWS CLI

例 1: メンテナンスウィンドウを作成するには

次の create-maintenance-window の例では、5 分ごとに最大 2 時間 (必要に応じて) の新しいメンテナンスウィンドウを作成し、メンテナンスウィンドウの実行終了から 1 時間以内に新しいタスクが開始されないようにします。また、関連付けられていないターゲット (メンテナンスウィンドウに登録されていないインスタンス) を許可し、カスタムタグを使用することで、その作成者がチュートリアルで使用する意図があることを示します。

```
aws ssm create-maintenance-window \  
  --name "My-Tutorial-Maintenance-Window" \  
  --schedule "rate(5 minutes)" \  
  --duration 2 --cutoff 1 \  
  --allow-unassociated-targets \  
  --tags "Key=Purpose, Value=Tutorial"
```

出力:

```
{  
  "WindowId": "mw-0c50858d01EXAMPLE"  
}
```

例 2: 1 回だけ実行されるメンテナンスウィンドウを作成するには

次の create-maintenance-window の例では、指定した日付および時刻に 1 回だけ実行される新しいメンテナンスウィンドウを作成します。

```
aws ssm create-maintenance-window \  
  --name My-One-Time-Maintenance-Window \  
  --schedule "at(2020-05-14T15:55:00)" \  
  --duration 5 \  
  --cutoff 2 \  
  --allow-unassociated-targets
```

```
--allow-unassociated-targets \  
--tags "Key=Environment,Value=Production"
```

出力:

```
{  
  "WindowId": "mw-01234567890abcdef"  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Maintenance Windows](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateMaintenanceWindow](#)」の「」を参照してください。

create-ops-item

次の例は、create-ops-item を使用する方法を説明しています。

AWS CLI

を作成するには OpsItems

次のcreate-ops-item例では OperationalData、the /aws/resourcesのキーを使用して、Amazon DynamoDB 関連のリソース OpsItem を使用して を作成します。

```
aws ssm create-ops-item \  
  --title "EC2 instance disk full" \  
  --description "Log clean up may have failed which caused the disk to be full" \  
  --priority 2 \  
  --source ec2 \  
  --operational-data '{"/aws/resources":{"Value":["arn  
\: \": {"arn:aws:dynamodb:us-west-2:12345678:table/OpsItems  
\:}"],"Type":"SearchableString"}}' \  
  --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

出力:

```
{  
  "OpsItemId": "oi-1a2b3c4d5e6f"
```

```
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「[作成 OpsItems](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateOpsItem](#)」の「」を参照してください。

create-patch-baseline

次のコード例は、create-patch-baseline を使用する方法を示しています。

AWS CLI

例 1: 自動承認を設定したパッチベースラインを作成するには

次の create-patch-baseline の例では、Microsoft からリリースされてから 7 日後に本番環境のパッチを承認する Windows Server のパッチベースラインを作成します。

```
aws ssm create-patch-baseline \  
  --name "Windows-Production-Baseline-AutoApproval" \  
  --operating-system "WINDOWS" \  
  --approval-  
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo  
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv  
 \  
  --description "Baseline containing all updates approved for Windows Server  
production systems"
```

出力:

```
{  
  "BaselineId": "pb-045f10b4f3EXAMPLE"  
}
```

例 2: 承認のカットオフ日を設定したパッチベースラインを作成するには

次の create-patch-baseline の例では、2020 年 7 月 7 日より前にリリースされた本番環境のすべてのパッチを承認する Windows Server のパッチベースラインを作成します。

```
aws ssm create-patch-baseline \  

```

```

--name "Windows-Production-Baseline-AutoApproval" \
--operating-system "WINDOWS" \
--approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impo
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},Approv
\
--description "Baseline containing all updates approved for Windows Server
production systems"

```

出力:

```

{
  "BaselineId": "pb-045f10b4f3EXAMPLE"
}

```

例 3: JSON ファイルに保存されている承認ルールを使用してパッチベースラインを作成するには次の `create-patch-baseline` の例では、Amazon Linux 2017.09 のパッチベースラインを作成し、リリースされてから 7 日後に本番環境のパッチを承認し、パッチベースラインの承認ルールを指定し、パッチのカスタムリポジトリを指定します。

```

aws ssm create-patch-baseline \
--cli-input-json file://my-amazon-linux-approval-rules-and-repo.json

```

`my-amazon-linux-approval-rules-and-repo.json` の内容:

```

{
  "Name": "Amazon-Linux-2017.09-Production-Baseline",
  "Description": "My approval rules patch baseline for Amazon Linux 2017.09
instances",
  "OperatingSystem": "AMAZON_LINUX",
  "Tags": [
    {
      "Key": "Environment",
      "Value": "Production"
    }
  ],
  "ApprovalRules": {
    "PatchRules": [
      {
        "ApproveAfterDays": 7,
        "EnableNonSecurity": true,

```

```

        "PatchFilterGroup": {
            "PatchFilters": [
                {
                    "Key": "SEVERITY",
                    "Values": [
                        "Important",
                        "Critical"
                    ]
                },
                {
                    "Key": "CLASSIFICATION",
                    "Values": [
                        "Security",
                        "Bugfix"
                    ]
                },
                {
                    "Key": "PRODUCT",
                    "Values": [
                        "AmazonLinux2017.09"
                    ]
                }
            ]
        }
    ],
    "Sources": [
        {
            "Name": "My-AL2017.09",
            "Products": [
                "AmazonLinux2017.09"
            ],
            "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo.$awsregion.$awsdomain/$releasever/main/mirror.list //
nmirrorlist_expire=300//nmetadata_expire=300 \npriority=10 \nfailovermethod=priority
\nfastestmirror_enabled=0 \ngpgcheck=1 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-
KEY-amazon-ga \nenabled=1 \nretries=3 \ntimeout=5\nreport_instanceid=yes"
        }
    ]
}

```

例 4: 承認するパッチと拒否するパッチを指定したパッチベースラインを作成するには

次の `create-patch-baseline` の例では、デフォルトの承認ルールの例外として、承認するパッチと拒否するパッチを明示的に指定します。

```
aws ssm create-patch-baseline \  
  --name "Amazon-Linux-2017.09-Alpha-Baseline" \  
  --description "My custom approve/reject patch baseline for Amazon Linux 2017.09 instances" \  
  --operating-system "AMAZON_LINUX" \  
  --approved-patches "CVE-2018-1234567,example-pkg-EE-2018*.amzn1.noarch" \  
  --approved-patches-compliance-level "HIGH" \  
  --approved-patches-enable-non-security \  
  --tags "Key=Environment,Value=Alpha"
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Create a Custom Patch Baseline](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePatchBaseline](#)」の「」を参照してください。

create-resource-data-sync

次の例は、`create-resource-data-sync` を使用する方法を説明しています。

AWS CLI

リソースデータ同期を作成するには

この例では、リソースデータ同期を作成します。コマンドが成功した場合、出力はありません。

コマンド:

```
aws ssm create-resource-data-sync --sync-name "ssm-resource-data-sync" --s3-destination "BucketName=ssm-bucket,Prefix=inventory,SyncFormat=JsonSerDe,Region=us-east-1"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResourceDataSync](#)」の「」を参照してください。

delete-activation

次の例は、`delete-activation` を使用する方法を説明しています。

AWS CLI

マネージドインスタンスのアクティベーションを削除するには

次の delete-activation の例では、マネージドインスタンスのアクティベーションを削除します。

```
aws ssm delete-activation \  
  --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

このコマンドでは何も出力されません。

詳細については、[AWS 「Systems Manager ユーザーガイド」の「ハイブリッド環境用の Systems Manager のセットアップ」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteActivation](#)」の「」を参照してください。

delete-association

次のコード例は、delete-association を使用する方法を示しています。

AWS CLI

例 1: 関連付け ID を使用して関連付けを削除するには

次の delete-association の例では、指定された関連付け ID の関連付けを削除します。コマンドが成功した場合、出力はありません。

```
aws ssm delete-association \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの編集と新しいバージョンの作成](#)」を参照してください。

例 2: 関連付けを削除するには

次の delete-association の例では、インスタンスとドキュメント間の関連付けを削除します。コマンドが成功した場合、出力はありません。

```
aws ssm delete-association \  
  --instance-id "i-1234567890abcdef0" \  
  --name "AWS-UpdateSSMAgent"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Systems Manager の関連付けの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAssociation](#)」の「」を参照してください。

delete-document

次の例は、delete-document を使用する方法を説明しています。

AWS CLI

ドキュメントを削除するには

次の delete-document の例では、Systems Manager ドキュメントを削除します。

```
aws ssm delete-document \  
  --name "Example"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[SSM ドキュメントコンテンツを作成する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDocument](#)」の「」を参照してください。

delete-inventory

次のコード例は、delete-inventory を使用する方法を示しています。

AWS CLI

カスタムインベントリタイプを削除するには

この例では、カスタムインベントリスキーマを削除します。

コマンド:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-  
option "DeLeteSchema"
```

出力:

```
{  
  "DeletionId": "d72ac9e8-1f60-4d40-b1c6-bf8c78c68c4d",  
  "TypeName": "Custom:RackInfo",  
  "DeletionSummary": {  
    "TotalCount": 1,  
    "RemainingCount": 1,  
    "SummaryItems": [  
      {  
        "Version": "1.0",  
        "Count": 1,  
        "RemainingCount": 1  
      }  
    ]  
  }  
}
```

カスタムインベントリタイプを無効にするには

この例では、カスタムインベントリスキーマを無効にします。

コマンド:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-  
option "DisableSchema"
```

出力:

```
{  
  "DeletionId": "6961492a-8163-44ec-aa1e-923364dd0850",  
  "TypeName": "Custom:RackInformation",  
  "DeletionSummary": {  
    "TotalCount": 0,  
    "RemainingCount": 0,  
  }  
}
```

```
    "SummaryItems": []
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteInventory](#)」の「」を参照してください。

delete-maintenance-window

次の例は、delete-maintenance-window を使用する方法を説明しています。

AWS CLI

メンテナンスウィンドウを削除するには

この delete-maintenance-window の例では、指定されたメンテナンスウィンドウを削除します。

```
aws ssm delete-maintenance-window \
  --window-id "mw-1a2b3c4d5e6f7g8h9"
```

出力:

```
{
  "WindowId": "mw-1a2b3c4d5e6f7g8h9"
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウの削除 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMaintenanceWindow](#)」の「」を参照してください。

delete-parameter

次の例は、delete-parameter を使用する方法を説明しています。

AWS CLI

パラメータを削除するには

次の `delete-parameter` の例では、指定された単一のパラメータを削除します。

```
aws ssm delete-parameter \  
  --name "MyParameter"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Parameter Store の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteParameter](#)」の「」を参照してください。

delete-parameters

次のコード例は、`delete-parameters` を使用する方法を示しています。

AWS CLI

パラメータのリストを削除するには

次の `delete-parameters` 例では、指定されたパラメータを削除します。

```
aws ssm delete-parameters \  
  --names "MyFirstParameter" "MySecondParameter" "MyInvalidParameterName"
```

出力:

```
{  
  "DeletedParameters": [  
    "MyFirstParameter",  
    "MySecondParameter"  
  ],  
  "InvalidParameters": [  
    "MyInvalidParameterName"  
  ]  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Parameter Store の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteParameters](#)」の「」を参照してください。

delete-patch-baseline

次のコード例は、delete-patch-baseline を使用する方法を示しています。

AWS CLI

パッチベースラインを削除するには

次の delete-patch-baseline の例では、指定されたパッチベースラインを削除します。

```
aws ssm delete-patch-baseline \  
  --baseline-id "pb-045f10b4f382baeda"
```

出力:

```
{  
  "BaselineId": "pb-045f10b4f382baeda"  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Update or Delete a Patch Baseline \(Console\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePatchBaseline](#)」の「」を参照してください。

delete-resource-data-sync

次のコード例は、delete-resource-data-sync を使用する方法を示しています。

AWS CLI

リソースデータ同期を削除するには

この例では、リソースデータ同期を削除します。コマンドが成功した場合、出力はありません。

コマンド:

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResourceDataSync](#)」の「」を参照してください。

deregister-managed-instance

次の例は、deregister-managed-instance を使用する方法を説明しています。

AWS CLI

マネージドインスタンスを登録解除するには

次の deregister-managed-instance の例では、指定されたマネージドインスタンスを登録解除します。

```
aws ssm deregister-managed-instance
  --instance-id "mi-08ab247cdfEXAMPLE"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[ハイブリッドおよびマルチクラウド環境でのマネージドノードの登録解除](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterManagedInstance](#)」の「」を参照してください。

deregister-patch-baseline-for-patch-group

次の例は、deregister-patch-baseline-for-patch-group を使用する方法を説明しています。

AWS CLI

パッチグループをパッチベースラインから登録解除するには

次の deregister-patch-baseline-for-patch-group の例では、指定されたパッチグループを指定されたパッチベースラインから登録解除します。

```
aws ssm deregister-patch-baseline-for-patch-group \
  --patch-group "Production" \
  --baseline-id "pb-0ca44a362fEXAMPLE"
```

出力:

```
{
  "PatchGroup": "Production",
  "BaselineId": "pb-0ca44a362fEXAMPLE"
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パッチベースラインにパッチグループを追加します](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterPatchBaselineForPatchGroup](#)」の「」を参照してください。

deregister-target-from-maintenance-window

次の例は、deregister-target-from-maintenance-window を使用する方法を説明していません。

AWS CLI

メンテナンスウィンドウからターゲットを削除するには

次のderegister-target-from-maintenance-window の例では、指定されたターゲットを指定されたメンテナンスウィンドウから削除します。

```
aws ssm deregister-target-from-maintenance-window \
  --window-id "mw-ab12cd34ef56gh78" \
  --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

出力:

```
{
  "WindowId": "mw-ab12cd34ef56gh78",
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウの更新 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeregisterTargetFromMaintenanceWindow](#)」の「」を参照してください。

deregister-task-from-maintenance-window

次のコード例は、`deregister-task-from-maintenance-window` を使用する方法を示しています。

AWS CLI

メンテナンスウィンドウからタスクを削除するには

次の `deregister-task-from-maintenance-window` 例では、指定されたタスクを指定されたメンテナンスウィンドウから削除します。

```
aws ssm deregister-task-from-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

出力:

```
{  
  "WindowTaskId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",  
  "WindowId": "mw-ab12cd34ef56gh78"  
}
```

詳細については、[「Systems Manager ユーザーガイド」の「Systems Manager メンテナンス Windows チュートリアル \(AWS CLI\)」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスDeregisterTaskFromMaintenanceWindow](#)」の「」を参照してください。

describe-activations

次のコード例は、`describe-activations` を使用する方法を示しています。

AWS CLI

アクティベーションについて表示するには

次の `describe-activations` 例では、AWS アカウントのアクティベーションの詳細を一覧表示します。

```
aws ssm describe-activations
```

出力:

```
{
  "ActivationList": [
    {
      "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
      "Description": "Example1",
      "IamRole": "HybridWebServersRole",
      "RegistrationLimit": 5,
      "RegistrationsCount": 5,
      "ExpirationDate": 1584316800.0,
      "Expired": false,
      "CreateDate": 1581954699.792
    },
    {
      "ActivationId": "3ee0322b-f62d-40eb-b672-13ebfEXAMPLE",
      "Description": "Example2",
      "IamRole": "HybridDatabaseServersRole",
      "RegistrationLimit": 5,
      "RegistrationsCount": 5,
      "ExpirationDate": 1580515200.0,
      "Expired": true,
      "CreateDate": 1578064132.002
    }
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Step 4: Create a Managed-Instance Activation for a Hybrid Environment](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeActivations](#)」の「」を参照してください。

describe-association-execution-targets

次の例は、describe-association-execution-targets を使用する方法を説明しています。

AWS CLI

関連付けの実行の詳細情報を取得するには

次の describe-association-execution-targets の例では、指定された関連付けの実行を記述します。


```
aws ssm describe-association-execution-targets \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \  
  --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

出力:

```
{  
  "AssociationExecutionTargets": [  
    {  
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",  
      "AssociationVersion": "1",  
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",  
      "ResourceId": "i-1234567890abcdef0",  
      "ResourceType": "ManagedInstance",  
      "Status": "Success",  
      "DetailedStatus": "Success",  
      "LastExecutionDate": 1550505538.497,  
      "OutputSource": {  
        "OutputSourceId": "97fff367-fc5a-4299-aed8-0123456789ab",  
        "OutputSourceType": "RunCommand"  
      }  
    }  
  ]  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの履歴の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAssociationExecutionTargets](#)」の「」を参照してください。

describe-association-executions

次のコード例は、describe-association-executions を使用する方法を示しています。

AWS CLI

例 1: 関連付けのすべての実行に関する詳細情報を取得するには

次の describe-association-executions の例では、指定された関連付けのすべての実行を記述します。

```
aws ssm describe-association-executions \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

出力:

```
{  
  "AssociationExecutions": [  
    {  
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",  
      "AssociationVersion": "1",  
      "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",  
      "Status": "Success",  
      "DetailedStatus": "Success",  
      "CreatedTime": 1550505827.119,  
      "ResourceCountByStatus": "{Success=1}"  
    },  
    {  
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",  
      "AssociationVersion": "1",  
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",  
      "Status": "Success",  
      "DetailedStatus": "Success",  
      "CreatedTime": 1550505536.843,  
      "ResourceCountByStatus": "{Success=1}"  
    },  
    ...  
  ]  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの履歴の表示](#)」を参照してください。

例 2: 特定の日付および時刻より後における、関連付けのすべての実行の詳細情報を取得するには次の describe-association-executions の例では、指定した日付および時刻より後における関連付けのすべての実行を記述します。

```
aws ssm describe-association-executions \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \  
  --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

出力:

```
{
  "AssociationExecutions": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
      "Status": "Success",
      "DetailedStatus": "Success",
      "CreatedTime": 1550505827.119,
      "ResourceCountByStatus": "{Success=1}"
    },
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
      "Status": "Success",
      "DetailedStatus": "Success",
      "CreatedTime": 1550505536.843,
      "ResourceCountByStatus": "{Success=1}"
    },
    ...
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの履歴の表示](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAssociationExecutions](#)」の「」を参照してください。

describe-association

次の例は、describe-association を使用する方法を説明しています。

AWS CLI

例 1: 関連付けの詳細情報を取得するには

次の describe-association の例では、指定された関連付け ID の関連付けを記述します。

```
aws ssm describe-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

出力:

```
{
  "AssociationDescription": {
    "Name": "AWS-GatherSoftwareInventory",
    "AssociationVersion": "1",
    "Date": 1534864780.995,
    "LastUpdateAssociationDate": 1543235759.81,
    "Overview": {
      "Status": "Success",
      "AssociationStatusAggregatedCount": {
        "Success": 2
      }
    }
  },
  "DocumentVersion": "$DEFAULT",
  "Parameters": {
    "applications": [
      "Enabled"
    ],
    "awsComponents": [
      "Enabled"
    ],
    "customInventory": [
      "Enabled"
    ],
    "files": [
      ""
    ],
    "instanceDetailedInformation": [
      "Enabled"
    ],
    "networkConfig": [
      "Enabled"
    ],
    "services": [
      "Enabled"
    ],
    "windowsRegistry": [
      ""
    ],
    "windowsRoles": [
      "Enabled"
    ],
    "windowsUpdates": [
```

```

        "Enabled"
      ]
    },
    "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
    "Targets": [
      {
        "Key": "InstanceIds",
        "Values": [
          "*"
        ]
      }
    ],
    "ScheduleExpression": "rate(24 hours)",
    "LastExecutionDate": 1550501886.0,
    "LastSuccessfulExecutionDate": 1550501886.0,
    "AssociationName": "Inventory-Association"
  }
}

```

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの編集と新しいバージョンの作成](#)」を参照してください。

例 2: 特定のインスタンスとドキュメントの関連付けの詳細情報を取得するには

次の describe-association の例では、インスタンスとドキュメントの関連付けを記述します。

```

aws ssm describe-association \
  --instance-id "i-1234567890abcdef0" \
  --name "AWS-UpdateSSMAgent"

```

出力:

```

{
  "AssociationDescription": {
    "Status": {
      "Date": 1487876122.564,
      "Message": "Associated with AWS-UpdateSSMAgent",
      "Name": "Associated"
    },
    "Name": "AWS-UpdateSSMAgent",
    "InstanceId": "i-1234567890abcdef0",
    "Overview": {

```

```

        "Status": "Pending",
        "DetailedStatus": "Associated",
        "AssociationStatusAggregatedCount": {
            "Pending": 1
        }
    },
    "AssociationId": "d8617c07-2079-4c18-9847-1234567890ab",
    "DocumentVersion": "$DEFAULT",
    "LastUpdateAssociationDate": 1487876122.564,
    "Date": 1487876122.564,
    "Targets": [
        {
            "Values": [
                "i-1234567890abcdef0"
            ],
            "Key": "InstanceIds"
        }
    ]
}

```

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの編集と新しいバージョンの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAssociation](#)」の「」を参照してください。

describe-automation-executions

次の例は、describe-automation-executions を使用する方法を説明しています。

AWS CLI

オートメーションの実行を記述するには

次の describe-automation-executions の例では、オートメーションの実行の詳細情報を表示します。

```
aws ssm describe-automation-executions \
  --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

出力:

```
{
  "AutomationExecutionMetadataList": [
    {
      "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
      "DocumentName": "AWS-StartEC2Instance",
      "DocumentVersion": "1",
      "AutomationExecutionStatus": "Success",
      "ExecutionStartTime": 1583737233.748,
      "ExecutionEndTime": 1583737234.719,
      "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
      "LogFile": "",
      "Outputs": {},
      "Mode": "Auto",
      "Targets": [],
      "ResolvedTargets": {
        "ParameterValues": [],
        "Truncated": false
      },
      "AutomationType": "Local"
    }
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[シンプルなオートメーションを実行する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAutomationExecutions](#)」の「」を参照してください。

describe-automation-step-executions

次の例は、describe-automation-step-executions を使用する方法を説明しています。

AWS CLI

例 1: オートメーションの実行におけるすべてのステップを表示するには

次の describe-automation-step-executions の例では、オートメーションの実行におけるステップの詳細情報を表示します。

```
aws ssm describe-automation-step-executions \
```

```
--automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

出力:

```
{
  "StepExecutions": [
    {
      "StepName": "startInstances",
      "Action": "aws:changeInstanceState",
      "ExecutionStartTime": 1583737234.134,
      "ExecutionEndTime": 1583737234.672,
      "StepStatus": "Success",
      "Inputs": {
        "DesiredState": "\"running\"",
        "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
      },
      "Outputs": {
        "InstanceStates": [
          "running"
        ]
      },
      "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
      "OverriddenParameters": {}
    }
  ]
}
```

例 2: オートメーションの実行における特定のステップを表示するには

次の `describe-automation-step-executions` の例では、オートメーションの実行における特定のステップの詳細情報を表示します。

```
aws ssm describe-automation-step-executions \
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
  --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[オートメーションをステップごとに実行する \(コマンドライン\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeAutomationStepExecutions](#)」の「」を参照してください。

describe-available-patches

次の例は、describe-available-patches を使用する方法を説明しています。

AWS CLI

利用可能なパッチを取得するには

次のdescribe-available-patches例では、MSRC重要度が Critical の Windows Server 2019 で使用可能なすべてのパッチの詳細を取得します。

```
aws ssm describe-available-patches \  
  --  
  filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"
```

出力:

```
{  
  "Patches": [  
    {  
      "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",  
      "ReleaseDate": 1544047205.0,  
      "Title": "2018-11 Update for Windows Server 2019 for x64-based Systems (KB4470788)",  
      "Description": "Install this update to resolve issues in Windows. For a complete listing of the issues that are included in this update, see the associated Microsoft Knowledge Base article for more information. After you install this item, you may have to restart your computer.",  
      "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",  
      "Vendor": "Microsoft",  
      "ProductFamily": "Windows",  
      "Product": "WindowsServer2019",  
      "Classification": "SecurityUpdates",  
      "MsrcSeverity": "Critical",  
      "KbNumber": "KB4470788",  
      "MsrcNumber": "",  
      "Language": "All"  
    },  
    {  
      "Id": "c96115e1-5587-4115-b851-22baa46a3f11",  
      "ReleaseDate": 1549994410.0,  
      "Title": "2019-02 Security Update for Adobe Flash Player for Windows Server 2019 for x64-based Systems (KB4487038)",
```

```

        "Description": "A security issue has been identified in a Microsoft
software product that could affect your system. You can help protect your system
by installing this update from Microsoft. For a complete listing of the issues that
are included in this update, see the associated Microsoft Knowledge Base article.
After you install this update, you may have to restart your system.",
        "ContentUrl": "https://support.microsoft.com/en-us/kb/4487038",
        "Vendor": "Microsoft",
        "ProductFamily": "Windows",
        "Product": "WindowsServer2019",
        "Classification": "SecurityUpdates",
        "MsrcSeverity": "Critical",
        "KbNumber": "KB4487038",
        "MsrcNumber": "",
        "Language": "All"
    },
    ...
]
}

```

特定のパッチの詳細情報を取得するには

次の `describe-available-patches` の例では、指定されたパッチの詳細情報を取得します。

```

aws ssm describe-available-patches \
  --filters "Key=PATCH_ID,Values=KB4480979"

```

出力:

```

{
  "Patches": [
    {
      "Id": "680861e3-fb75-432e-818e-d72e5f2be719",
      "ReleaseDate": 1546970408.0,
      "Title": "2019-01 Security Update for Adobe Flash Player for Windows
Server 2016 for x64-based Systems (KB4480979)",
      "Description": "A security issue has been identified in a Microsoft
software product that could affect your system. You can help protect your system
by installing this update from Microsoft. For a complete listing of the issues that
are included in this update, see the associated Microsoft Knowledge Base article.
After you install this update, you may have to restart your system.",
      "ContentUrl": "https://support.microsoft.com/en-us/kb/4480979",
      "Vendor": "Microsoft",
      "ProductFamily": "Windows",
    }
  ]
}

```

```
        "Product": "WindowsServer2016",
        "Classification": "SecurityUpdates",
        "MsrcSeverity": "Critical",
        "KbNumber": "KB4480979",
        "MsrcNumber": "",
        "Language": "All"
    }
]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Patch Manager の動作の仕組み](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeAvailablePatches](#)」の「」を参照してください。

describe-document-permission

次の例は、describe-document-permission を使用する方法を説明しています。

AWS CLI

ドキュメントのアクセス許可を表示するには

次の describe-document-permission の例では、パブリックに共有されている Systems Manager ドキュメントに関するアクセス許可の詳細情報を表示します。

```
aws ssm describe-document-permission \
  --name "Example" \
  --permission-type "Share"
```

出力:

```
{
  "AccountIds": [
    "all"
  ],
  "AccountSharingInfoList": [
    {
      "AccountId": "all",
      "SharedDocumentVersion": "$DEFAULT"
    }
  ]
}
```

```
]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Share a Systems Manager Document](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDocumentPermission](#)」の「」を参照してください。

describe-document

次の例は、describe-document を使用する方法を説明しています。

AWS CLI

ドキュメントの詳細情報を表示するには

次のdescribe-document例では、AWS アカウントの Systems Manager ドキュメントの詳細を表示します。

```
aws ssm describe-document \
  --name "Example"
```

出力:

```
{
  "Document": {
    "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
    "HashType": "Sha256",
    "Name": "Example",
    "Owner": "29884EXAMPLE",
    "CreateDate": 1583257938.266,
    "Status": "Active",
    "DocumentVersion": "1",
    "Description": "Document Example",
    "Parameters": [
      {
        "Name": "AutomationAssumeRole",
        "Type": "String",
        "Description": "(Required) The ARN of the role that allows Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses your IAM permissions to execute this document.",
        "DefaultValue": ""
      }
    ]
  }
}
```

```
    },
    {
      "Name": "InstanceId",
      "Type": "String",
      "Description": "(Required) The ID of the Amazon EC2 instance.",
      "DefaultValue": ""
    }
  ],
  "PlatformTypes": [
    "Windows",
    "Linux"
  ],
  "DocumentType": "Automation",
  "SchemaVersion": "0.3",
  "LatestVersion": "1",
  "DefaultVersion": "1",
  "DocumentFormat": "YAML",
  "Tags": []
}
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[SSM ドキュメントコンテンツを作成する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeDocument](#)」の「」を参照してください。

describe-effective-instance-associations

次の例は、describe-effective-instance-associations を使用する方法を説明しています。

AWS CLI

インスタンスの有効な関連付けの詳細情報を取得するには

次の describe-effective-instance-associations の例では、インスタンスの有効な関連付けに関する詳細情報を取得します。

コマンド:

```
aws ssm describe-effective-instance-associations --instance-id "i-1234567890abcdef0"
```

出力:

```
{
  "Associations": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "InstanceId": "i-1234567890abcdef0",
      "Content": "{\n  \"schemaVersion\": \"1.2\",\n  \"description\":\n  \"Update the Amazon SSM Agent to the latest version or specified version.\",\n  \"parameters\": {\n    \"version\": {\n      \"default\": \"\",\n      \"description\": \"(Optional) A specific version of the Amazon SSM Agent\n  to install. If not specified, the agent will be updated to the latest version.\",\n      \"type\": \"String\"\n    },\n    \"allowDowngrade\": {\n      \"default\": \"false\",\n      \"description\": \"(Optional)\n  Allow the Amazon SSM Agent service to be downgraded to an earlier version. If\n  set to false, the service can be upgraded to newer versions only (default). If\n  set to true, specify the earlier version.\",\n      \"type\": \"String\",\n      \"allowedValues\": [\n        \"true\",\n        \"false\"\n      ]\n    },\n    \"runtimeConfig\": {\n      \"aws:updateSsmAgent\": {\n        \"properties\": [\n          {\n            \"agentName\": \"amazon-ssm-agent\",\n            \"source\":\n            \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-manifest.json\",\n            \"allowDowngrade\": \"{{ allowDowngrade }}\",\n            \"targetVersion\": \"{{ version }}\"\n          }\n        ]\n      }\n    }\n  }\n  \"AssociationVersion\": \"1\"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEffectiveInstanceAssociations](#)」の「」を参照してください。

describe-effective-patches-for-patch-baseline

次の例は、describe-effective-patches-for-patch-baseline を使用する方法を説明しています。

AWS CLI

例 1: カスタムパッチベースラインで定義されている、すべてのパッチを取得するには

次のdescribe-effective-patches-for-patch-baseline例では、現在の AWS アカウントのカスタムパッチベースラインで定義されたパッチを返します。カスタムベースラインの場合、--baseline-id には ID のみが必要であることを注意してください。

```
aws ssm describe-effective-patches-for-patch-baseline \  
  --baseline-id "pb-08b654cf9b9681f04"
```

出力:

```
{  
  "EffectivePatches": [  
    {  
      "Patch": {  
        "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",  
        "ReleaseDate": 1544047205.0,  
        "Title": "2018-11 Update for Windows Server 2019 for x64-based  
Systems (KB4470788)",  
        "Description": "Install this update to resolve issues in Windows.  
For a complete listing of the issues that are included in this update, see the  
associated Microsoft Knowledge Base article for more information. After you install  
this item, you may have to restart your computer.",  
        "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",  
        "Vendor": "Microsoft",  
        "ProductFamily": "Windows",  
        "Product": "WindowsServer2019",  
        "Classification": "SecurityUpdates",  
        "MsrcSeverity": "Critical",  
        "KbNumber": "KB4470788",  
        "MsrcNumber": "",  
        "Language": "All"  
      },  
      "PatchStatus": {  
        "DeploymentStatus": "APPROVED",  
        "ComplianceLevel": "CRITICAL",  
        "ApprovalDate": 1544047205.0  
      }  
    },  
    {  
      "Patch": {  
        "Id": "915a6b1a-f556-4d83-8f50-b2e75a9a7e58",  
        "ReleaseDate": 1549994400.0,  
        "Title": "2019-02 Cumulative Update for .NET Framework 3.5 and 4.7.2  
for Windows Server 2019 for x64 (KB4483452)",
```

```

        "Description": "A security issue has been identified in a Microsoft
software product that could affect your system. You can help protect your system by
installing this update from Microsoft. For a complete listing of the issues that
are included in this update, see the associated Microsoft Knowledge Base article.
After you install this update, you may have to restart your system.",
        "ContentUrl": "https://support.microsoft.com/en-us/kb/4483452",
        "Vendor": "Microsoft",
        "ProductFamily": "Windows",
        "Product": "WindowsServer2019",
        "Classification": "SecurityUpdates",
        "MsrcSeverity": "Important",
        "KbNumber": "KB4483452",
        "MsrcNumber": "",
        "Language": "All"
    },
    "PatchStatus": {
        "DeploymentStatus": "APPROVED",
        "ComplianceLevel": "CRITICAL",
        "ApprovalDate": 1549994400.0
    }
},
...
],
"NextToken": "--token string truncated--"
}

```

例 2: AWS マネージドパッチベースラインで定義されたすべてのパッチを取得するには

次のdescribe-effective-patches-for-patch-baseline例では、AWS マネージドパッチベースラインで定義されたパッチを返します。AWS マネージドベースラインでは、の完全なベースラインARNが に必要であることに注意してください。 --baseline-id

```

aws ssm describe-effective-patches-for-patch-baseline \
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"

```

出力例については、例 1 を参照してください。

詳細については、「AWS Systems Manager ユーザーガイド」の「[セキュリティに関連するパッチの選択方法](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeEffectivePatchesForPatchBaseline](#)」の「」を参照してください。

describe-instance-associations-status

次の例は、describe-instance-associations-status を使用する方法を説明しています。

AWS CLI

インスタンスの関連付けのステータスを表示するには

この例では、インスタンスの関連付けの詳細情報を表示します。

コマンド:

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

出力:

```
{
  "InstanceAssociationStatusInfos": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "Name": "AWS-GatherSoftwareInventory",
      "DocumentVersion": "1",
      "AssociationVersion": "1",
      "InstanceId": "i-1234567890abcdef0",
      "ExecutionDate": 1550501886.0,
      "Status": "Success",
      "ExecutionSummary": "1 out of 1 plugin processed, 1 success, 0 failed, 0
timedout, 0 skipped. ",
      "AssociationName": "Inventory-Association"
    },
    {
      "AssociationId": "5c5a31f6-6dae-46f9-944c-0123456789ab",
      "Name": "AWS-UpdateSSMAgent",
      "DocumentVersion": "1",
      "AssociationVersion": "1",
      "InstanceId": "i-1234567890abcdef0",
      "ExecutionDate": 1550505828.548,
      "Status": "Success",
      "DetailedStatus": "Success",
      "AssociationName": "UpdateSSMAgent"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceAssociationsStatus](#)」の「」を参照してください。

describe-instance-information

次の例は、describe-instance-information を使用する方法を説明しています。

AWS CLI

例 1: マネージドインスタンスの情報を表示するには

次の describe-instance-information の例では、各マネージドインスタンスの詳細情報を取得します。

```
aws ssm describe-instance-information
```

例 2: 特定のマネージドインスタンスに関する情報を表示するには

次の describe-instance-information の例では、マネージドインスタンス i-028ea792daEXAMPLE の詳細情報を表示します。

```
aws ssm describe-instance-information \  
  --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

例 3: 特定のタグキーを持つマネージドインスタンスに関する情報を表示するには

次の describe-instance-information の例では、タグキー DEV を持つマネージドインスタンスの詳細情報を表示します。

```
aws ssm describe-instance-information \  
  --filters "Key=tag-key,Values=DEV"
```

出力:

```
{  
  "InstanceInformationList": [  
    {  
      "InstanceId": "i-028ea792daEXAMPLE",  
      "PingStatus": "Online",
```

```
    "LastPingDateTime": 1582221233.421,
    "AgentVersion": "2.3.842.0",
    "IsLatestVersion": true,
    "PlatformType": "Linux",
    "PlatformName": "SLES",
    "PlatformVersion": "15.1",
    "ResourceType": "EC2Instance",
    "IPAddress": "192.0.2.0",
    "ComputerName": "ip-198.51.100.0.us-east-2.compute.internal",
    "AssociationStatus": "Success",
    "LastAssociationExecutionDate": 1582220806.0,
    "LastSuccessfulAssociationExecutionDate": 1582220806.0,
    "AssociationOverview": {
      "DetailedStatus": "Success",
      "InstanceAssociationStatusAggregatedCount": {
        "Success": 2
      }
    }
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Managed Instances](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInstanceInformation](#)」の「」を参照してください。

describe-instance-patch-states-for-patch-group

次の例は、describe-instance-patch-states-for-patch-group を使用する方法を説明しています。

AWS CLI

例 1: パッチグループのインスタンスの状態を取得するには

次のdescribe-instance-patch-states-for-patch-group の例では、指定されたパッチグループにおけるインスタンスごとのパッチの概要状態に関する詳細情報を取得します。

```
aws ssm describe-instance-patch-states-for-patch-group \
  --patch-group "Production"
```

出力:

```
{
  "InstancePatchStates": [
    {
      "InstanceId": "i-02573cafcfEXAMPLE",
      "PatchGroup": "Production",
      "BaselineId": "pb-0c10e65780EXAMPLE",
      "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
      "OwnerInformation": "",
      "InstalledCount": 32,
      "InstalledOtherCount": 1,
      "InstalledPendingRebootCount": 0,
      "InstalledRejectedCount": 0,
      "MissingCount": 2,
      "FailedCount": 0,
      "UnreportedNotApplicableCount": 2671,
      "NotApplicableCount": 400,
      "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
      "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
      "Operation": "Scan",
      "RebootOption": "NoReboot",
      "CriticalNonCompliantCount": 0,
      "SecurityNonCompliantCount": 1,
      "OtherNonCompliantCount": 0
    },
    {
      "InstanceId": "i-0471e04240EXAMPLE",
      "PatchGroup": "Production",
      "BaselineId": "pb-09ca3fb51fEXAMPLE",
      "SnapshotId": "05d8ffb0-1bbe-4812-ba2d-d9b7bEXAMPLE",
      "OwnerInformation": "",
      "InstalledCount": 32,
      "InstalledOtherCount": 1,
      "InstalledPendingRebootCount": 0,
      "InstalledRejectedCount": 0,
      "MissingCount": 2,
      "FailedCount": 0,
      "UnreportedNotApplicableCount": 2671,
      "NotApplicableCount": 400,
      "OperationStartTime": "2021-08-04T22:06:20.340000-07:00",
      "OperationEndTime": "2021-08-04T22:07:11.220000-07:00",
      "Operation": "Scan",
      "RebootOption": "NoReboot",
    }
  ]
}
```

```

        "CriticalNonCompliantCount": 0,
        "SecurityNonCompliantCount": 1,
        "OtherNonCompliantCount": 0
    }
]
}

```

例 2: パッチグループの欠落しているパッチが 5 個以上あるインスタンスの状態を取得するには次の `describe-instance-patch-states-for-patch-group` の例では、指定されたパッチグループにおいて、欠落しているパッチが 5 個以上あるインスタンスのパッチの概要状態に関する詳細情報を取得します。

```

aws ssm describe-instance-patch-states-for-patch-group \
  --filters Key=MissingCount,Type=GreaterThan,Values=5 \
  --patch-group "Production"

```

出力:

```

{
  "InstancePatchStates": [
    {
      "InstanceId": "i-02573cafcfEXAMPLE",
      "PatchGroup": "Production",
      "BaselineId": "pb-0c10e65780EXAMPLE",
      "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
      "OwnerInformation": "",
      "InstalledCount": 46,
      "InstalledOtherCount": 4,
      "InstalledPendingRebootCount": 1,
      "InstalledRejectedCount": 1,
      "MissingCount": 7,
      "FailedCount": 0,
      "UnreportedNotApplicableCount": 232,
      "NotApplicableCount": 654,
      "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
      "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
      "Operation": "Scan",
      "RebootOption": "NoReboot",
      "CriticalNonCompliantCount": 0,
      "SecurityNonCompliantCount": 1,
      "OtherNonCompliantCount": 1
    }
  ]
}

```

```
]
}
```

例 3: パッチグループにおいて、再起動が必要なインスタンスが 10 個未満のインスタンスの状態を取得するには

次のdescribe-instance-patch-states-for-patch-group の例では、指定されたパッチグループにおいて、再起動が必要なインスタンスが 10 個未満であるインスタンスのパッチの概要状態に関する詳細情報を取得します。

```
aws ssm describe-instance-patch-states-for-patch-group \
  --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
  --patch-group "Production"
```

出力:

```
{
  "InstancePatchStates": [
    {
      "InstanceId": "i-02573cafcfEXAMPLE",
      "BaselineId": "pb-0c10e65780EXAMPLE",
      "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
      "PatchGroup": "Production",
      "OwnerInformation": "",
      "InstalledCount": 32,
      "InstalledOtherCount": 1,
      "InstalledPendingRebootCount": 4,
      "InstalledRejectedCount": 0,
      "MissingCount": 2,
      "FailedCount": 0,
      "UnreportedNotApplicableCount": 846,
      "NotApplicableCount": 212,
      "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
      "OperationEndTime": "2021-08-06T11:04:21.555000-07:00",
      "Operation": "Scan",
      "RebootOption": "NoReboot",
      "CriticalNonCompliantCount": 0,
      "SecurityNonCompliantCount": 1,
      "OtherNonCompliantCount": 0
    }
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パッチコンプライアンス状態の値について](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstancePatchStatesForPatchGroup](#)」の「」を参照してください。

describe-instance-patch-states

次の例は、describe-instance-patch-states を使用する方法を説明しています。

AWS CLI

インスタンスのパッチの概要状態を取得するには

この describe-instance-patch-states の例では、インスタンスのパッチの概要状態を取得します。

```
aws ssm describe-instance-patch-states \  
  --instance-ids "i-1234567890abcdef0"
```

出力:

```
{  
  "InstancePatchStates": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "PatchGroup": "my-patch-group",  
      "BaselineId": "pb-0713accee01234567",  
      "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",  
      "CriticalNonCompliantCount": 2,  
      "SecurityNonCompliantCount": 2,  
      "OtherNonCompliantCount": 1,  
      "InstalledCount": 123,  
      "InstalledOtherCount": 334,  
      "InstalledPendingRebootCount": 0,  
      "InstalledRejectedCount": 0,  
      "MissingCount": 1,  
      "FailedCount": 2,  
      "UnreportedNotApplicableCount": 11,  
      "NotApplicableCount": 2063,  
      "OperationStartTime": "2021-05-03T11:00:56-07:00",  
      "OperationEndTime": "2021-05-03T11:01:09-07:00",  
      "Operation": "Scan",
```

```
        "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
        "RebootOption": "RebootIfNeeded"
    }
]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[About Patch Compliance](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInstancePatchStates](#)」の「」を参照してください。

describe-instance-patches

次の例は、describe-instance-patches を使用する方法を説明しています。

AWS CLI

例 1: インスタンスのパッチ状態の詳細を取得するには

次の describe-instance-patches の例では、指定されたインスタンスのパッチに関する詳細情報を取得します。

```
aws ssm describe-instance-patches \
  --instance-id "i-1234567890abcdef0"
```

出力:

```
{
  "Patches": [
    {
      "Title": "2019-01 Security Update for Adobe Flash Player for Windows Server 2016 for x64-based Systems (KB4480979)",
      "KBId": "KB4480979",
      "Classification": "SecurityUpdates",
      "Severity": "Critical",
      "State": "Installed",
      "InstalledTime": "2019-01-09T00:00:00+00:00"
    },
    {
      "Title": "",
      "KBId": "KB4481031",
```



```

        "Classification": "",
        "Severity": "",
        "State": "InstalledOther",
        "InstalledTime": "2019-02-08T00:00:00+00:00"
    },
    ...
],
"NextToken": "--token string truncated--"
}

```

例 2: インスタンスで欠落しているパッチのリストを取得するには

次のdescribe-instance-patches の例では、指定されたインスタンスで欠落しているパッチに関する情報を取得します。

```

aws ssm describe-instance-patches \
  --instance-id "i-1234567890abcdef0" \
  --filters Key=State,Values=Missing

```

出力:

```

{
  "Patches": [
    {
      "Title": "Windows Malicious Software Removal Tool x64 - February 2019 (KB890830)",
      "KBId": "KB890830",
      "Classification": "UpdateRollups",
      "Severity": "Unspecified",
      "State": "Missing",
      "InstalledTime": "1970-01-01T00:00:00+00:00"
    },
    ...
  ],
  "NextToken": "--token string truncated--"
}

```

詳細については、「AWS Systems Manager」の「[パッチコンプライアンス状態の値について](#)」を参照してください。

例 3: インスタンス InstalledTime に指定された 以降にインストールされたパッチのリストを取得するには

次のdescribe-instance-patches の例では、--filters と --query を組み合わせて、指定されたインスタンスに対して、指定された時刻よりも後にインストールされたパッチに関する情報を取得します。

```
aws ssm describe-instance-patches \  
  --instance-id "i-1234567890abcdef0" \  
  --filters Key=State,Values=Installed \  
  --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

出力:

```
{  
  "Patches": [  
    {  
      "Title": "2023-03 Cumulative Update for Windows Server 2019 (1809) for  
x64-based Systems (KB5023702)",  
      "KBId": "KB5023702",  
      "Classification": "SecurityUpdates",  
      "Severity": "Critical",  
      "State": "Installed",  
      "InstalledTime": "2023-03-16T11:00:00+00:00"  
    },  
    ...  
  ],  
  "NextToken": "--token string truncated--"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInstancePatches](#)」の「」を参照してください。

describe-inventory-deletions

次のコード例は、describe-inventory-deletions を使用する方法を示しています。

AWS CLI

インベントリの削除を取得するには

この例では、インベントリ削除オペレーションの詳細を取得します。

コマンド:

aws ssm describe-inventory-deletions

出力:

```
{
  "InventoryDeletions": [
    {
      "DeletionId": "6961492a-8163-44ec-aa1e-01234567850",
      "TypeName": "Custom:RackInformation",
      "DeletionStartTime": 1550254911.0,
      "LastStatus": "InProgress",
      "LastStatusMessage": "The Delete is in progress",
      "DeletionSummary": {
        "TotalCount": 0,
        "RemainingCount": 0,
        "SummaryItems": []
      },
      "LastStatusUpdateTime": 1550254911.0
    },
    {
      "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
      "TypeName": "Custom:RackInfo",
      "DeletionStartTime": 1550254859.0,
      "LastStatus": "InProgress",
      "LastStatusMessage": "The Delete is in progress",
      "DeletionSummary": {
        "TotalCount": 1,
        "RemainingCount": 1,
        "SummaryItems": [
          {
            "Version": "1.0",
            "Count": 1,
            "RemainingCount": 1
          }
        ]
      },
      "LastStatusUpdateTime": 1550254859.0
    }
  ]
}
```

特定のインベントリ削除の詳細を取得するには

この例では、特定のインベントリ削除オペレーションの詳細を取得します。

コマンド:

```
aws ssm describe-inventory-deletions --deletion-id "d72ac9e8-1f60-4d40-b1c6-987654321c4d"
```

出力:

```
{
  "InventoryDeletions": [
    {
      "DeletionId": "d72ac9e8-1f60-4d40-b1c6-987654321c4d",
      "TypeName": "Custom:RackInfo",
      "DeletionStartTime": 1550254859.0,
      "LastStatus": "InProgress",
      "LastStatusMessage": "The Delete is in progress",
      "DeletionSummary": {
        "TotalCount": 1,
        "RemainingCount": 1,
        "SummaryItems": [
          {
            "Version": "1.0",
            "Count": 1,
            "RemainingCount": 1
          }
        ]
      },
      "LastStatusUpdateTime": 1550254859.0
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeInventoryDeletions](#)」の「」を参照してください。

describe-maintenance-window-execution-task-invocations

次の例は、describe-maintenance-window-execution-task-invocations を使用する方法を説明しています。

AWS CLI

メンテナンスウィンドウのタスク実行で実行される、特定のタスク呼び出しを取得するには

次の `describe-maintenance-window-execution-task-invocations` の例では、指定されたメンテナンスウィンドウの実行の一部として実行される、指定したタスクの呼び出しを一覧表示します。

```
aws ssm describe-maintenance-window-execution-task-invocations \
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
  --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

出力:

```
{
  "WindowExecutionTaskInvocationIdentities": [
    {
      "Status": "SUCCESS",
      "Parameters": "{\"documentName\": \"AWS-RunShellScript\", \"instanceIds\": [\"i-0000293ffd8c57862\"], \"parameters\": {\"commands\": [\"df\"]}, \"maxConcurrency\": \"1\", \"maxErrors\": \"1\"}",
      "InvocationId": "e274b6e1-fe56-4e32-bd2a-8073c6381d8b",
      "StartTime": 1487692834.723,
      "EndTime": 1487692834.871,
      "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2a638355",
      "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d384503b6586"
    }
  ]
}
```

詳細については、AWS「Systems Manager ユーザーガイド」の [「タスクとタスク実行に関する情報の表示 \(AWS CLI\)」](#) を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DescribeMaintenanceWindowExecutionTaskInvocations](#)」の「」を参照してください。

`describe-maintenance-window-execution-tasks`

次のコード例は、`describe-maintenance-window-execution-tasks` を使用する方法を示しています。

AWS CLI

メンテナンスウィンドウの実行に関連するすべてのタスクを一覧表示するには

次の `ssm describe-maintenance-window-execution-tasks` の例では、指定されたメンテナンスウィンドウの実行に関連するタスクを一覧表示します。

```
aws ssm describe-maintenance-window-execution-tasks \  
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

出力:

```
{  
  "WindowExecutionTaskIdentities": [  
    {  
      "Status": "SUCCESS",  
      "TaskArn": "AWS-RunShellScript",  
      "StartTime": 1487692834.684,  
      "TaskType": "RUN_COMMAND",  
      "EndTime": 1487692835.005,  
      "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",  
      "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"  
    }  
  ]  
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の [「タスクとタスク実行に関する情報の表示 \(AWS CLI\)」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeMaintenanceWindowExecutionTasks](#)」の「」を参照してください。

describe-maintenance-window-executions

次の例は、`describe-maintenance-window-executions` を使用する方法を説明しています。

AWS CLI

例 1: メンテナンスウィンドウにおけるすべての実行を一覧表示するには

次の `describe-maintenance-window-executions` の例では、指定されたメンテナンスウィンドウにおけるすべての実行を一覧表示します。

```
aws ssm describe-maintenance-window-executions \  
  --window-id "mw-ab12cd34eEXAMPLE"
```

出力:

```
{  
  "WindowExecutions": [  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",  
      "Status": "IN_PROGRESS",  
      "StartTime": "2021-08-04T11:00:00.000000-07:00"  
    },  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "WindowExecutionId": "ff75b750-4834-4377-8f61-b3cadEXAMPLE",  
      "Status": "SUCCESS",  
      "StartTime": "2021-08-03T11:00:00.000000-07:00",  
      "EndTime": "2021-08-03T11:37:21.450000-07:00"  
    },  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",  
      "Status": "FAILED",  
      "StatusDetails": "One or more tasks in the orchestration failed.",  
      "StartTime": "2021-08-02T11:00:00.000000-07:00",  
      "EndTime": "2021-08-02T11:22:36.190000-07:00"  
    }  
  ]  
}
```

例 2: メンテナンスウィンドウにおける指定された日付より前のすべての実行を一覧表示するには次の `describe-maintenance-window-executions` の例では、指定されたメンテナンスウィンドウにおける指定された日付より前のすべての実行を一覧表示します。

```
aws ssm describe-maintenance-window-executions \  
  --window-id "mw-ab12cd34eEXAMPLE" \  
  --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

出力:

```
{
  "WindowExecutions": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
      "Status": "FAILED",
      "StatusDetails": "One or more tasks in the orchestration failed.",
      "StartTime": "2021-08-02T11:00:00.000000-07:00",
      "EndTime": "2021-08-02T11:22:36.190000-07:00"
    }
  ]
}
```

例 3: メンテナンスウィンドウにおける指定された日付より後のすべての実行を一覧表示するには次の `describe-maintenance-window-executions` の例では、メンテナンスウィンドウにおける指定された日付より後のすべての実行を一覧表示します。

```
aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

出力:

```
{
  "WindowExecutions": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
      "Status": "IN_PROGRESS",
      "StartTime": "2021-08-04T11:00:00.000000-07:00"
    }
  ]
}
```

詳細については、AWS Systems Manager ユーザーガイドの [「タスクとタスク実行 \(AWS CLI\) に関する情報を表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeMaintenanceWindowExecutions](#)」の「」を参照してください。

describe-maintenance-window-schedule

次のコード例は、describe-maintenance-window-schedule を使用する方法を示しています。

AWS CLI

例 1: メンテナンスウィンドウの今後の実行を一覧表示するには

次のdescribe-maintenance-window-schedule例では、指定されたメンテナンスウィンドウに対して予定されているすべての実行を一覧表示します。

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-ab12cd34eEXAMPLE
```

出力:

```
{  
  "ScheduledWindowExecutions": [  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "Name": "My-First-Maintenance-Window",  
      "ExecutionTime": "2020-02-19T16:00Z"  
    },  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "Name": "My-First-Maintenance-Window",  
      "ExecutionTime": "2020-02-26T16:00Z"  
    },  
    ...  
  ]  
}
```

例 2: 指定した日付より前のメンテナンスウィンドウの今後の実行をすべて一覧表示するには

次のdescribe-maintenance-window-schedule例では、指定された日付より前に発生した、指定されたメンテナンスウィンドウの今後の実行をすべて一覧表示します。

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-0ecb1226dd7b2e9a6 \  
  --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

例 3: 指定した日付以降にメンテナンスウィンドウに対して予定されているすべての実行を一覧表示するには

次のdescribe-maintenance-window-schedule例では、指定された日付以降に発生する、指定されたメンテナンスウィンドウの今後の実行をすべて一覧表示します。

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-0ecb1226dd7b2e9a6 \  
  --filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

詳細については、AWS「Systems Manager ユーザーガイド」の[「メンテナンスウィンドウ \(AWS CLI\) に関する情報を表示する」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeMaintenanceWindowSchedule](#)」の「」を参照してください。

describe-maintenance-window-targets

次のコード例は、describe-maintenance-window-targets を使用する方法を示しています。

AWS CLI

例 1: メンテナンスウィンドウのすべてのターゲットを一覧表示するには

次の describe-maintenance-window-targets の例では、メンテナンスウィンドウのすべてのターゲットを一覧表示します。

```
aws ssm describe-maintenance-window-targets \  
  --window-id "mw-06cf17cbefEXAMPLE"
```

出力:

```
{  
  "Targets": [  
    {  
      "ResourceType": "INSTANCE",  
      "OwnerInformation": "Single instance",  
      "WindowId": "mw-06cf17cbefEXAMPLE",  
      "Targets": [  
        {  
          "Values": [  
            "i-0000293ffdEXAMPLE"  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```

        ],
        "Key": "InstanceIds"
      }
    ],
    "WindowTargetId": "350d44e6-28cc-44e2-951f-4b2c9EXAMPLE"
  },
  {
    "ResourceType": "INSTANCE",
    "OwnerInformation": "Two instances in a list",
    "WindowId": "mw-06cf17cbefEXAMPLE",
    "Targets": [
      {
        "Values": [
          "i-0000293ffdEXAMPLE",
          "i-0cb2b964d3EXAMPLE"
        ],
        "Key": "InstanceIds"
      }
    ],
    "WindowTargetId": "e078a987-2866-47be-bedd-d9cf4EXAMPLE"
  }
]
}

```

例 2: 特定の所有者情報の値に一致するメンテナンスウィンドウのターゲットを一覧表示するには

この `describe-maintenance-window-targets` の例では、特定の値を持つメンテナンスウィンドウにおける、すべてのターゲットを一覧表示します。

```

aws ssm describe-maintenance-window-targets \
  --window-id "mw-0ecb1226ddEXAMPLE" \
  --filters "Key=OwnerInformation,Values=CostCenter1"

```

出力:

```

{
  "Targets": [
    {
      "WindowId": "mw-0ecb1226ddEXAMPLE",
      "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
      "ResourceType": "INSTANCE",
      "Targets": [
        {

```

```

        "Key": "tag:Environment",
        "Values": [
            "Prod"
        ]
    },
    "OwnerInformation": "CostCenter1",
    "Name": "ProdTarget1"
}
]
}

```

詳細については、AWS「Systems Manager ユーザーガイド」の[「メンテナンスウィンドウ \(AWS CLI\) に関する情報を表示する」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [DescribeMaintenanceWindowTargets](#)」の「」を参照してください。

describe-maintenance-window-tasks

次の例は、describe-maintenance-window-tasks を使用する方法を説明しています。

AWS CLI

例 1: メンテナンスウィンドウのすべてのタスクを一覧表示するには

次の describe-maintenance-window-tasks の例では、指定されたメンテナンスウィンドウのすべてのタスクを一覧表示します。

```
aws ssm describe-maintenance-window-tasks \
  --window-id "mw-06cf17cbefEXAMPLE"
```

出力:

```
{
  "Tasks": [
    {
      "WindowId": "mw-06cf17cbefEXAMPLE",
      "WindowTaskId": "018b31c3-2d77-4b9e-bd48-c91edEXAMPLE",
      "TaskArn": "AWS-RestartEC2Instance",
      "TaskParameters": {},
      "Type": "AUTOMATION",
      "Description": "Restarting EC2 Instance for maintenance",
    }
  ]
}
```

```

        "MaxConcurrency": "1",
        "MaxErrors": "1",
        "Name": "My-Automation-Example-Task",
        "Priority": 0,
        "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
        "Targets": [
            {
                "Key": "WindowTargetIds",
                "Values": [
                    "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                ]
            }
        ]
    },
    {
        "WindowId": "mw-06cf17cbefEXAMPLE",
        "WindowTaskId": "1943dee0-0a17-4978-9bf4-3cc2fEXAMPLE",
        "TaskArn": "AWS-DisableS3BucketPublicReadWrite",
        "TaskParameters": {},
        "Type": "AUTOMATION",
        "Description": "Automation task to disable read/write access on public
S3 buckets",
        "MaxConcurrency": "10",
        "MaxErrors": "5",
        "Name": "My-Disable-S3-Public-Read-Write-Access-Automation-Task",
        "Priority": 0,
        "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
        "Targets": [
            {
                "Key": "WindowTargetIds",
                "Values": [
                    "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                ]
            }
        ]
    }
]
}

```

例 2: AWS- RunPowerShellScript コマンドドキュメントを呼び出すメンテナンスウィンドウのすべてのタスクを一覧表示するには

次の `describe-maintenance-window-tasks` の例では、`AWS-RunPowerShellScript` コマンドドキュメントを呼び出す、指定されたメンテナンスウィンドウのタスクを一覧表示します。

```
aws ssm describe-maintenance-window-tasks \  
  --window-id "mw-ab12cd34eEXAMPLE" \  
  --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

出力:

```
{  
  "Tasks": [  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",  
      "TaskArn": "AWS-RunPowerShellScript",  
      "Type": "RUN_COMMAND",  
      "Targets": [  
        {  
          "Key": "WindowTargetIds",  
          "Values": [  
            "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"  
          ]  
        }  
      ],  
      "TaskParameters": {},  
      "Priority": 1,  
      "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/  
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",  
      "MaxConcurrency": "1",  
      "MaxErrors": "1",  
      "Name": "MyTask"  
    }  
  ]  
}
```

例 3: メンテナンスウィンドウのタスクのうち、`Priority` が 3 のすべてのタスクを一覧表示するには

次の `describe-maintenance-window-tasks` の例では、指定されたメンテナンスウィンドウのタスクのうち、`Priority` が 3 であるすべてのタスクを一覧表示します。

```
aws ssm describe-maintenance-window-tasks \  
  --window-id "mw-ab12cd34eEXAMPLE" \  
  --filters "Key=Priority,Values=3"
```

```
--window-id "mw-ab12cd34eEXAMPLE" \  
--filters "Key=Priority,Values=3"
```

出力:

```
{  
  "Tasks": [  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",  
      "TaskArn": "AWS-RunPowerShellScript",  
      "Type": "RUN_COMMAND",  
      "Targets": [  
        {  
          "Key": "WindowTargetIds",  
          "Values": [  
            "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"  
          ]  
        }  
      ],  
      "TaskParameters": {},  
      "Priority": 3,  
      "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/  
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",  
      "MaxConcurrency": "1",  
      "MaxErrors": "1",  
      "Name": "MyRunCommandTask"  
    },  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "WindowTaskId": "ee45feff-ad65-4a6c-b478-5cab8EXAMPLE",  
      "TaskArn": "AWS-RestartEC2Instance",  
      "Type": "AUTOMATION",  
      "Targets": [  
        {  
          "Key": "WindowTargetIds",  
          "Values": [  
            "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"  
          ]  
        }  
      ],  
      "TaskParameters": {},  
      "Priority": 3,  
    }  
  ]  
}
```

```

        "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
        "MaxConcurrency": "10",
        "MaxErrors": "5",
        "Name": "My-Automation-Task",
        "Description": "A description for my Automation task"
    }
]
}

```

例 4: メンテナンスウィンドウのタスクのうち、Priority が 1 で Run Command を使用するすべてのタスクを一覧表示するには

この describe-maintenance-window-tasks の例では、指定されたメンテナンスウィンドウのタスクのうち、Priority が 1 で Run Command を使用するすべてのタスクを一覧表示します。

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"

```

出力:

```

{
  "Tasks": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
      "TaskArn": "AWS-RunPowerShellScript",
      "Type": "RUN_COMMAND",
      "Targets": [
        {
          "Key": "WindowTargetIds",
          "Values": [
            "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
          ]
        }
      ],
      "TaskParameters": {},
      "Priority": 1,
      "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",

```



```
        "MaxConcurrency": "1",
        "MaxErrors": "1",
        "Name": "MyRunCommandTask"
    }
]
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウ \(AWS CLI\) に関する情報を表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeMaintenanceWindowTasks](#)」の「」を参照してください。

describe-maintenance-windows-for-target

次の例は、describe-maintenance-windows-for-target を使用する方法を説明しています。

AWS CLI

特定のインスタンスに関連付けられているすべてのメンテナンスウィンドウを一覧表示するには、次のdescribe-maintenance-windows-for-target例では、指定されたインスタンスに関連付けられたターゲットまたはタスクを持つメンテナンスウィンドウを一覧表示します。

```
aws ssm describe-maintenance-windows-for-target \
  --targets Key=InstanceIds,Values=i-1234567890EXAMPLE \
  --resource-type INSTANCE
```

出力:

```
{
  "WindowIdentities": [
    {
      "WindowId": "mw-0c5ed765acEXAMPLE",
      "Name": "My-First-Maintenance-Window"
    }
  ]
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「[メンテナンスウィンドウ \(AWS CLI\) に関する情報を表示する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeMaintenanceWindowsForTarget](#)」の「」を参照してください。

describe-maintenance-windows

次の例は、describe-maintenance-windows を使用する方法を説明しています。

AWS CLI

例 1: すべてのメンテナンスウィンドウを一覧表示するには

次のdescribe-maintenance-windows例では、現在のリージョンの AWS アカウント内のすべてのメンテナンスウィンドウを一覧表示します。

```
aws ssm describe-maintenance-windows
```

出力:

```
{
  "WindowIdentities": [
    {
      "WindowId": "mw-0ecb1226ddEXAMPLE",
      "Name": "MyMaintenanceWindow-1",
      "Enabled": true,
      "Duration": 2,
      "Cutoff": 1,
      "Schedule": "rate(180 minutes)",
      "NextExecutionTime": "2020-02-12T23:19:20.596Z"
    },
    {
      "WindowId": "mw-03eb9db428EXAMPLE",
      "Name": "MyMaintenanceWindow-2",
      "Enabled": true,
      "Duration": 3,
      "Cutoff": 1,
      "Schedule": "rate(7 days)",
      "NextExecutionTime": "2020-02-17T23:22:00.956Z"
    }
  ]
}
```

例 2: すべての有効なメンテナンスウィンドウを一覧表示するには

次の describe-maintenance-windows の例では、すべての有効なメンテナンスウィンドウを一覧表示します。

```
aws ssm describe-maintenance-windows \  
  --filters "Key=Enabled,Values=true"
```

例 3: 特定の名前に一致するメンテナンスウィンドウを一覧表示するには

この describe-maintenance-windows の例では、指定された名前を持つすべてのメンテナンスウィンドウを一覧表示します。

```
aws ssm describe-maintenance-windows \  
  --filters "Key=Name,Values=MyMaintenanceWindow"
```

詳細については、AWS 「Systems Manager ユーザーガイド」の [「メンテナンスウィンドウ \(AWS CLI\) に関する情報を表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeMaintenanceWindows](#)」の「」を参照してください。

describe-ops-items

次の例は、describe-ops-items を使用する方法を説明しています。

AWS CLI

のセットを一覧表示するには OpsItems

次の describe-ops-items 例では、AWS アカウント OpsItems で開いているすべての のリストを表示します。

```
aws ssm describe-ops-items \  
  --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

出力:

```
{  
  "OpsItemSummaries": [  
    {  
      "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/  
fbf77cbe264a33509569f23e4EXAMPLE",  
      "CreatedTime": "2020-03-14T17:02:46.375000-07:00",
```

```

    "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
    "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",
    "Source": "SSM",
    "Status": "Open",
    "OpsItemId": "oi-7cfc5EXAMPLE",
    "Title": "SSM Maintenance Window execution failed",
    "OperationalData": {
      "/aws/dedup": {
        "Value": "{\"dedupString\": \"SSMOpsItems-SSM-maintenance-window-
execution-failed\"}",
        "Type": "SearchableString"
      },
      "/aws/resources": {
        "Value": "[{\"arn\": \"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\"}]",
        "Type": "SearchableString"
      }
    },
    "Category": "Availability",
    "Severity": "3"
  },
  {
    "CreatedBy": "arn:aws:sts::1112223233444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
    "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
    "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
    "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
    "Source": "EC2",
    "Status": "Open",
    "OpsItemId": "oi-6f966EXAMPLE",
    "Title": "EC2 instance stopped",
    "OperationalData": {
      "/aws/automations": {
        "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
\"automationId\": \"AWS-RestartEC2Instance\" } ]",
        "Type": "SearchableString"
      },
      "/aws/dedup": {
        "Value": "{\"dedupString\": \"SSMOpsItems-EC2-instance-stopped
\"}",
        "Type": "SearchableString"
      }
    },
  }

```

```
    "/aws/resources": {
      "Value": "[{\"arn\":\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\"}]",
      "Type": "SearchableString"
    }
  },
  "Category": "Availability",
  "Severity": "3"
}
]
```

詳細については、AWS「Systems Manager ユーザーガイド」の「[の使用 OpsItems](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeOpsItems](#)」の「」を参照してください。

describe-parameters

次の例は、describe-parameters を使用する方法を説明しています。

AWS CLI

例 1: すべてのパラメータを一覧表示するには

次のdescribe-parameters例では、現在の AWS アカウントとリージョンのすべてのパラメータを一覧表示します。

```
aws ssm describe-parameters
```

出力:

```
{
  "Parameters": [
    {
      "Name": "MySecureStringParameter",
      "Type": "SecureString",
      "KeyId": "alias/aws/ssm",
      "LastModifiedDate": 1582155479.205,
      "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/Admin/Richard-Roe-Managed",
      "Description": "This is a SecureString parameter",
```

```
    "Version": 2,
    "Tier": "Advanced",
    "Policies": [
      {
        "PolicyText": "{\"Type\":\"Expiration\",\"Version\":\"1.0\",
\\Attributes\":{\"Timestamp\":\"2020-07-07T22:30:00Z\"}}",
        "PolicyType": "Expiration",
        "PolicyStatus": "Pending"
      },
      {
        "PolicyText": "{\"Type\":\"ExpirationNotification\",\"Version\":
\\1.0\",\\Attributes\":{\"Before\":\"12\",\"Unit\":\"Hours\"}}",
        "PolicyType": "ExpirationNotification",
        "PolicyStatus": "Pending"
      }
    ]
  },
  {
    "Name": "MyStringListParameter",
    "Type": "StringList",
    "LastModifiedDate": 1582154764.222,
    "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
    "Description": "This is a StringList parameter",
    "Version": 1,
    "Tier": "Standard",
    "Policies": []
  },
  {
    "Name": "MyStringParameter",
    "Type": "String",
    "LastModifiedDate": 1582154711.976,
    "LastModifiedUser": "arn:aws:iam::111222333444:user/Alejandro-Rosalez",
    "Description": "This is a String parameter",
    "Version": 1,
    "Tier": "Standard",
    "Policies": []
  },
  {
    "Name": "latestAmi",
    "Type": "String",
    "LastModifiedDate": 1580862415.521,
    "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/lambda-ssm-
role/Automation-UpdateSSM-Param",
    "Version": 3,
```

```
        "Tier": "Standard",
        "Policies": []
    }
]
}
```

例 2: 特定のメタデータに一致するすべてのパラメータを一覧表示するには

この `describe-parameters` の例は、フィルターに一致するすべてのパラメータを一覧表示します。

```
aws ssm describe-parameters --filters 「Key=Type,Values=StringList」
```

出力:

```
{
  "Parameters": [
    {
      "Name": "MyStringListParameter",
      "Type": "StringList",
      "LastModifiedDate": 1582154764.222,
      "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
      "Description": "This is a StringList parameter",
      "Version": 1,
      "Tier": "Standard",
      "Policies": []
    }
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Systems Manager のパラメータを検索する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeParameters](#)」の「」を参照してください。

describe-patch-baselines

次の例は、`describe-patch-baselines` を使用する方法を説明しています。

AWS CLI

例 1: すべてのパッチベースラインを一覧表示するには

次の `describe-patch-baselines` の例では、現在のリージョンにおけるアカウントのすべてのパッチベースラインに関する詳細情報を取得します。

```
aws ssm describe-patch-baselines
```

出力:

```
{
  "BaselineIdentities": [
    {
      "BaselineName": "AWS-SuseDefaultPatchBaseline",
      "DefaultBaseline": true,
      "BaselineDescription": "Default Patch Baseline for Suse Provided by
AWS.",
      "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0123fdb36e334a3b2",
      "OperatingSystem": "SUSE"
    },
    {
      "BaselineName": "AWS-DefaultPatchBaseline",
      "DefaultBaseline": false,
      "BaselineDescription": "Default Patch Baseline Provided by AWS.",
      "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed",
      "OperatingSystem": "WINDOWS"
    },
    ...
    {
      "BaselineName": "MyWindowsPatchBaseline",
      "DefaultBaseline": true,
      "BaselineDescription": "My patch baseline for EC2 instances for Windows
Server",
      "BaselineId": "pb-0ad00e0dd7EXAMPLE",
      "OperatingSystem": "WINDOWS"
    }
  ]
}
```

例 2: によって提供されるすべてのパッチベースラインを一覧表示するには AWS

次の `describe-patch-baselines` 例では、 が提供するすべてのパッチベースラインを一覧表示します AWS。


```
aws ssm describe-patch-baselines \  
  --filters "Key=OWNER,Values=[AWS]"
```

例 3: 所有しているすべてのパッチベースラインを一覧表示するには

次の describe-patch-baselines の例では、現在のリージョンにおけるアカウントで作成されたすべてのカスタムパッチベースラインを一覧表示します。

```
aws ssm describe-patch-baselines \  
  --filters "Key=OWNER,Values=[Self]"
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[事前定義されたパッチベースラインについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePatchBaselines](#)」の「」を参照してください。

describe-patch-group-state

次のコード例は、describe-patch-group-state を使用する方法を示しています。

AWS CLI

パッチグループの状態を取得するには

次の describe-patch-group-state の例では、パッチグループのパッチコンプライアンスの概要を取得します。

```
aws ssm describe-patch-group-state \  
  --patch-group "Production"
```

出力:

```
{  
  "Instances": 21,  
  "InstancesWithCriticalNonCompliantPatches": 1,  
  "InstancesWithFailedPatches": 2,  
  "InstancesWithInstalledOtherPatches": 3,  
  "InstancesWithInstalledPatches": 21,  
  "InstancesWithInstalledPendingRebootPatches": 2,
```

```
"InstancesWithInstalledRejectedPatches": 1,  
"InstancesWithMissingPatches": 3,  
"InstancesWithNotApplicablePatches": 4,  
"InstancesWithOtherNonCompliantPatches": 1,  
"InstancesWithSecurityNonCompliantPatches": 1,  
"InstancesWithUnreportedNotApplicablePatches": 2  
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「パッチグループについて」 <<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html>>_」および「[パッチコンプライアンス状態値について](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePatchGroupState](#)」の「」を参照してください。

describe-patch-groups

次の例は、describe-patch-groups を使用する方法を説明しています。

AWS CLI

パッチグループの登録を表示するには

次の describe-patch-groups の例では、パッチグループの登録を一覧表示します。

```
aws ssm describe-patch-groups
```

出力:

```
{  
  "Mappings": [  
    {  
      "PatchGroup": "Production",  
      "BaselineIdentity": {  
        "BaselineId": "pb-0123456789abcdef0",  
        "BaselineName": "ProdPatching",  
        "OperatingSystem": "WINDOWS",  
        "BaselineDescription": "Patches for Production",  
        "DefaultBaseline": false  
      }  
    },  
    {
```

```
    "PatchGroup": "Development",
    "BaselineIdentity": {
      "BaselineId": "pb-0713accee01234567",
      "BaselineName": "DevPatching",
      "OperatingSystem": "WINDOWS",
      "BaselineDescription": "Patches for Development",
      "DefaultBaseline": true
    }
  },
  ...
]
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「パッチグループ <<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>>__の作成」および「[パッチベースラインへのパッチグループの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribePatchGroups](#)」の「」を参照してください。

describe-patch-properties

次の例は、describe-patch-properties を使用する方法を説明しています。

AWS CLI

Amazon Linux パッチの可用性を一覧表示するには

次のdescribe-patch-properties例は、AWS アカウントでパッチが利用可能な Amazon Linux 製品のリストを示しています。

```
aws ssm describe-patch-properties \
  --operating-system AMAZON_LINUX \
  --property PRODUCT
```

出力:

```
{
  "Properties": [
    {
      "Name": "AmazonLinux2012.03"
    }
  ],
```

```
{
  "Name": "AmazonLinux2012.09"
},
{
  "Name": "AmazonLinux2013.03"
},
{
  "Name": "AmazonLinux2013.09"
},
{
  "Name": "AmazonLinux2014.03"
},
{
  "Name": "AmazonLinux2014.09"
},
{
  "Name": "AmazonLinux2015.03"
},
{
  "Name": "AmazonLinux2015.09"
},
{
  "Name": "AmazonLinux2016.03"
},
{
  "Name": "AmazonLinux2016.09"
},
{
  "Name": "AmazonLinux2017.03"
},
{
  "Name": "AmazonLinux2017.09"
},
{
  "Name": "AmazonLinux2018.03"
}
]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パッチベースラインについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribePatchProperties](#)」の「」を参照してください。

describe-sessions

次のコード例は、describe-sessions を使用する方法を示しています。

AWS CLI

例 1: すべてのアクティブな Session Manager セッションを一覧表示するには

この例では、指定したユーザーが過去 30 日間に最後に作成したアクティブなセッション (接続されたセッションと切断されたセッションの両方) のリスト describe-sessions を取得します。このコマンドは、Session Manager を使用して開始されたターゲットへの接続の結果のみを返します。リモートデスクトップ接続や など、他の方法で行われた接続はリストされません SSH。

```
aws ssm describe-sessions \  
  --state "Active" \  
  --filters "key=Owner,value=arn:aws:sts::123456789012:assumed-role/Administrator/  
Shirley-Rodriguez"
```

出力:

```
{  
  "Sessions": [  
    {  
      "SessionId": "John-07a16060613c408b5",  
      "Target": "i-1234567890abcdef0",  
      "Status": "Connected",  
      "StartDate": 1550676938.352,  
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-  
Rodriguez",  
      "OutputUrl": {}  
    },  
    {  
      "SessionId": "John-01edf534b8b56e8eb",  
      "Target": "i-9876543210abcdef0",  
      "Status": "Connected",  
      "StartDate": 1550676842.194,  
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Shirley-  
Rodriguez",  
      "OutputUrl": {}  
    }  
  ]  
}
```

例 2: 終了したすべての Session Manager セッションを一覧表示するには

この例では、すべてのユーザーについて、過去 30 日間の最後に終了したセッションのリスト `describe-sessions` を取得します。

```
aws ssm describe-sessions \  
  --state "History"
```

出力:

```
{  
  "Sessions": [  
    {  
      "SessionId": "Mary-Major-0022b1eb2b0d9e3bd",  
      "Target": "i-1234567890abcdef0",  
      "Status": "Terminated",  
      "StartDate": 1550520701.256,  
      "EndDate": 1550521931.563,  
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Mary-  
Major"  
    },  
    {  
      "SessionId": "Jane-Roe-0db53f487931ed9d4",  
      "Target": "i-9876543210abcdef0",  
      "Status": "Terminated",  
      "StartDate": 1550161369.149,  
      "EndDate": 1550162580.329,  
      "Owner": "arn:aws:sts::123456789012:assumed-role/Administrator/Jane-Roe"  
    },  
    ...  
  ],  
  "NextToken": "--token string truncated--"  
}
```

詳細については、「Systems Manager ユーザーガイド」の「[セッション履歴の表示](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeSessions](#)」の「」を参照してください。

disassociate-ops-item-related-item

次のコード例は、disassociate-ops-item-related-item を使用する方法を示しています。

AWS CLI

関連項目の関連付けを削除するには

次のdisassociate-ops-item-related-item例では、OpsItem と関連項目の間の関連付けを削除します。

```
aws ssm disassociate-ops-item-related-item \  
  --ops-item-id "oi-f99f2EXAMPLE" \  
  --association-id "e2036148-cccb-490e-ac2a-390e5EXAMPLE"
```

このコマンドでは何も出力されません。

詳細については、AWS Systems [Manager ユーザーガイドの「」の「Incident Manager インシデントの使用 OpsCenter」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateOpsItemRelatedItem](#)」の「」を参照してください。

get-automation-execution

次のコード例は、get-automation-execution を使用する方法を示しています。

AWS CLI

オートメーションの実行に関する詳細情報を表示するには

次の get-automation-execution の例では、オートメーションの実行に関する詳細情報を表示します。

```
aws ssm get-automation-execution \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

出力:

```
{  
  "AutomationExecution": {  
    "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",  
    "DocumentName": "AWS-StartEC2Instance",
```

```
"DocumentVersion": "1",
"ExecutionStartTime": 1583737233.748,
"ExecutionEndTime": 1583737234.719,
"AutomationExecutionStatus": "Success",
"StepExecutions": [
  {
    "StepName": "startInstances",
    "Action": "aws:changeInstanceState",
    "ExecutionStartTime": 1583737234.134,
    "ExecutionEndTime": 1583737234.672,
    "StepStatus": "Success",
    "Inputs": {
      "DesiredState": "\"running\"",
      "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
    },
    "Outputs": {
      "InstanceStates": [
        "running"
      ]
    },
    "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
    "OverriddenParameters": {}
  }
],
"StepExecutionsTruncated": false,
"Parameters": {
  "AutomationAssumeRole": [
    ""
  ],
  "InstanceId": [
    "i-0cb99161f6EXAMPLE"
  ]
},
"Outputs": {},
"Mode": "Auto",
"ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
"Targets": [],
"ResolvedTargets": {
  "ParameterValues": [],
  "Truncated": false
}
}
```



```
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の [「チュートリアル: Linux AMI \(AWS CLI\) にパッチを適用する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetAutomationExecution](#)」の「」を参照してください。

get-calendar-state

次の例は、get-calendar-state を使用する方法を説明しています。

AWS CLI

例 1: 変更カレンダーの現在の状態を取得するには

この例では、現在のカレンダーの状態 get-calendar-state を返します。この例では時刻を指定していないため、カレンダーの現在の状態が報告されます。

```
aws ssm get-calendar-state \  
  --calendar-names "MyCalendar"
```

出力:

```
{  
  "State": "OPEN",  
  "AtTime": "2020-02-19T22:28:51Z",  
  "NextTransitionTime": "2020-02-24T21:15:19Z"  
}
```

例 2: 指定された時刻に変更カレンダーの状態を取得するには

この例では、指定された時刻のカレンダーの状態 get-calendar-state を返します。

```
aws ssm get-calendar-state \  
  --calendar-names "MyCalendar" \  
  --at-time "2020-07-19T21:15:19Z"
```

出力:

```
{
```

```
"State": "CLOSED",  
"AtTime": "2020-07-19T21:15:19Z"  
}
```

詳細については、AWS Systems Manager ユーザーガイドの [「変更カレンダーの状態を取得する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetCalendarState](#)」の「」を参照してください。

get-command-invocation

次のコード例は、get-command-invocation を使用する方法を示しています。

AWS CLI

コマンド呼び出しの詳細情報を表示するには

次の get-command-invocation の例では、指定されたインスタンスにおける指定されたコマンドのすべての呼び出しを一覧表示します。

```
aws ssm get-command-invocation \  
  --command-id "ef7fd8-9b57-4151-a15c-db9a12345678" \  
  --instance-id "i-1234567890abcdef0"
```

出力:

```
{  
  "CommandId": "ef7fd8-9b57-4151-a15c-db9a12345678",  
  "InstanceId": "i-1234567890abcdef0",  
  "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",  
  "DocumentName": "AWS-UpdateSSMAgent",  
  "DocumentVersion": "",  
  "PluginName": "aws:updateSsmAgent",  
  "ResponseCode": 0,  
  "ExecutionStartDateTime": "2020-02-19T18:18:03.419Z",  
  "ExecutionElapsedTime": "PT0.091S",  
  "ExecutionEndDateTime": "2020-02-19T18:18:03.419Z",  
  "Status": "Success",  
  "StatusDetails": "Success",  
}
```

```
"StandardOutputContent": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
update skipped\n",
"StandardOutputUrl": "",
"StandardErrorContent": "",
"StandardErrorUrl": "",
"CloudWatchOutputConfig": {
  "CloudWatchLogGroupName": "",
  "CloudWatchOutputEnabled": false
}
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[コマンドのステータスについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetCommandInvocation](#)」の「」を参照してください。

get-connection-status

次のコード例は、get-connection-status を使用する方法を示しています。

AWS CLI

マネージドインスタンスの接続ステータスを表示するには

この get-connection-status の例では、指定されたマネージドインスタンスの接続ステータスを返します。

```
aws ssm get-connection-status \
  --target i-1234567890abcdef0
```

出力:

```
{
  "Target": "i-1234567890abcdef0",
  "Status": "connected"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetConnectionStatus](#)」の「」を参照してください。

get-default-patch-baseline

次の例は、get-default-patch-baseline を使用する方法を説明しています。

AWS CLI

例 1: デフォルトの Windows パッチベースラインを表示するには

次の get-default-patch-baseline の例では、Windows Server のデフォルトのパッチベースラインの詳細を取得します。

```
aws ssm get-default-patch-baseline
```

出力:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

例 2: デフォルトの Amazon Linux パッチベースラインを表示するには

次の get-default-patch-baseline の例では、Amazon Linux のデフォルトのパッチベースラインの詳細を取得します。

```
aws ssm get-default-patch-baseline \
  --operating-system AMAZON_LINUX
```

出力:

```
{
  "BaselineId": "pb-047c6eb9c8fc12345",
  "OperatingSystem": "AMAZON_LINUX"
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「事前定義およびカスタムパッチベースラインについて」 <<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>>__ および [既存のパッチベースラインをデフォルトとして設定する](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDefaultPatchBaseline](#)」の「」を参照してください。

get-deployable-patch-snapshot-for-instance

次の例は、get-deployable-patch-snapshot-for-instance を使用する方法を説明しています。

AWS CLI

インスタンスが使用するパッチベースラインの現在のスナップショットを取得するには

次の get-deployable-patch-snapshot-for-instance の例では、インスタンスが使用する指定されたパッチベースラインの現在のスナップショットの詳細を取得します。このコマンドは、インスタンス認証情報を使用してインスタンスから実行する必要があります。インスタンス認証情報が使用されるようにするため、aws configure を実行し、インスタンスのリージョンのみを指定します。Access Key および Secret Key フィールドは空のままにします。

ヒント: uuidgen を使用して snapshot-id を生成します。

```
aws ssm get-deployable-patch-snapshot-for-instance \
  --instance-id "i-1234567890abcdef0" \
  --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

出力:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
  "Product": "AmazonLinux2018.03",
  "SnapshotDownloadUrl": "https://patch-baseline-snapshot-us-east-1.s3.amazonaws.com/ed85194ef27214f5984f28b4d664d14f7313568fea7d4b6ac6c10ad1f729d7e7-773304212436/AMAZON_LINUX-521c3536-930c-4aa9-950e-01234567abcd?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190215T164031Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-Credential=AKIAJ5C56P35AEBRX2QQ%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Parameter name: Snapshot ID](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDeployablePatchSnapshotForInstance](#)」の「」を参照してください。

get-document

次のコード例は、get-document を使用する方法を示しています。

AWS CLI

ドキュメントコンテンツを取得するには

次の get-document の例では、Systems Manager ドキュメントのコンテンツを表示します。

```
aws ssm get-document \
  --name "AWS-RunShellScript"
```

出力:

```
{
  "Name": "AWS-RunShellScript",
  "DocumentVersion": "1",
  "Status": "Active",
  "Content": "{\n  \"schemaVersion\": \"1.2\",\n  \"description\": \"Run a\nshell script or specify the commands to run.\",\n  \"parameters\": {\n    \"commands\": {\n      \"type\": \"StringList\",\n      \"description\": \"(Required) Specify a shell script or a command to run.\",\n      \"minItems\": 1,\n      \"displayType\": \"textarea\"\n    },\n    \"workingDirectory\": {\n      \"type\": \"String\",\n      \"default\": \"\",\n      \"description\": \"(Optional) The path to the working\ndirectory on your instance.\",\n      \"maxChars\": 4096\n    },\n    \"executionTimeout\": {\n      \"type\": \"String\",\n      \"default\": \"3600\",\n      \"description\": \"(Optional) The time in seconds for a\ncommand to complete before it is considered to have failed. Default is 3600 (1\nhour). Maximum is 172800 (48 hours).\",\n      \"allowedPattern\": \"([1-9][0-9]{0,4})|(1[0-6][0-9]{4})|(17[0-1][0-9]{3})|(172[0-7][0-9]{2})|(172800)\"\n    }\n  },\n  \"runtimeConfig\": {\n    \"aws:runShellScript\": {\n      \"properties\": [\n        {\n          \"id\":\n\"0.aws:runShellScript\",\n          \"runCommand\": \"{{ commands }}\",\n          \"workingDirectory\": \"{{ workingDirectory }}\",\n          \"timeoutSeconds\": \"{{ executionTimeout }}\"\n        }\n      ]\n    }\n  },\n  \"DocumentType\": \"Command\",\n  \"DocumentFormat\": \"JSON\"
```

```
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[AWS Systems Manager ドキュメント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDocument](#)」の「」を参照してください。

get-inventory-schema

次の例は、get-inventory-schema を使用する方法を説明しています。

AWS CLI

インベントリスキーマを表示するには

この例では、アカウントのインベントリタイプ名のリストを返します。

コマンド:

```
aws ssm get-inventory-schema
```

出力:

```
{
  "Schemas": [
    {
      "TypeName": "AWS:AWSComponent",
      "Version": "1.0",
      "Attributes": [
        {
          "Name": "Name",
          "DataType": "STRING"
        },
        {
          "Name": "ApplicationType",
          "DataType": "STRING"
        },
        {
          "Name": "Publisher",
          "DataType": "STRING"
        },
        {
```

```
        "Name": "Version",
        "DataType": "STRING"
    },
    {
        "Name": "InstalledTime",
        "DataType": "STRING"
    },
    {
        "Name": "Architecture",
        "DataType": "STRING"
    },
    {
        "Name": "URL",
        "DataType": "STRING"
    }
]
},
...
],
"NextToken": "--token string truncated--"
}
```

特定のインベントリタイプのインベントリスキーマを表示するには

この例では、AWS : AWSコンポーネントインベントリタイプのインベントリスキーマを返します。

コマンド:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInventorySchema](#)」の「」を参照してください。

get-inventory

次の例は、get-inventory を使用する方法を説明しています。

AWS CLI

インベントリを表示するには

この例では、インベントリのカスタムメタデータを取得します。

コマンド:

```
aws ssm get-inventory
```

出力:

```
{
  "Entities": [
    {
      "Data": {
        "AWS:InstanceInformation": {
          "Content": [
            {
              "ComputerName": "ip-172-31-44-222.us-
west-2.compute.internal",
              "InstanceId": "i-0cb2b964d3e14fd9f",
              "IpAddress": "172.31.44.222",
              "AgentType": "amazon-ssm-agent",
              "ResourceType": "EC2Instance",
              "AgentVersion": "2.0.672.0",
              "PlatformVersion": "2016.09",
              "PlatformName": "Amazon Linux AMI",
              "PlatformType": "Linux"
            }
          ],
          "TypeName": "AWS:InstanceInformation",
          "SchemaVersion": "1.0",
          "CaptureTime": "2017-02-20T18:03:58Z"
        }
      },
      "Id": "i-0cb2b964d3e14fd9f"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetInventory](#)」の「」を参照してください。

get-maintenance-window-execution-task-invocation

次のコード例は、`get-maintenance-window-execution-task-invocation` を使用する方法を示しています。

AWS CLI

メンテナンスウィンドウタスクの呼び出しに関する情報を取得するには

次のget-maintenance-window-execution-task-invocation例は、指定されたメンテナンスウィンドウの実行の一部である、指定されたタスク呼び出しに関する情報を示しています。

```
aws ssm get-maintenance-window-execution-task-invocation \
  --window-execution-id "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE" \
  --task-id "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE" \
  --invocation-id "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE"
```

出力:

```
{
  "Status": "SUCCESS",
  "Parameters": "{\"comment\":\"\", \"documentName\":\"AWS-RunPowerShellScript\", \"instanceIds\": [\"i-1234567890EXAMPLE\"], \"maxConcurrency\": \"1\", \"maxErrors\": \"1\", \"parameters\": {\"executionTimeout\": [\"3600\"], \"workingDirectory\": [\"\"], \"commands\": [\"echo Hello\"]}, \"timeoutSeconds\": 600}\",
  "ExecutionId": "03b6baa0-5460-4e15-83f2-ea685EXAMPLE",
  "InvocationId": "a5273e2c-d2c6-4880-b3e1-5e550EXAMPLE",
  "StartTime": 1549998326.421,
  "TaskType": "RUN_COMMAND",
  "EndTime": 1550001931.784,
  "WindowExecutionId": "bc494bfa-e63b-49f6-8ad1-aa9f2EXAMPLE",
  "StatusDetails": "Failed",
  "TaskExecutionId": "96f2ad59-97e3-461d-a63d-40c8aEXAMPLE"
}
```

詳細については、AWS「Systems Manager ユーザーガイド」の[「タスクとタスク実行に関する情報の表示 \(AWS CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetMaintenanceWindowExecutionTaskInvocation](#)」の「」を参照してください。

get-maintenance-window-execution-task

次の例は、get-maintenance-window-execution-task を使用する方法を説明しています。

AWS CLI

メンテナンスウィンドウのタスクの実行に関する情報を取得するには

次の `get-maintenance-window-execution-task` の例では、指定されたメンテナンスウィンドウの実行の一部であるタスクに関する情報を一覧表示します。

```
aws ssm get-maintenance-window-execution-task \
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
  --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

出力:

```
{
  "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
  "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE",
  "TaskArn": "AWS-RunPatchBaseline",
  "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
  ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
  "Type": "RUN_COMMAND",
  "TaskParameters": [
    {
      "BaselineOverride": {
        "Values": [
          ""
        ]
      },
      "InstallOverrideList": {
        "Values": [
          ""
        ]
      },
      "Operation": {
        "Values": [
          "Scan"
        ]
      },
      "RebootOption": {
        "Values": [
          "RebootIfNeeded"
        ]
      },
      "SnapshotId": {
```

```
        "Values": [
            "{ aws:ORCHESTRATION_ID }"
        ],
        "aws:InstanceId": {
            "Values": [
                "i-02573cafcfEXAMPLE",
                "i-0471e04240EXAMPLE",
                "i-07782c72faEXAMPLE"
            ]
        }
    ],
    "Priority": 1,
    "MaxConcurrency": "1",
    "MaxErrors": "3",
    "Status": "SUCCESS",
    "StartTime": "2021-08-04T11:45:35.088000-07:00",
    "EndTime": "2021-08-04T11:53:09.079000-07:00"
}
```

詳細については、AWS Systems Manager ユーザーガイドの [「タスクとタスク実行 \(AWS CLI\) に関する情報を表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetMaintenanceWindowExecutionTask](#)」の「」を参照してください。

get-maintenance-window-execution

次の例は、get-maintenance-window-execution を使用する方法を説明しています。

AWS CLI

メンテナンスウィンドウのタスクの実行に関する情報を取得するには

次の get-maintenance-window-execution の例では、指定されたメンテナンスウィンドウの一部として実行されるタスクに関する情報を一覧表示します。

```
aws ssm get-maintenance-window-execution \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

出力:

```
{
  "Status": "SUCCESS",
  "TaskIds": [
    "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
  ],
  "StartTime": 1487692834.595,
  "EndTime": 1487692835.051,
  "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
}
```

詳細については、AWS「Systems Manager ユーザーガイド」の[「タスクとタスク実行に関する情報の表示 \(AWS CLI\)」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetMaintenanceWindowExecution](#)」の「」を参照してください。

get-maintenance-window-task

次のコード例は、get-maintenance-window-task を使用する方法を示しています。

AWS CLI

メンテナンスウィンドウタスクに関する情報を取得するには

次のget-maintenance-window-task例では、指定されたメンテナンスウィンドウタスクの詳細を取得します。

```
aws ssm get-maintenance-window-task \
  --window-id mw-0c5ed765acEXAMPLE \
  --window-task-id 0e842a8d-2d44-4886-bb62-af8dcEXAMPLE
```

出力:

```
{
  "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
  ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
  "MaxErrors": "1",
  "TaskArn": "AWS-RunPowerShellScript",
  "MaxConcurrency": "1",
  "WindowTaskId": "0e842a8d-2d44-4886-bb62-af8dcEXAMPLE",
  "TaskParameters": {},
  "Priority": 1,
```

```
"TaskInvocationParameters": {
  "RunCommand": {
    "Comment": "",
    "TimeoutSeconds": 600,
    "Parameters": {
      "commands": [
        "echo Hello"
      ],
      "executionTimeout": [
        "3600"
      ],
      "workingDirectory": [
        ""
      ]
    }
  },
  "WindowId": "mw-0c5ed765acEXAMPLE",
  "TaskType": "RUN_COMMAND",
  "Targets": [
    {
      "Values": [
        "84c818da-b619-4d3d-9651-946f3EXAMPLE"
      ],
      "Key": "WindowTargetIds"
    }
  ],
  "Name": "ExampleTask"
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の [「メンテナンスウィンドウ \(AWS CLI\) に関する情報を表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetMaintenanceWindowTask](#)」の「」を参照してください。

get-maintenance-window

次の例は、get-maintenance-window を使用する方法を説明しています。

AWS CLI

メンテナンスウィンドウに関する情報を取得するには

次の `get-maintenance-window` の例では、指定されたメンテナンスウィンドウの詳細情報を取得します。

```
aws ssm get-maintenance-window \  
  --window-id "mw-03eb9db428EXAMPLE"
```

出力:

```
{  
  "AllowUnassociatedTargets": true,  
  "CreateDate": 1515006912.957,  
  "Cutoff": 1,  
  "Duration": 6,  
  "Enabled": true,  
  "ModifiedDate": 2020-01-01T10:04:04.099Z,  
  "Name": "My-Maintenance-Window",  
  "Schedule": "rate(3 days)",  
  "WindowId": "mw-03eb9db428EXAMPLE",  
  "NextExecutionTime": "2020-02-25T00:08:15.099Z"  
}
```

詳細については、AWS Systems Manager ユーザーガイドの [「メンテナンスウィンドウ \(AWS CLI\) に関する情報を表示する」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetMaintenanceWindow](#)」の「」を参照してください。

get-ops-item

次の例は、`get-ops-item` を使用する方法を説明しています。

AWS CLI

に関する情報を表示するには `OpsItem`

次の `get-ops-item` 例では、指定された の詳細を表示します `OpsItem`。

```
aws ssm get-ops-item \  
  --ops-item-id oi-0b725EXAMPLE
```

出力:

```

{
  "OpsItem": {
    "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
    "CreatedTime": "2019-12-04T15:52:16.793000-08:00",
    "Description": "CloudWatch Event Rule SSMOpsItems-EC2-instance-terminated
was triggered. Your EC2 instance has terminated. See below for more details.",
    "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
    "LastModifiedTime": "2019-12-04T15:52:16.793000-08:00",
    "Notifications": [],
    "RelatedOpsItems": [],
    "Status": "Open",
    "OpsItemId": "oi-0b725EXAMPLE",
    "Title": "EC2 instance terminated",
    "Source": "EC2",
    "OperationalData": {
      "/aws/automations": {
        "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
\"automationId\": \"AWS>CreateManagedWindowsInstance\" }, { \"automationType\":
\"AWS:SSM:Automation\", \"automationId\": \"AWS>CreateManagedLinuxInstance\" } ]",
        "Type": "SearchableString"
      },
      "/aws/dedup": {
        "Value": "{ \"dedupString\": \"SSMOpsItems-EC2-instance-terminated
\" }",
        "Type": "SearchableString"
      },
      "/aws/resources": {
        "Value": "[ { \"arn\": \"arn:aws:ec2:us-east-2:111222333444:instance/
i-05adec7e97EXAMPLE\" } ]",
        "Type": "SearchableString"
      },
      "event-time": {
        "Value": "2019-12-04T23:52:16Z",
        "Type": "String"
      },
      "instance-state": {
        "Value": "terminated",
        "Type": "String"
      }
    }
  },
  "Category": "Availability",

```



```
    "Severity": "4"
  }
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「[の使用 OpsItems](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOpsItem](#)」の「」を参照してください。

get-ops-summary

次の例は、get-ops-summary を使用する方法を説明しています。

AWS CLI

すべての の概要を表示するには OpsItems

次のget-ops-summary例では、AWS アカウント OpsItems 内のすべての の概要を表示します。

```
aws ssm get-ops-summary
```

出力:

```
{
  "Entities": [
    {
      "Id": "oi-4309fEXAMPLE",
      "Data": {
        "AWS:OpsItem": {
          "CaptureTime": "2020-02-26T18:58:32.918Z",
          "Content": [
            {
              "AccountId": "111222333444",
              "Category": "Availability",
              "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
              "CreatedTime": "2020-02-26T19:10:44.149Z",
              "Description": "CloudWatch Event Rule SSM0psItems-EC2-instance-terminated was triggered. Your EC2 instance has terminated. See below for more details.",
            }
          ]
        }
      }
    }
  ]
}
```

```

        "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
        "LastModifiedTime": "2020-02-26T19:10:44.149Z",
        "Notifications": "",
        "OperationalData": "{\"/aws/automations\":
{\"type\": \"SearchableString\", \"value\": \"[ { \\\"automationType\\\": \\
\\\"AWS:SSM:Automation\\\", \\\"automationId\\\": \\\"AWS-CreateManagedWindowsInstance
\\\" }, { \\\"automationType\\\": \\\"AWS:SSM:Automation\\\", \\\"automationId
\\\": \\\"AWS-CreateManagedLinuxInstance\\\" } ]\", \"/aws/resources\":
{\"type\": \"SearchableString\", \"value\": \"[{\\\"arn\\\": \\\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0acbd0800fEXAMPLE\\\"]\", \"/aws/dedup\": {\"type\":
\"SearchableString\", \"value\": \"{\\\"dedupString\\\": \\\"SSMOpsItems-EC2-instance-
terminated\\\"}\"}}",
        "OpsItemId": "oi-4309fEXAMPLE",
        "RelatedItems": "",
        "Severity": "3",
        "Source": "EC2",
        "Status": "Open",
        "Title": "EC2 instance terminated"
    }
  ]
}
},
{
  "Id": "oi-bb2a0e6a4541",
  "Data": {
    "AWS:OpsItem": {
      "CaptureTime": "2019-11-26T19:20:06.161Z",
      "Content": [
        {
          "AccountId": "111222333444",
          "Category": "Availability",
          "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
          "CreatedTime": "2019-11-26T20:00:07.237Z",
          "Description": "CloudWatch Event Rule SSMOpsItems-SSM-
maintenance-window-execution-failed was triggered. Your SSM Maintenance Window
execution has failed. See below for more details.",
          "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
          "LastModifiedTime": "2019-11-26T20:00:07.237Z",
          "Notifications": "",

```

```

      "OperationalData": "{\"/aws/resources\":{\"type
\\":\"SearchableString\", \"value\": \"[\\\"arn\\\":\\\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-0e83ba440dEXAMPLE\\\"]\", \"/aws/dedup\":
{\"type\": \"SearchableString\", \"value\": \"[\\\"dedupString\\\":\\\"SSM0psItems-SSM-
maintenance-window-execution-failed\\\"]\"}},
      "OpsItemId": "oi-bb2a0EXAMPLE",
      "RelatedItems": "",
      "Severity": "3",
      "Source": "SSM",
      "Status": "Open",
      "Title": "SSM Maintenance Window execution failed"
    }
  ]
}

```

詳細については、AWS「Systems Manager ユーザーガイド」の「[の使用 OpsItems](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[GetOpsSummary](#)」の「」を参照してください。

get-parameter-history

次の例は、get-parameter-history を使用する方法を説明しています。

AWS CLI

パラメータの値の履歴を取得するには

次の get-parameter-history の例では、指定されたパラメータの変更履歴を、値を含めて一覧表示します。

```
aws ssm get-parameter-history \
  --name "MyStringParameter"
```

出力:

```
{
  "Parameters": [
```

```
{
  "Name": "MyStringParameter",
  "Type": "String",
  "LastModifiedDate": 1582154711.976,
  "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
  "Description": "This is the first version of my String parameter",
  "Value": "Veni",
  "Version": 1,
  "Labels": [],
  "Tier": "Standard",
  "Policies": []
},
{
  "Name": "MyStringParameter",
  "Type": "String",
  "LastModifiedDate": 1582156093.471,
  "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
  "Description": "This is the second version of my String parameter",
  "Value": "Vidi",
  "Version": 2,
  "Labels": [],
  "Tier": "Standard",
  "Policies": []
},
{
  "Name": "MyStringParameter",
  "Type": "String",
  "LastModifiedDate": 1582156117.545,
  "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
  "Description": "This is the third version of my String parameter",
  "Value": "Vici",
  "Version": 3,
  "Labels": [],
  "Tier": "Standard",
  "Policies": []
}
]
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パラメータバージョンの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParameterHistory](#)」の「」を参照してください。

get-parameter

次のコード例は、get-parameter を使用する方法を示しています。

AWS CLI

例 1: パラメータの値を表示する

次の get-parameter 例は、指定された単一のパラメータの値を一覧表示します。

```
aws ssm get-parameter \  
  --name "MyStringParameter"
```

出力:

```
{  
  "Parameter": {  
    "Name": "MyStringParameter",  
    "Type": "String",  
    "Value": "Veni",  
    "Version": 1,  
    "LastModifiedDate": 1530018761.888,  
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"  
    "DataType": "text"  
  }  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Parameter Store の使用](#)」を参照してください。

例 2: SecureString パラメータの値を復号するには

次の get-parameter 例は、指定された SecureString パラメータの値を複合化します。

```
aws ssm get-parameter \  
  --name "MySecureStringParameter" \  
  --with-decryption
```

出力:

```
{  
  "Parameter": {  
    "Name": "MySecureStringParameter",
```

```
    "Type": "SecureString",
    "Value": "16679b88-310b-4895-a943-e0764EXAMPLE",
    "Version": 2,
    "LastModifiedDate": 1582155479.205,
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MySecureStringParameter"
    "DataType": "text"
  }
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Parameter Store の使用](#)」を参照してください。

例 3: ラベルを使用してパラメータの値を表示する

次の `get-parameter` 例は、特定のラベルを使用して、指定された単一のパラメータの値を一覧表示します。

```
aws ssm get-parameter \
  --name "MyParameter:label"
```

出力:

```
{
  "Parameter": {
    "Name": "MyParameter",
    "Type": "String",
    "Value": "parameter version 2",
    "Version": 2,
    "Selector": ":label",
    "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
    "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
    "DataType": "text"
  }
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パラメータラベルの操作](#)」を参照してください。

例 4: バージョンを使用してパラメータの値を表示する

次の `get-parameter` 例は、指定された単一のパラメータバージョンの値を一覧表示します。

```
aws ssm get-parameter \  
  --name "MyParameter:2"
```

出力:

```
{  
  "Parameter": {  
    "Name": "MyParameter",  
    "Type": "String",  
    "Value": "parameter version 2",  
    "Version": 2,  
    "Selector": ":2",  
    "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",  
    "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",  
    "DataType": "text"  
  }  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パラメータラベルの操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParameter](#)」の「」を参照してください。

get-parameters-by-path

次の例は、get-parameters-by-path を使用する方法を説明しています。

AWS CLI

特定のパスのパラメータを一覧表示するには

次のget-parameters-by-path例では、指定された階層内のパラメータを一覧表示します。

```
aws ssm get-parameters-by-path \  
  --path "/site/newyork/department/"
```

出力:

```
{  
  "Parameters": [  
    {
```

```

        "Name": "/site/newyork/department/marketing",
        "Type": "String",
        "Value": "Floor 2",
        "Version": 1,
        "LastModifiedDate": 1530018761.888,
        "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/
department/marketing"
    },
    {
        "Name": "/site/newyork/department/infotech",
        "Type": "String",
        "Value": "Floor 3",
        "Version": 1,
        "LastModifiedDate": 1530018823.429,
        "ARN": "arn:aws:ssm:us-east-1:111222333444:parameter/site/newyork/
department/infotech"
    },
    ...
]
}

```

詳細については、AWS Systems Manager ユーザーガイドの「[パラメータ階層の操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParametersByPath](#)」の「」を参照してください。

get-parameters

次のコード例は、get-parameters を使用する方法を示しています。

AWS CLI

例 1: パラメータの値を一覧表示するには

次の get-parameters の例では、指定した 3 つのパラメータの値を一覧表示します。

```
aws ssm get-parameters \
  --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

出力:

```
{
```



```

"Parameters": [
  {
    "Name": "MyStringListParameter",
    "Type": "StringList",
    "Value": "alpha,beta,gamma",
    "Version": 1,
    "LastModifiedDate": 1582154764.222,
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringListParameter"
    "DataType": "text"
  },
  {
    "Name": "MyStringParameter",
    "Type": "String",
    "Value": "Vici",
    "Version": 3,
    "LastModifiedDate": 1582156117.545,
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
    "DataType": "text"
  }
],
"InvalidParameters": [
  "MyInvalidParameterName"
]
}

```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Parameter Store の使用](#)」を参照してください。

例 2: ``--query`` オプションを使用して複数のパラメータの名前と値を一覧表示するには次の `get-parameters` の例では、指定したパラメータの名前と値を一覧表示します。

```

aws ssm get-parameters \
  --names MyStringParameter MyStringListParameter \
  --query "Parameters[*].{Name:Name,Value:Value}"

```

出力:

```

[
  {
    "Name": "MyStringListParameter",
    "Value": "alpha,beta,gamma"
  }
]

```

```
  },
  {
    "Name": "MyStringParameter",
    "Value": "Vidi"
  }
]
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Parameter Store の使用](#)」を参照してください。

例 3: ラベルを使用してパラメータの値を表示する

次の `get-parameter` 例は、特定のラベルを使用して、指定された単一のパラメータの値を一覧表示します。

```
aws ssm get-parameter \
  --name "MyParameter:label"
```

出力:

```
{
  "Parameters": [
    {
      "Name": "MyLabelParameter",
      "Type": "String",
      "Value": "parameter by label",
      "Version": 1,
      "Selector": ":label",
      "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
      "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
      "DataType": "text"
    },
    {
      "Name": "MyVersionParameter",
      "Type": "String",
      "Value": "parameter by version",
      "Version": 2,
      "Selector": ":2",
      "LastModifiedDate": "2021-03-24T16:20:28.236000-07:00",
      "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/unlabel-param",
      "DataType": "text"
    }
  ],
}
```

```
"InvalidParameters": []
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パラメータラベルの操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetParameters](#)」の「」を参照してください。

get-patch-baseline-for-patch-group

次の例は、get-patch-baseline-for-patch-group を使用する方法を説明しています。

AWS CLI

パッチグループのパッチベースラインを表示するには

次の get-patch-baseline-for-patch-group の例では、指定されたパッチグループのパッチベースラインに関する詳細を取得します。

```
aws ssm get-patch-baseline-for-patch-group \
  --patch-group "DEV"
```

出力:

```
{
  "PatchGroup": "DEV",
  "BaselineId": "pb-0123456789abcdef0",
  "OperatingSystem": "WINDOWS"
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「パッチグループ <<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>>__の作成」および「[パッチベースラインへのパッチグループの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPatchBaselineForPatchGroup](#)」の「」を参照してください。

get-patch-baseline

次の例は、get-patch-baseline を使用する方法を説明しています。

AWS CLI

パッチベースラインを表示するには

次の `get-patch-baseline` の例では、指定されたパッチベースラインの詳細を取得します。

```
aws ssm get-patch-baseline \  
  --baseline-id "pb-0123456789abcdef0"
```

出力:

```
{  
  "BaselineId": "pb-0123456789abcdef0",  
  "Name": "WindowsPatching",  
  "OperatingSystem": "WINDOWS",  
  "GlobalFilters": {  
    "PatchFilters": []  
  },  
  "ApprovalRules": {  
    "PatchRules": [  
      {  
        "PatchFilterGroup": {  
          "PatchFilters": [  
            {  
              "Key": "PRODUCT",  
              "Values": [  
                "WindowsServer2016"  
              ]  
            }  
          ]  
        },  
        "ComplianceLevel": "CRITICAL",  
        "ApproveAfterDays": 0,  
        "EnableNonSecurity": false  
      }  
    ]  
  },  
  "ApprovedPatches": [],  
  "ApprovedPatchesComplianceLevel": "UNSPECIFIED",  
  "ApprovedPatchesEnableNonSecurity": false,  
  "RejectedPatches": [],  
  "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",  
  "PatchGroups": [  
    {  
      "PatchGroup": "WindowsServer2016",  
      "ComplianceLevel": "CRITICAL",  
      "ApproveAfterDays": 0,  
      "EnableNonSecurity": false,  
      "ApprovedPatches": [],  
      "RejectedPatches": [],  
      "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY"  
    }  
  ]  
}
```

```
    "QA",
    "DEV"
  ],
  "CreateDate": 1550244180.465,
  "ModifiedDate": 1550244180.465,
  "Description": "Patches for Windows Servers",
  "Sources": []
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パッチベースラインについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPatchBaseline](#)」の「」を参照してください。

get-service-setting

次の例は、get-service-setting を使用する方法を説明しています。

AWS CLI

Parameter Store スループットのサービス設定を取得するには

次のget-service-setting例では、指定したリージョンの Parameter Store スループットの現在のサービス設定を取得します。

```
aws ssm get-service-setting \
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
  store/high-throughput-enabled
```

出力:

```
{
  "ServiceSetting": {
    "SettingId": "/ssm/parameter-store/high-throughput-enabled",
    "SettingValue": "false",
    "LastModifiedDate": 1555532818.578,
    "LastModifiedUser": "System",
    "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
    store/high-throughput-enabled",
    "Status": "Default"
  }
}
```

```
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[パラメータストアスループットの増加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceSetting](#)」の「」を参照してください。

label-parameter-version

次の例は、label-parameter-version を使用する方法を説明しています。

AWS CLI

例 1: パラメータの最新バージョンにラベルを追加するには

次のlabel-parameter-version例では、指定されたパラメータの最新バージョンにラベルを追加します。

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --labels "ProductionReady"
```

出力:

```
{  
  "InvalidLabels": [],  
  "ParameterVersion": 3  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パラメータラベルの操作](#)」を参照してください。

例 2: 特定のバージョンのパラメータにラベルを追加するには

次のlabel-parameter-version例では、指定されたバージョンのパラメータにラベルを追加します。

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --version 1
```

```
--labels "ProductionReady" \  
--parameter-version "2" --labels "DevelopmentReady"
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[パラメータラベルの操作](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[LabelParameterVersion](#)」の「」を参照してください。

list-association-versions

次のコード例は、list-association-versions を使用する方法を示しています。

AWS CLI

特定の関連付け ID のすべてのバージョンの関連付けを取得するには

次の list-association-versions の例では、指定された関連付けのすべてのバージョンを一覧表示します。

```
aws ssm list-association-versions \  
--association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

出力:

```
{  
  "AssociationVersions": [  
    {  
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",  
      "AssociationVersion": "1",  
      "CreateDate": 1550505536.726,  
      "Name": "AWS-UpdateSSMAgent",  
      "Parameters": {  
        "allowDowngrade": [  
          "false"  
        ],  
        "version": [  
          ""  
        ]  
      },  
      "Targets": [  
        {  
          "Name": "AWS-UpdateSSMAgent",  
          "Parameters": {  
            "allowDowngrade": [  
              "false"  
            ],  
            "version": [  
              ""  
            ]  
          },  
          "Type": "AWS-UpdateSSMAgent"  
        }  
      ]  
    }  
  ]  
}
```

```

        "Key": "InstanceIds",
        "Values": [
            "i-1234567890abcdef0"
        ]
    },
    "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
    "AssociationName": "UpdateSSMAgent"
}
]
}

```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Systems Manager の関連付けの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssociationVersions](#)」の「」を参照してください。

list-associations

次のコード例は、list-associations を使用する方法を示しています。

AWS CLI

例 1: 特定のインスタンスの関連付けを一覧表示するには

次の list-associations の例は AssociationName、`UpdateSSMAgent` のすべての関連付けを一覧表示します。

```
aws ssm list-associations /
--association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

出力:

```
{
  "Associations": [
    {
      "Name": "AWS-UpdateSSMAgent",
      "InstanceId": "i-1234567890abcdef0",
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "Targets": [

```



```

        {
            "Key": "InstanceIds",
            "Values": [
                "i-016648b75dd622dab"
            ]
        }
    ],
    "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Associated",
        "AssociationStatusAggregatedCount": {
            "Pending": 1
        }
    },
    "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
    "AssociationName": "UpdateSSMAgent"
}
]
}

```

詳細については、「Systems Manager ユーザーガイド」の「[Systems Manager の関連付けの使用](#)」を参照してください。

例 2: 特定のドキュメントの関連付けを一覧表示するには

次の list-associations の例では、指定したドキュメントのすべての関連付けを一覧表示します。

```

aws ssm list-associations /
--association-filter-list "key=Name,value=AWS-UpdateSSMAgent"

```

出力:

```

{
  "Associations": [
    {
      "Name": "AWS-UpdateSSMAgent",
      "InstanceId": "i-1234567890abcdef0",
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "AssociationVersion": "1",
      "Targets": [
        {
          "Key": "InstanceIds",
          "Values": [

```

```

        "i-1234567890abcdef0"
      ]
    }
  ],
  "LastExecutionDate": 1550505828.548,
  "Overview": {
    "Status": "Success",
    "DetailedStatus": "Success",
    "AssociationStatusAggregatedCount": {
      "Success": 1
    }
  },
  "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
  "AssociationName": "UpdateSSMAgent"
},
{
  "Name": "AWS-UpdateSSMAgent",
  "InstanceId": "i-9876543210abcdef0",
  "AssociationId": "fbc07ef7-b985-4684-b82b-0123456789ab",
  "AssociationVersion": "1",
  "Targets": [
    {
      "Key": "InstanceIds",
      "Values": [
        "i-9876543210abcdef0"
      ]
    }
  ],
  "LastExecutionDate": 1550507531.0,
  "Overview": {
    "Status": "Success",
    "AssociationStatusAggregatedCount": {
      "Success": 1
    }
  }
}
]
}

```

詳細については、「Systems Manager ユーザーガイド」の「[Systems Manager の関連付けの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAssociations](#)」の「」を参照してください。

list-command-invocations

次のコード例は、list-command-invocations を使用する方法を示しています。

AWS CLI

特定のコマンドの呼び出しを一覧表示するには

次の list-command-invocations の例では、コマンドのすべての呼び出しを一覧表示します。

```
aws ssm list-command-invocations \  
  --command-id "ef7fd8-9b57-4151-a15c-db9a12345678" \  
  --details
```

出力:

```
{  
  "CommandInvocations": [  
    {  
      "CommandId": "ef7fd8-9b57-4151-a15c-db9a12345678",  
      "InstanceId": "i-02573cafcfEXAMPLE",  
      "InstanceName": "",  
      "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",  
      "DocumentName": "AWS-UpdateSSMAgent",  
      "DocumentVersion": "",  
      "RequestedDateTime": 1582136283.089,  
      "Status": "Success",  
      "StatusDetails": "Success",  
      "StandardOutputUrl": "",  
      "StandardErrorUrl": "",  
      "CommandPlugins": [  
        {  
          "Name": "aws:updateSsmAgent",  
          "Status": "Success",  
          "StatusDetails": "Success",  
          "ResponseCode": 0,  
          "ResponseStartDateTime": 1582136283.419,  
          "ResponseFinishDateTime": 1582136283.51,  
          "Output": "Updating amazon-ssm-agent from 2.3.842.0 to latest  
\\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/"
```

```

ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
update skipped\n",
    "StandardOutputUrl": "",
    "StandardErrorUrl": "",
    "OutputS3Region": "us-east-2",
    "OutputS3BucketName": "",
    "OutputS3KeyPrefix": ""
  }
],
"ServiceRole": "",
"NotificationConfig": {
  "NotificationArn": "",
  "NotificationEvents": [],
  "NotificationType": ""
},
"CloudWatchOutputConfig": {
  "CloudWatchLogGroupName": "",
  "CloudWatchOutputEnabled": false
}
},
{
  "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
  "InstanceId": "i-0471e04240EXAMPLE",
  "InstanceName": "",
  "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
  "DocumentName": "AWS-UpdateSSMAgent",
  "DocumentVersion": "",
  "RequestedDateTime": 1582136283.02,
  "Status": "Success",
  "StatusDetails": "Success",
  "StandardOutputUrl": "",
  "StandardErrorUrl": "",
  "CommandPlugins": [
    {
      "Name": "aws:updateSsmAgent",
      "Status": "Success",
      "StatusDetails": "Success",
      "ResponseCode": 0,
      "ResponseStartDateTime": 1582136283.812,
      "ResponseFinishDateTime": 1582136295.031,
      "Output": "Updating amazon-ssm-agent from 2.3.672.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/

```

```

amazon-ssm-us-east-2/amazon-ssm-agent-updater/2.3.842.0/amazon-ssm-agent-updater-
snap-amd64.tar.gz\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent/2.3.672.0/amazon-ssm-agent-snap-amd64.tar.gz
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
amazon-ssm-agent/2.3.842.0/amazon-ssm-agent-snap-amd64.tar.gz\nInitiating amazon-
ssm-agent update to 2.3.842.0\namazon-ssm-agent updated successfully to 2.3.842.0",
    "StandardOutputUrl": "",
    "StandardErrorUrl": "",
    "OutputS3Region": "us-east-2",
    "OutputS3BucketName": "",
    "OutputS3KeyPrefix": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE/i-0471e04240EXAMPLE/awsupdateSsmAgent"
  }
],
"ServiceRole": "",
"NotificationConfig": {
  "NotificationArn": "",
  "NotificationEvents": [],
  "NotificationType": ""
},
"CloudWatchOutputConfig": {
  "CloudWatchLogGroupName": "",
  "CloudWatchOutputEnabled": false
}
}
]
}

```

詳細については、「AWS Systems Manager ユーザーガイド」の「[コマンドのステータスについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCommandInvocations](#)」の「」を参照してください。

list-commands

次のコード例は、list-commands を使用する方法を示しています。

AWS CLI

例 1: 特定のコマンドのステータスを取得するには

次の list-commands の例では、指定されたコマンドのステータスを取得して表示します。

```
aws ssm list-commands \  
  --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

例 2: 特定の日付より後にリクエストされたコマンドのステータスを取得するには

次の list-commands の例では、指定した日付より後にリクエストされたコマンドの詳細を取得します。

```
aws ssm list-commands \  
  --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

例 3: AWS アカウントでリクエストされたすべてのコマンドを一覧表示するには

次の list-commands 例では、現在の AWS アカウントとリージョンのユーザーがリクエストしたすべてのコマンドを一覧表示します。

```
aws ssm list-commands
```

出力:

```
{  
  "Commands": [  
    {  
      "CommandId": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE",  
      "DocumentName": "AWS-UpdateSSMAgent",  
      "DocumentVersion": "",  
      "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",  
      "ExpiresAfter": "2020-02-19T11:28:02.500000-08:00",  
      "Parameters": {},  
      "InstanceIds": [  
        "i-028ea792daEXAMPLE",  
        "i-02feef8c46EXAMPLE",  
        "i-038613f3f0EXAMPLE",  
        "i-03a530a2d4EXAMPLE",  
        "i-083b678d37EXAMPLE",  
        "i-0dee81debaEXAMPLE"  
      ],  
      "Targets": [],  
      "RequestedDateTime": "2020-02-19T10:18:02.500000-08:00",  
      "Status": "Success",  
      "StatusDetails": "Success",
```

```
"OutputS3BucketName": "",
"OutputS3KeyPrefix": "",
"MaxConcurrency": "50",
"MaxErrors": "100%",
"TargetCount": 6,
"CompletedCount": 6,
"ErrorCount": 0,
"DeliveryTimedOutCount": 0,
"ServiceRole": "",
"NotificationConfig": {
  "NotificationArn": "",
  "NotificationEvents": [],
  "NotificationType": ""
},
"CloudWatchOutputConfig": {
  "CloudWatchLogGroupName": "",
  "CloudWatchOutputEnabled": false
}
}
{
  "CommandId": "e9ade581-c03d-476b-9b07-26667EXAMPLE",
  "DocumentName": "AWS-FindWindowsUpdates",
  "DocumentVersion": "1",
  "Comment": "",
  "ExpiresAfter": "2020-01-24T12:37:31.874000-08:00",
  "Parameters": {
    "KbArticleIds": [
      ""
    ],
    "UpdateLevel": [
      "All"
    ]
  },
  "InstanceIds": [],
  "Targets": [
    {
      "Key": "InstanceIds",
      "Values": [
        "i-00ec29b21eEXAMPLE",
        "i-09911ddd90EXAMPLE"
      ]
    }
  ],
  "RequestedDateTime": "2020-01-24T11:27:31.874000-08:00",
```

```

    "Status": "Success",
    "StatusDetails": "Success",
    "OutputS3BucketName": "my-us-east-2-bucket",
    "OutputS3KeyPrefix": "my-rc-output",
    "MaxConcurrency": "50",
    "MaxErrors": "0",
    "TargetCount": 2,
    "CompletedCount": 2,
    "ErrorCount": 0,
    "DeliveryTimedOutCount": 0,
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "NotificationConfig": {
      "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
      "NotificationEvents": [
        "All"
      ],
      "NotificationType": "Invocation"
    },
    "CloudWatchOutputConfig": {
      "CloudWatchLogGroupName": "",
      "CloudWatchOutputEnabled": false
    }
  }
}
{
  "CommandId": "d539b6c3-70e8-4853-80e5-0ce4fEXAMPLE",
  "DocumentName": "AWS-RunPatchBaseline",
  "DocumentVersion": "1",
  "Comment": "",
  "ExpiresAfter": "2020-01-24T12:21:04.350000-08:00",
  "Parameters": {
    "InstallOverrideList": [
      ""
    ],
    "Operation": [
      "Install"
    ],
    "RebootOption": [
      "RebootIfNeeded"
    ],
    "SnapshotId": [
      ""
    ]
  ]
}

```



```
    },
    "InstanceIds": [],
    "Targets": [
      {
        "Key": "InstanceIds",
        "Values": [
          "i-00ec29b21eEXAMPLE",
          "i-09911ddd90EXAMPLE"
        ]
      }
    ],
    "RequestedDateTime": "2020-01-24T11:11:04.350000-08:00",
    "Status": "Success",
    "StatusDetails": "Success",
    "OutputS3BucketName": "my-us-east-2-bucket",
    "OutputS3KeyPrefix": "my-rc-output",
    "MaxConcurrency": "50",
    "MaxErrors": "0",
    "TargetCount": 2,
    "CompletedCount": 2,
    "ErrorCount": 0,
    "DeliveryTimedOutCount": 0,
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "NotificationConfig": {
      "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-east-2-
notification-arn",
      "NotificationEvents": [
        "All"
      ],
      "NotificationType": "Invocation"
    },
    "CloudWatchOutputConfig": {
      "CloudWatchLogGroupName": "",
      "CloudWatchOutputEnabled": false
    }
  }
]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListCommands](#)」の「」を参照してください。

list-compliance-items

次の例は、list-compliance-items を使用する方法を説明しています。

AWS CLI

特定のインスタンスのコンプライアンス項目を一覧表示するには

この例では、指定したインスタンスのすべてのコンプライアンス項目を一覧表示します。

コマンド:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-types "ManagedInstance"
```

出力:

```
{
  "ComplianceItems": [
    {
      "ComplianceType": "Association",
      "ResourceType": "ManagedInstance",
      "ResourceId": "i-1234567890abcdef0",
      "Id": "8dfe3659-4309-493a-8755-0123456789ab",
      "Title": "",
      "Status": "COMPLIANT",
      "Severity": "UNSPECIFIED",
      "ExecutionSummary": {
        "ExecutionTime": 1550408470.0
      },
      "Details": {
        "DocumentName": "AWS-GatherSoftwareInventory",
        "DocumentVersion": "1"
      }
    },
    {
      "ComplianceType": "Association",
      "ResourceType": "ManagedInstance",
      "ResourceId": "i-1234567890abcdef0",
```

```

    "Id": "e4c2ed6d-516f-41aa-aa2a-0123456789ab",
    "Title": "",
    "Status": "COMPLIANT",
    "Severity": "UNSPECIFIED",
    "ExecutionSummary": {
      "ExecutionTime": 1550508475.0
    },
    "Details": {
      "DocumentName": "AWS-UpdateSSMAgent",
      "DocumentVersion": "1"
    }
  },
  ...
],
"NextToken": "--token string truncated--"
}

```

特定のインスタンスと関連付け ID のコンプライアンス項目を一覧表示するには

この例では、指定したインスタンスと関連付け ID のすべてのコンプライアンス項目を一覧表示します。

コマンド:

```

aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ComplianceType,Values=Association,Type=EQUAL" "Key=Id,Values=e4c2ed6d-516f-41aa-
aa2a-0123456789ab,Type=EQUAL"

```

特定の日時より後のインスタンスのコンプライアンス項目を一覧表示するには

この例では、指定した日時より後のインスタンスのすべてのコンプライアンス項目を一覧表示します。

コマンド:

```

aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"

```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListComplianceItems](#)」の「」を参照してください。

list-compliance-summaries

次の例は、list-compliance-summaries を使用する方法を説明しています。

AWS CLI

すべてのコンプライアンスタイプのコンプライアンス概要を一覧表示するには

この例では、アカウント内のすべてのコンプライアンスタイプのコンプライアンス概要を一覧表示します。

コマンド:

```
aws ssm list-compliance-summaries
```

出力:

```
{
  "ComplianceSummaryItems": [
    {
      "ComplianceType": "Association",
      "CompliantSummary": {
        "CompliantCount": 2,
        "SeveritySummary": {
          "CriticalCount": 0,
          "HighCount": 0,
          "MediumCount": 0,
          "LowCount": 0,
          "InformationalCount": 0,
          "UnspecifiedCount": 2
        }
      },
      "NonCompliantSummary": {
        "NonCompliantCount": 0,
        "SeveritySummary": {
          "CriticalCount": 0,
          "HighCount": 0,
          "MediumCount": 0,
          "LowCount": 0,
          "InformationalCount": 0,
          "UnspecifiedCount": 0
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "ComplianceType": "Patch",
    "CompliantSummary": {
      "CompliantCount": 1,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 1
      }
    },
    "NonCompliantSummary": {
      "NonCompliantCount": 1,
      "SeveritySummary": {
        "CriticalCount": 1,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 0
      }
    }
  },
  ...
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

特定のコンプライアンスタイプのコンプライアンス概要を一覧表示するには

この例では、パッチコンプライアンスタイプのコンプライアンス概要を一覧表示します。

コマンド:

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListComplianceSummaries](#)」の「」を参照してください。

list-document-metadata-history

次の例は、list-document-metadata-history を使用する方法を説明しています。

AWS CLI

例: 変更テンプレートの承認履歴とステータスを表示するには

次のlist-document-metadata-history例では、指定された Change Manager 変更テンプレートの承認履歴を返します。

```
aws ssm list-document-metadata-history \  
  --name MyChangeManageTemplate \  
  --metadata DocumentReviews
```

出力:

```
{  
  "Name": "MyChangeManagerTemplate",  
  "DocumentVersion": "1",  
  "Author": "arn:aws:iam::111222333444:user/JohnDoe",  
  "Metadata": {  
    "ReviewerResponse": [  
      {  
        "CreateTime": "2021-07-30T11:58:28.025000-07:00",  
        "UpdateTime": "2021-07-30T12:01:19.274000-07:00",  
        "ReviewStatus": "APPROVED",  
        "Comment": [  
          {  
            "Type": "COMMENT",  
            "Content": "I approve this template version"  
          }  
        ],  
        "Reviewer": "arn:aws:iam::111222333444:user/ShirleyRodriguez"  
      },  
      {  
        "CreateTime": "2021-07-30T11:58:28.025000-07:00",  
        "UpdateTime": "2021-07-30T11:58:28.025000-07:00",  
        "ReviewStatus": "PENDING"  
      }  
    ]  
  }  
}
```

詳細については、[「Systems Manager ユーザーガイド」の「変更テンプレートの確認と承認または拒否」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDocumentMetadataHistory](#)」の「」を参照してください。

list-document-versions

次のコード例は、list-document-versions を使用する方法を示しています。

AWS CLI

ドキュメントバージョンを一覧表示するには

次の list-document-versions の例では、Systems Manager ドキュメントのすべてのバージョンを一覧表示します。

```
aws ssm list-document-versions \  
  --name "Example"
```

出力:

```
{  
  "DocumentVersions": [  
    {  
      "Name": "Example",  
      "DocumentVersion": "1",  
      "CreateDate": 1583257938.266,  
      "IsDefaultVersion": true,  
      "DocumentFormat": "YAML",  
      "Status": "Active"  
    }  
  ]  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Sending Commands that Use the Document Version Parameter](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDocumentVersions](#)」の「」を参照してください。

list-documents

次のコード例は、list-documents を使用する方法を示しています。

AWS CLI

例 1: ドキュメントを一覧表示するには

次の list-documents の例では、カスタムタグでタグ付けされたリクエスト元のアカウントが所有するドキュメントを一覧表示します。

```
aws ssm list-documents \
  --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

出力:

```
{
  "DocumentIdentifiers": [
    {
      "Name": "Example",
      "Owner": "29884EXAMPLE",
      "PlatformTypes": [
        "Windows",
        "Linux"
      ],
      "DocumentVersion": "1",
      "DocumentType": "Automation",
      "SchemaVersion": "0.3",
      "DocumentFormat": "YAML",
      "Tags": [
        {
          "Key": "DocUse",
          "Value": "Testing"
        }
      ]
    }
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[AWS Systems Manager ドキュメント](#)」を参照してください。

例 2: 共有ドキュメントを一覧表示するには

次のlist-documents例では、 が所有していないプライベート共有ドキュメントなど、共有ドキュメントを一覧表示します AWS。

```
aws ssm list-documents \
  --filters Key=Name,Values=sharedDocNamePrefix Key=Owner,Values=Private
```

出力:

```
{
  "DocumentIdentifiers": [
    {
      "Name": "Example",
      "Owner": "12345EXAMPLE",
      "PlatformTypes": [
        "Windows",
        "Linux"
      ],
      "DocumentVersion": "1",
      "DocumentType": "Command",
      "SchemaVersion": "0.3",
      "DocumentFormat": "YAML",
      "Tags": []
    }
  ]
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[AWS Systems Manager ドキュメント](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDocuments](#)」の「」を参照してください。

list-inventory-entries

次のコード例は、list-inventory-entries を使用する方法を示しています。

AWS CLI

例 1: インスタンスの特定のインベントリタイプのエントリを表示するには

次のlist-inventory-entries例では、特定のインスタンスの AWS:Application インベントリタイプのインベントリエントリを一覧表示します。

```
aws ssm list-inventory-entries \  
  --instance-id "i-1234567890abcdef0" \  
  --type-name "AWS:Application"
```

出力:

```
{  
  "TypeName": "AWS:Application",  
  "InstanceId": "i-1234567890abcdef0",  
  "SchemaVersion": "1.1",  
  "CaptureTime": "2019-02-15T12:17:55Z",  
  "Entries": [  
    {  
      "Architecture": "i386",  
      "Name": "Amazon SSM Agent",  
      "PackageId": "{88a60be2-89a1-4df8-812a-80863c2a2b68}",  
      "Publisher": "Amazon Web Services",  
      "Version": "2.3.274.0"  
    },  
    {  
      "Architecture": "x86_64",  
      "InstalledTime": "2018-05-03T13:42:34Z",  
      "Name": "AmazonCloudWatchAgent",  
      "Publisher": "",  
      "Version": "1.200442.0"  
    }  
  ]  
}
```

例 2: インスタンスに割り当てられたカスタムインベントリエントリを表示するには

次の `list-inventory-entries` の例では、インスタンスに割り当てられたカスタムインベントリエントリを一覧表示します。

```
aws ssm list-inventory-entries \  
  --instance-id "i-1234567890abcdef0" \  
  --type-name "Custom:RackInfo"
```

出力:

```
{  
  "TypeName": "Custom:RackInfo",
```

```
"InstanceId": "i-1234567890abcdef0",
"SchemaVersion": "1.0",
"CaptureTime": "2021-05-22T10:01:01Z",
"Entries": [
  {
    "RackLocation": "Bay B/Row C/Rack D/Shelf E"
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListInventoryEntries](#)」の「」を参照してください。

list-ops-item-related-items

次のコード例は、list-ops-item-related-items を使用する方法を示しています。

AWS CLI

の関連項目リソースを一覧表示するには OpsItem

次のlist-ops-item-related-items例では、 の関連項目リソースを一覧表示します OpsItem。

```
aws ssm list-ops-item-related-items \
  --ops-item-id "oi-f99f2EXAMPLE"
```

出力:

```
{
  "Summaries": [
    {
      "OpsItemId": "oi-f99f2EXAMPLE",
      "AssociationId": "e2036148-cccb-490e-ac2a-390e5EXAMPLE",
      "ResourceType": "AWS::SSMIncidents::IncidentRecord",
      "AssociationType": "IsParentOf",
      "ResourceUri": "arn:aws:ssm-incidents::111122223333:incident-record/example-response/64bd9b45-1d0e-2622-840d-03a87a1451fa",
      "CreatedBy": {
        "Arn": "arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForIncidentManager/IncidentResponse"
      },
    },
  ],
}
```

```
    "CreatedTime": "2021-08-11T18:47:14.994000+00:00",
    "LastModifiedBy": {
      "Arn": "arn:aws:sts::111122223333:assumed-role/
AWSServiceRoleForIncidentManager/IncidentResponse"
    },
    "LastModifiedTime": "2021-08-11T18:47:14.994000+00:00"
  }
]
}
```

詳細については、AWS Systems [Manager ユーザーガイドの「」の「Incident Manager インシデントの使用 OpsCenter」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOpsItemRelatedItems](#)」の「」を参照してください。

list-resource-compliance-summaries

次のコード例は、list-resource-compliance-summaries を使用する方法を示しています。

AWS CLI

リソースレベルのコンプライアンス概要数を一覧表示するには

この例では、リソースレベルのコンプライアンス概要数を一覧表示します。

コマンド:

```
aws ssm list-resource-compliance-summaries
```

出力:

```
{
  "ResourceComplianceSummaryItems": [
    {
      "ComplianceType": "Association",
      "ResourceType": "ManagedInstance",
      "ResourceId": "i-1234567890abcdef0",
      "Status": "COMPLIANT",
      "OverallSeverity": "UNSPECIFIED",
      "ExecutionSummary": {
        "ExecutionTime": 1550509273.0
      }
    },
  ],
}
```

```
    "CompliantSummary": {
      "CompliantCount": 2,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 2
      }
    },
    "NonCompliantSummary": {
      "NonCompliantCount": 0,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 0
      }
    }
  },
  {
    "ComplianceType": "Patch",
    "ResourceType": "ManagedInstance",
    "ResourceId": "i-9876543210abcdef0",
    "Status": "COMPLIANT",
    "OverallSeverity": "UNSPECIFIED",
    "ExecutionSummary": {
      "ExecutionTime": 1550248550.0,
      "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
      "ExecutionType": "Command"
    },
    "CompliantSummary": {
      "CompliantCount": 397,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 397
      }
    }
  }
}
```

```
    },
    "NonCompliantSummary": {
      "NonCompliantCount": 0,
      "SeveritySummary": {
        "CriticalCount": 0,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 0
      }
    }
  }
],
"NextToken": "--token string truncated--"
}
```

特定のコンプライアンスタイプのリソースレベルのコンプライアンス概要を一覧表示するには
この例では、パッチコンプライアンスタイプのリソースレベルのコンプライアンス概要を一覧表示します。

コマンド:

```
aws ssm list-resource-compliance-summaries --  
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListResourceComplianceSummaries](#)」の「」を参照してください。

list-resource-data-sync

次の例は、list-resource-data-sync を使用する方法を説明しています。

AWS CLI

リソースデータ同期設定を一覧表示するには

この例では、リソースデータ同期設定に関する情報を取得します。

```
aws ssm list-resource-data-sync
```

出力:

```
{
  "ResourceDataSyncItems": [
    {
      "SyncName": "MyResourceDataSync",
      "S3Destination": {
        "BucketName": "ssm-resource-data-sync",
        "SyncFormat": "JsonSerDe",
        "Region": "us-east-1"
      },
      "LastSyncTime": 1550261472.003,
      "LastSuccessfulSyncTime": 1550261472.003,
      "LastStatus": "Successful",
      "SyncCreatedTime": 1543235736.72,
      "LastSyncStatusMessage": "The sync was successfully completed"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceDataSync](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

パッチベースラインに適用されたタグを一覧表示するには

次の list-tags-for-resource の例では、パッチベースラインのタグを一覧表示します。

```
aws ssm list-tags-for-resource \
  --resource-type "PatchBaseline" \
  --resource-id "pb-0123456789abcdef0"
```

出力:

```
{
  "TagList": [
```

```
{
  "Key": "Environment",
  "Value": "Production"
},
{
  "Key": "Region",
  "Value": "EMEA"
}
]
```

詳細については、AWS 「全般のリファレンス」の[AWS 「リソースのタグ付け」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

modify-document-permission

次の例は、modify-document-permission を使用する方法を説明しています。

AWS CLI

ドキュメントのアクセス許可を変更するには

次の modify-document-permission の例では、Systems Manager ドキュメントをパブリックで共有します。

```
aws ssm modify-document-permission \
  --name "Example" \
  --permission-type "Share" \
  --account-ids-to-add "All"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Share a Systems Manager Document](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyDocumentPermission](#)」の「」を参照してください。

put-compliance-items

次の例は、put-compliance-items を使用する方法を説明しています。

AWS CLI

コンプライアンスタイプおよびコンプライアンスの詳細を指定されたインスタンスに登録するには

この例では、指定されたマネージドインスタンスにコンプライアンスタイプ Custom:AVCheck を登録します。コマンドが成功した場合、出力はありません。

コマンド:

```
aws ssm put-compliance-items --resource-id "i-1234567890abcdef0" --  
resource-type "ManagedInstance" --compliance-type "Custom:AVCheck"  
--execution-summary "ExecutionTime=2019-02-18T16:00:00Z" --  
items "Id=Version2.0,Title=ScanHost,Severity=CRITICAL,Status=COMPLIANT"
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutComplianceItems](#)」の「」を参照してください。

put-inventory

次のコード例は、put-inventory を使用する方法を示しています。

AWS CLI

顧客のメタデータをインスタンスに割り当てるには

この例では、ラックの場所情報をインスタンスに割り当てます。コマンドが成功した場合、出力はありません。

コマンド (Linux):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --items  
'[{"TypeName": "Custom:RackInfo", "SchemaVersion": "1.0", "CaptureTime":  
"2019-01-22T10:01:01Z", "Content": [{"RackLocation": "Bay B/Row C/Rack D/Shelf  
E"}]}'
```

コマンド (Windows):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --
items "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2019-01-22T10:01:01Z,Content=[
B/Row C/Rack D/Shelf F']]"
```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutInventory](#)」の「」を参照してください。

put-parameter

次のコード例は、put-parameter を使用する方法を示しています。

AWS CLI

例 1: パラメータ値を変更するには

次の put-parameter の例は、指定されたパラメータの値を変更します。

```
aws ssm put-parameter \
  --name "MyStringParameter" \
  --type "String" \
  --value "Vici" \
  --overwrite
```

出力:

```
{
  "Version": 2,
  "Tier": "Standard"
}
```

詳細については、[「Systems Manager ユーザーガイド AWS CLI」](#)の「[Systems Manager パラメータの作成](#)」、[「パラメータ階層の管理」](#) <<https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>、[「パラメータポリシーの操作](#)」を参照してください。AWS

例 2: アドバンスドパラメータを作成するには

次の put-parameter の例は、アドバンスドパラメータを作成します。

```
aws ssm put-parameter \
  --name "MyAdvancedParameter" \
```

```
--description "This is an advanced parameter" \  
--value "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod  
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,  
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat  
[truncated]" \  
--type "String" \  
--tier Advanced
```

出力:

```
{  
  "Version": 1,  
  "Tier": "Advanced"  
}
```

詳細については、[「Systems Manager ユーザーガイドAWS CLI」の「Systems Manager パラメータの作成」](#)、[「パラメータ階層の管理」](#) <<https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>、[「パラメータポリシーの操作」](#)を参照してください。AWS

例 3: スタンダードパラメータをアドバンスドパラメータに変換するには

次の `put-parameter` の例は、既存のスタンダードパラメータをアドバンスドパラメータに変換します。

```
aws ssm put-parameter \  
  --name "MyConvertedParameter" \  
  --value "abc123" \  
  --type "String" \  
  --tier Advanced \  
  --overwrite
```

出力:

```
{  
  "Version": 2,  
  "Tier": "Advanced"  
}
```

詳細については、[「Systems Manager ユーザーガイドAWS CLI」の「Systems Manager パラメータの作成」](#)、[「パラメータ階層の管理」](#) <<https://docs.aws.amazon.com/systems-manager/>

latest/userguide/parameter-store-advanced-parameters.html>`__」、[「パラメータポリシーの操作」](#)を参照してください。AWS

例 4: ポリシーがアタッチされたパラメータを作成するには

次の put-parameter の例は、パラメータポリシーがアタッチされたアドバンストパラメータを作成します。

```
aws ssm put-parameter \  
  --name "/Finance/Payroll/q2accesskey" \  
  --value "P@sSw)rd" \  
  --type "SecureString" \  
  --tier Advanced \  
  --policies "[{\\"Type\\":\\"Expiration\\",\\"Version\\":\\"1.0\\",\\"Attributes\\":  
{\\"Timestamp\\":\\"2020-06-30T00:00:00.000Z\\"}},{\\"Type\\":\\"ExpirationNotification\\",  
\\"Version\\":\\"1.0\\",\\"Attributes\\":{\\"Before\\":\\"5\\",\\"Unit\\":\\"Days\\"}},{\\"Type\\":  
\\"NoChangeNotification\\",\\"Version\\":\\"1.0\\",\\"Attributes\\":{\\"After\\":\\"60\\",\\"Unit  
\\":\\"Days\\"}}]"
```

出力:

```
{  
  "Version": 1,  
  "Tier": "Advanced"  
}
```

詳細については、[「Systems Manager ユーザーガイドAWS CLI」](#)の[「Systems Manager パラメータの作成」](#)、[「パラメータ階層の管理」](#) < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>`__」、[「パラメータポリシーの操作」](#)を参照してください。AWS

例 5: 既存のパラメータにポリシーを追加するには

次の put-parameter の例は、ポリシーを既存のアドバンストパラメータにアタッチします。

```
aws ssm put-parameter \  
  --name "/Finance/Payroll/q2accesskey" \  
  --value "N3wP@sSw)rd" \  
  --type "SecureString" \  
  --tier Advanced \  
  --policies "[{\\"Type\\":\\"Expiration\\",\\"Version\\":\\"1.0\\",\\"Attributes\\":  
{\\"Timestamp\\":\\"2020-06-30T00:00:00.000Z\\"}},{\\"Type\\":\\"ExpirationNotification\\",
```

```
\"Version\": \"1.0\", \"Attributes\": {\"Before\": \"5\", \"Unit\": \"Days\"}}, {\"Type\": \"NoChangeNotification\", \"Version\": \"1.0\", \"Attributes\": {\"After\": \"60\", \"Unit\": \"Days\"}}]"
--overwrite
```

出力:

```
{
  "Version": 2,
  "Tier": "Advanced"
}
```

詳細については、[「Systems Manager ユーザーガイドAWS CLI」の「Systems Manager パラメータの作成」](#)、[「パラメータ階層の管理」](#) <<https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>、[「パラメータポリシーの操作」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[PutParameter](#)」の「」を参照してください。

register-default-patch-baseline

次の例は、register-default-patch-baseline を使用する方法を説明しています。

AWS CLI

デフォルトパッチベースラインを設定するには

次の register-default-patch-baseline の例では、指定したカスタムパッチベースラインを、サポートするオペレーティングシステムタイプのデフォルトのパッチベースラインとして登録します。

```
aws ssm register-default-patch-baseline \
  --baseline-id "pb-abc123cf9bEXAMPLE"
```

出力:

```
{
  "BaselineId": "pb-abc123cf9bEXAMPLE"
}
```

次のregister-default-patch-baseline例では、AWS for CentOS が提供するデフォルトのパッチベースラインをデフォルトのパッチベースラインとして登録します。

```
aws ssm register-default-patch-baseline \  
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/  
pb-0574b43a65ea646ed"
```

出力:

```
{  
  "BaselineId": "pb-abc123cf9bEXAMPLE"  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[事前定義されたパッチベースラインについて](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterDefaultPatchBaseline](#)」の「」を参照してください。

register-patch-baseline-for-patch-group

次の例は、register-patch-baseline-for-patch-group を使用する方法を説明しています。

AWS CLI

パッチグループのパッチベースラインを登録するには

次の register-patch-baseline-for-patch-group の例では、パッチグループのパッチベースラインを登録します。

```
aws ssm register-patch-baseline-for-patch-group \  
  --baseline-id "pb-045f10b4f382baeda" \  
  --patch-group "Production"
```

出力:

```
{  
  "BaselineId": "pb-045f10b4f382baeda",  
  "PatchGroup": "Production"  
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「パッチグループ <<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>>__の作成」および「[パッチベースラインへのパッチグループの追加](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RegisterPatchBaselineForPatchGroup](#)」の「」を参照してください。

register-target-with-maintenance-window

次のコード例は、register-target-with-maintenance-window を使用する方法を示しています。

AWS CLI

例 1: メンテナンスウィンドウに単一のターゲットを登録するには

次の register-target-with-maintenance-window の例では、インスタンスをメンテナンスウィンドウに登録します。

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \  
  --owner-information "Single instance" \  
  --resource-type "INSTANCE"
```

出力:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

例 2: インスタンスを使用して複数のターゲットをメンテナンスウィンドウに登録するには IDs

次の register-target-with-maintenance-window 例では、インスタンスを指定して、2つのインスタンスをメンテナンスウィンドウに登録します IDs。

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --target "Key=InstanceIds,Values=i-0000293ffd8c57862,i-0cb2b964d3e14fd9f" \  
  --owner-information "Two instances in a list" \  
  --resource-type "INSTANCE"
```

```
--resource-type "INSTANCE"
```

出力:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

例 3: リソースタグを使用してメンテナンスウィンドウにターゲットを登録するには

次の `register-target-with-maintenance-window` の例では、インスタンスに適用されたリソースタグを指定して、インスタンスをメンテナンスウィンドウに登録します。

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-06cf17cbefcb4bf4f" \  
  --targets "Key=tag:Environment,Values=Prod" "Key=Role,Values=Web" \  
  --owner-information "Production Web Servers" \  
  --resource-type "INSTANCE"
```

出力:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

例 4: タグキーのグループを使用してターゲットを登録するには

次の `register-target-with-maintenance-window` の例では、キー値に関係なく、1 つまたは複数のタグキーが割り当てられているインスタンスをすべて登録します。

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "INSTANCE" \  
  --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

出力:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```



```
}
```

例 5: リソースグループ名を使用してターゲットを登録するには

次の `register-target-with-maintenance-window` の例では、含まれるリソースタイプに関係なく、指定されたリソースグループを登録します。

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "RESOURCE_GROUP" \  
  --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

出力:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

詳細については、AWS Systems Manager [ユーザーガイドの「メンテナンスウィンドウ \(AWS CLI\) でターゲットインスタンスを登録する」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RegisterTargetWithMaintenanceWindow](#)」の「」を参照してください。

register-task-with-maintenance-window

次の例は、`register-task-with-maintenance-window` を使用する方法を説明しています。

AWS CLI

例 1: メンテナンスウィンドウにオートメーションタスクを登録するには

次の `register-task-with-maintenance-window` の例では、インスタンスをターゲットとするメンテナンスウィンドウにオートメーションタスクを登録します。

```
aws ssm register-task-with-maintenance-window \  
  --window-id "mw-082dcd7649EXAMPLE" \  
  --targets Key=InstanceIds,Values=i-1234520122EXAMPLE \  
  --task-arn AWS-RestartEC2Instance \  
  --service-role-arn arn:aws:iam::111222333444:role/SSM --task-type AUTOMATION \  
  --
```

```

--task-invocation-parameters "{\"Automation\":{\"DocumentVersion\":{\"\"$LATEST\"},
\"Parameters\":{\"\"InstanceId\":[\"{{RESOURCE_ID}}\"]}}}" \
--priority 0 \
--max-concurrency 1 \
--max-errors 1 \
--name "AutomationExample" \
--description "Restarting EC2 Instance for maintenance"

```

出力:

```

{
  "WindowTaskId": "11144444-5555-6666-7777-88888888"
}

```

詳細については、AWS Systems Manager ユーザーガイドの [「メンテナンスウィンドウ \(AWS CLI\) でタスクを登録する」](#) を参照してください。

例 2: メンテナンスウィンドウに Lambda タスクを登録するには

次の `register-task-with-maintenance-window` の例では、インスタンスをターゲットとするメンテナンスウィンドウに Lambda タスクを登録します。

```

aws ssm register-task-with-maintenance-window \
--window-id "mw-082dcd7649dee04e4" \
--targets Key=InstanceIds,Values=i-12344d305eEXAMPLE \
--task-arn arn:aws:lambda:us-east-1:111222333444:function:SSMTestLAMBDA \
--service-role-arn arn:aws:iam::111222333444:role/SSM \
--task-type LAMBDA \
--task-invocation-parameters '{"Lambda":{"Payload":{"\"InstanceId\":
\"{{RESOURCE_ID}}\"},\"targetType\":{\"\"{{TARGET_TYPE}}\""},\"Qualifier\":\"$LATEST\"}}' \
--priority 0 \
--max-concurrency 10 \
--max-errors 5 \
--name "Lambda_Example" \
--description "My Lambda Example"

```

出力:

```

{
  "WindowTaskId": "22244444-5555-6666-7777-88888888"
}

```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウ \(AWS CLI\) でタスクを登録する](#)」を参照してください。

例 3: メンテナンスウィンドウに Run Command タスクを登録するには

次の `register-task-with-maintenance-window` の例では、インスタンスをターゲットとするメンテナンスウィンドウに Run Command タスクを登録します。

```
aws ssm register-task-with-maintenance-window \  
  --window-id "mw-082dcd7649dee04e4" \  
  --targets "Key=InstanceIds,Values=i-12344d305eEXAMPLE" \  
  --service-role-arn "arn:aws:iam::111222333444:role/SSM" \  
  --task-type "RUN_COMMAND" \  
  --name "SSMInstallPowerShellModule" \  
  --task-arn "AWS-InstallPowerShellModule" \  
  --task-invocation-parameters "{\"RunCommand\":{\"Comment\":"\"\",  
\"OutputS3BucketName\":"\"runcommandlogs\"},\"Parameters\":{\"commands\":[\"Get-  
Module -ListAvailable\"],\"executionTimeout\":[\"3600\"],\"source\":[\"https://  
\\gallery.technet.microsoft.com/EZ0ut-33ae0fb7/file/110351/1/EZ0ut.zip\"],  
\"workingDirectory\":[\"\\\\\\\\\\\\\\\\\"]},\"TimeoutSeconds\":600}" \  
  --max-concurrency 1 \  
  --max-errors 1 \  
  --priority 10
```

出力:

```
{  
  "WindowTaskId": "333444444-5555-6666-7777-888888888"  
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウ \(AWS CLI\) でタスクを登録する](#)」を参照してください。

例 4: Step Functions タスクをメンテナンスウィンドウに登録するには

次の `register-task-with-maintenance-window` の例では、インスタンスをターゲットとするメンテナンスウィンドウに Step Functions タスクを登録します。

```
aws ssm register-task-with-maintenance-window \  
  --window-id "mw-1234d787d6EXAMPLE" \  
  --targets Key=WindowTargetIds,Values=12347414-69c3-49f8-95b8-ed2dcEXAMPLE \  
  --task-type "STEP_FUNCTIONS"
```

```

--task-arn arn:aws:states:us-east-1:111222333444:stateMachine:SSMTestStateMachine \
--service-role-arn arn:aws:iam::111222333444:role/MaintenanceWindows \
--task-type STEP_FUNCTIONS \
--task-invocation-parameters '{"StepFunctions":{"Input":{"InstanceId\":"{{RESOURCE_ID}}\}}"}' \
--priority 0 \
--max-concurrency 10 \
--max-errors 5 \
--name "Step_Functions_Example" \
--description "My Step Functions Example"

```

出力:

```

{
  "WindowTaskId": "444444444-5555-6666-7777-88888888"
}

```

詳細については、AWS Systems Manager ユーザーガイドの [「メンテナンスウィンドウ \(AWS CLI\) でタスクを登録する」](#) を参照してください。

例 5: メンテナンスウィンドウのターゲット ID を使用してタスクを登録するには

次の `register-task-with-maintenance-window` の例では、メンテナンスウィンドウのターゲット ID を使用してタスクを登録します。メンテナンスウィンドウのターゲット ID は、`aws ssm register-target-with-maintenance-window` コマンドの出力に含まれていました。この情報は `aws ssm describe-maintenance-window-targets` コマンドの出力から取得することもできます。

```

aws ssm register-task-with-maintenance-window \
--targets "Key=WindowTargetIds,Values=350d44e6-28cc-44e2-951f-4b2c9EXAMPLE" \
--task-arn "AWS-RunShellScript" \
--service-role-arn "arn:aws:iam::111222333444:role/MaintenanceWindowsRole" \
--window-id "mw-ab12cd34eEXAMPLE" \
--task-type "RUN_COMMAND" \
--task-parameters '{"commands\":"Values\":[\df\}]}' \
--max-concurrency 1 \
--max-errors 1 \
--priority 10

```

出力:

```
{
  "WindowTaskId": "33344444-5555-6666-7777-88888888"
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウ \(AWS CLI\) でタスクを登録する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RegisterTaskWithMaintenanceWindow](#)」の「」を参照してください。

remove-tags-from-resource

次のコード例は、remove-tags-from-resource を使用する方法を示しています。

AWS CLI

パッチベースラインからタグを削除するには

次の remove-tags-from-resource の例では、パッチベースラインからタグが削除されません。

```
aws ssm remove-tags-from-resource \
  --resource-type "PatchBaseline" \
  --resource-id "pb-0123456789abcdef0" \
  --tag-keys "Region"
```

このコマンドでは何も出力されません。

詳細については、AWS 「全般のリファレンス」の [AWS 「リソースのタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [RemoveTagsFromResource](#)」の「」を参照してください。

reset-service-setting

次の例は、reset-service-setting を使用する方法を説明しています。

AWS CLI

Parameter Store スループットのサービス設定をリセットするには

次のreset-service-setting例では、指定したリージョンの Parameter Store スループットのサービス設定をリセットして、スループットの増加を使用しなくなります。

```
aws ssm reset-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
  store/high-throughput-enabled
```

出力:

```
{  
  "ServiceSetting": {  
    "SettingId": "/ssm/parameter-store/high-throughput-enabled",  
    "SettingValue": "false",  
    "LastModifiedDate": 1555532818.578,  
    "LastModifiedUser": "System",  
    "ARN": "arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
    store/high-throughput-enabled",  
    "Status": "Default"  
  }  
}
```

詳細については、AWS Systems Manager ユーザーガイドの [「パラメータストアスループットの増加」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetServiceSetting](#)」の「」を参照してください。

resume-session

次のコード例は、resume-session を使用する方法を示しています。

AWS CLI

Session Manager セッションを再開するには

この例では、インスタンスが切断された後に、インスタンスとの Session Manager セッションresume-sessionを再開します。このインタラクティブなコマンドでは、呼び出しを実行するクライアントマシンに Session Manager プラグインがインストールされている必要があることに留意してください。

```
aws ssm resume-session \  

```

```
--session-id Mary-Major-07a16060613c408b5
```

出力:

```
{
  "SessionId": "Mary-Major-07a16060613c408b5",
  "TokenValue":
  "AAEAAVbTGsa0nyvcUoNGqifbv5r/81gxuQljCuY8qVcv0noBAAAAAFxtd3jIXAFUUXGTJ7zF/
  AWJpWdvi0lF5p3dlAgrqVIV06IEXhkHLz0/1gXKRKEME71E6TL0p1LDJAMZ
  +kREejkZu4c5AxMkrQjMF+gtHP1bYJKTwtHQd1wju1PLex08SH17g5R/
  wekrj6WsDUpnEegFBfGftpAIz2GXQVfTJXKfkc5qepQ11C11D0IT2doz0qXgHwfQHfAKLErM5dWDZqKwyT1Z3iw7unQd
  +ihfGa6MEJJ97Jmat/a2TspEn0jNn9Mvu5iwXIW2yCvWZrGUj+/
  QI5Xr7s1XJBEEnSKR54o4fN0GV9RWl0RZsZm1mki0JJtiwWgZ",
  "StreamUrl": "wss://ssmmessages.us-east-2.amazonaws.com/v1/data-channel/Mary-
  Major-07a16060613c408b5?role=publish_subscribe"
}
```

詳細については、AWS「[Systems Manager ユーザーガイド](#)」の「[の Session Manager プラグインをインストールする AWS CLI](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンス ResumeSession](#)」の「」を参照してください。

send-automation-signal

次の例は、send-automation-signal を使用する方法を説明しています。

AWS CLI

オートメーション実行にシグナルを送信するには

次のsend-automation-signal例では、オートメーション実行に Approve シグナルを送信します。

```
aws ssm send-automation-signal \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \  
  --signal-type "Approve"
```

このコマンドでは何も出力されません。

詳細については、AWS Systems Manager ユーザーガイドの「[承認者によるオートメーションワークフローの実行](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendAutomationSignal](#)」の「」を参照してください。

send-command

次のコード例は、send-command を使用する方法を示しています。

AWS CLI

例 1: 1 つ以上のリモートインスタンスでコマンドを実行するには

次の send-command の例では、ターゲットインスタンスで echo コマンドを実行します。

```
aws ssm send-command \  
  --document-name "AWS-RunShellScript" \  
  --parameters 'commands=["echo HelloWorld"]' \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0" \  
  --comment "echo HelloWorld"
```

出力:

```
{  
  "Command": {  
    "CommandId": "92853adf-ba41-4cd6-9a88-142d1EXAMPLE",  
    "DocumentName": "AWS-RunShellScript",  
    "DocumentVersion": "",  
    "Comment": "echo HelloWorld",  
    "ExpiresAfter": 1550181014.717,  
    "Parameters": {  
      "commands": [  
        "echo HelloWorld"  
      ]  
    },  
    "InstanceIds": [  
      "i-0f00f008a2dcbef2"  
    ],  
    "Targets": [],  
    "RequestedDateTime": 1550173814.717,  
    "Status": "Pending",  
    "StatusDetails": "Pending",  
    "OutputS3BucketName": "",  
    "OutputS3KeyPrefix": "",
```



```
"MaxConcurrency": "50",
"MaxErrors": "0",
"TargetCount": 1,
"CompletedCount": 0,
"ErrorCount": 0,
"DeliveryTimedOutCount": 0,
"ServiceRole": "",
"NotificationConfig": {
  "NotificationArn": "",
  "NotificationEvents": [],
  "NotificationType": ""
},
"CloudWatchOutputConfig": {
  "CloudWatchLogGroupName": "",
  "CloudWatchOutputEnabled": false
}
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

例 2: インスタンスの IP 情報を取得するには

次の send-command の例では、インスタンスに関する IP 情報を取得します。

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig"
```

出力例については、例 1 を参照してください。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

例 3: 特定のタグを持つインスタンスでコマンドを実行するには

次の send-command 例では、タグキー ENV「」と値「Dev」を持つインスタンスに対してコマンドを実行します。

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig" \  
  --tags "Key=ENV,Value=Dev"
```

```
--targets "Key=tag:ENV,Values=Dev" \  
--document-name "AWS-RunShellScript" \  
--parameters "commands=ifconfig"
```

出力例については、例 1 を参照してください。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

例 4: SNS通知を送信するコマンドを実行するには

次のsend-command例では、すべてのSNS通知イベントと通知Commandタイプの通知を送信するコマンドを実行します。

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig" \  
  --service-role-arn "arn:aws:iam::123456789012:role/SNS_Role" \  
  --notification-config "NotificationArn=arn:aws:sns:us-  
east-1:123456789012:SNSTopicName,NotificationEvents=All,NotificationType=Command"
```

出力例については、例 1 を参照してください。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

例 5: S3 および に出力するコマンドを実行するには CloudWatch

次のsend-command例では、コマンドの詳細を S3 バケットと CloudWatch Logs ロググループに出力するコマンドを実行します。

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig" \  
  --output-s3-bucket-name "s3-bucket-name" \  
  --output-s3-key-prefix "runcommand" \  
  --cloud-watch-output-  
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

出力例については、例 1 を参照してください。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

例 6: タグが異なる複数のインスタンスでコマンドを実行するには

次の send-command の例では、2 つの異なるタグキーと値を持つインスタンスでコマンドを実行します。

```
aws ssm send-command \  
  --document-name "AWS-RunPowerShellScript" \  
  --parameters commands=["echo helloWorld"] \  
  --targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

出力例については、例 1 を参照してください。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

例 7: 同じタグキーを持つ複数のインスタンスをターゲットにするには

次の send-command の例では、タグキーは同じだが異なる値を持つインスタンスにコマンドを実行します。

```
aws ssm send-command \  
  --document-name "AWS-RunPowerShellScript" \  
  --parameters commands=["echo helloWorld"] \  
  --targets Key=tag:Env,Values=Dev,Test
```

出力例については、例 1 を参照してください。

詳細については、「AWS Systems Manager ユーザーガイド」の「[Running Commands Using Systems Manager Run Command](#)」を参照してください。

例 8: 共有ドキュメントを使用するコマンドを実行するには

次の send-command の例では、ターゲットインスタンスで共有ドキュメントを実行します。

```
aws ssm send-command \  
  --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument" \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

出力例については、例 1 を参照してください。

詳細については、AWS 「Systems Manager ユーザーガイド」の [「共有SSMドキュメントの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[SendCommand](#)」の「」を参照してください。

start-associations-once

次の例は、start-associations-once を使用する方法を説明しています。

AWS CLI

関連付けをすぐに 1 回のみ実行するには

次のstart-associations-once例では、指定された関連付けを 1 回だけすぐに実行します。コマンドが成功した場合、出力はありません。

```
aws ssm start-associations-once \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の [「関連付けの履歴の表示」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartAssociationsOnce](#)」の「」を参照してください。

start-automation-execution

次のコード例は、start-automation-execution を使用する方法を示しています。

AWS CLI

例 1: オートメーションドキュメントを実行するには

次の start-automation-execution の例では、オートメーションドキュメントを実行します。

```
aws ssm start-automation-execution \  
  --document-name "AWS-RunShellScript"
```

```
--document-name "AWS-UpdateLinuxAmi" \  
--parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/  
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

出力:

```
{  
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"  
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[オートメーションを手動で実行する](#)」を参照してください。

例 2: 共有オートメシヨンドキュメントを実行するには

次の start-automation-execution の例では、共有オートメシヨンドキュメントを実行します。

```
aws ssm start-automation-execution \  
--document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

出力:

```
{  
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"  
}
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「[共有SSMドキュメントの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartAutomationExecution](#)」の「」を参照してください。

start-change-request-execution

次の例は、start-change-request-execution を使用する方法を説明しています。

AWS CLI

例 1: 変更リクエストを開始するには

次のstart-change-request-execution例では、最小限のオプションを指定して変更リクエストを開始します。

```
aws ssm start-change-request-execution \  
  --change-request-name MyChangeRequest \  
  --document-name AWS-HelloWorldChangeTemplate \  
  --runbooks '[{"DocumentName": "AWS-HelloWorld", "Parameters":  
  {"AutomationAssumeRole": [{"arn:aws:iam:us-east-2:1112223233444:role/  
  MyChangeManagerAssumeRole"}]}]' \  
  --parameters  
  Approver="JohnDoe", ApproverType="IamUser", ApproverSnsTopicArn="arn:aws:sns:us-  
  east-2:1112223233444:MyNotificationTopic"
```

出力:

```
{  
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"  
}
```

例 2: 外部JSONファイルを使用して変更リクエストを開始するには

次のstart-automation-execution例では、JSON ファイルに複数のオプションを指定して変更リクエストを開始します。

```
aws ssm start-change-request-execution \  
  --cli-input-json file://MyChangeRequest.json
```

MyChangeRequest.json の内容:

```
{  
  "ChangeRequestName": "MyChangeRequest",  
  "DocumentName": "AWS-HelloWorldChangeTemplate",  
  "DocumentVersion": "$DEFAULT",  
  "ScheduledTime": "2021-12-30T03:00:00",  
  "ScheduledEndTime": "2021-12-30T03:05:00",  
  "Tags": [  
    {  
      "Key": "Purpose",  
      "Value": "Testing"  
    }  
  ],  
}
```

```

"Parameters": {
  "Approver": [
    "JohnDoe"
  ],
  "ApproverType": [
    "IamUser"
  ],
  "ApproverSnsTopicArn": [
    "arn:aws:sns:us-east-2:111222333444;:MyNotificationTopic"
  ]
},
"Runbooks": [
  {
    "DocumentName": "AWS-HelloWorld",
    "DocumentVersion": "1",
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Parameters": {
      "AutomationAssumeRole": [
        "arn:aws:iam::111222333444:role/MyChangeManagerAssumeRole"
      ]
    }
  }
],
"ChangeDetails": "### Document Name: HelloWorldChangeTemplate\n\n## What does this document do?\n\nThis change template demonstrates the feature set available for creating change templates for Change Manager. This template starts a Runbook workflow for the Automation document called AWS-HelloWorld.\n\n## Input Parameters\n\n* ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for approvers.\n* Approver: (Required) The name of the approver to send this request to.\n* ApproverType: (Required) The type of reviewer.\n  * Allowed Values: IamUser, IamGroup, IamRole, SSOGroup, SSOUser\n\n## Output Parameters\n\nThis document has no outputs \n"
}

```

出力:

```

{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}

```

詳細については、AWS Systems Manager ユーザーガイドの [「変更リクエストの作成」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartChangeRequestExecution](#)」の「」を参照してください。

start-session

次のコード例は、start-session を使用する方法を示しています。

AWS CLI

例 1: Session Manager のセッションを開始するには

この start-session の例では、Session Manager のセッションのためにインスタンスとの接続を確立します。このインタラクティブなコマンドでは、呼び出しを実行するクライアントマシンに Session Manager プラグインがインストールされている必要があることに留意してください。

```
aws ssm start-session \  
  --target "i-1234567890abcdef0"
```

出力:

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

例 2: を使用して Session Manager セッションを開始するには SSH

この例では、 を使用して Session Manager セッションのインスタンスとの接続start-sessionを確立しますSSH。このインタラクティブコマンドでは、Session Manager プラグインを呼び出しを行うクライアントマシンにインストールする必要があり、コマンドは Linux ec2-userのインスタンスなど、EC2インスタンスのデフォルトユーザーを使用することに注意してください。

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

出力:

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

詳細については、「Systems Manager ユーザーガイド」の「[セッションの開始](#)」および「[Session Manager プラグインのインストール](#)」を参照してください。 [AWS CLI AWS](#)

- API 詳細については、AWS CLI 「コマンドリファレンス[StartSession](#)」の「」を参照してください。

stop-automation-execution

次のコード例は、stop-automation-execution を使用する方法を示しています。

AWS CLI

オートメーションの実行を停止するには

次の stop-automation-execution の例では、オートメーションドキュメントを停止します。

```
aws ssm stop-automation-execution
  --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

このコマンドでは何も出力されません。

詳細については、「AWS Systems Manager ユーザーガイド」の「[オートメーションを手動で実行する](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StopAutomationExecution](#)」の「」を参照してください。

terminate-session

次のコード例は、terminate-session を使用する方法を示しています。

AWS CLI

Session Manager セッションを終了するには

この例では、ユーザー「Shirley-Rodriguez」によって作成されたセッション terminate-session を完全に終了し、インスタンス上の Session Manager クライアントと SSM エージェント間のデータ接続を閉じます。

```
aws ssm terminate-session \
  --session-id "Shirley-Rodriguez-07a16060613c408b5"
```

出力:

```
{
  "SessionId": "Shirley-Rodriguez-07a16060613c408b5"
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[セッションの終了](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TerminateSession](#)」の「」を参照してください。

unlabel-parameter-version

次の例は、unlabel-parameter-version を使用する方法を説明しています。

AWS CLI

パラメータラベルを削除するには

次のunlabel-parameter-version例では、指定されたパラメータバージョンから指定されたラベルを削除します。

```
aws ssm unlabel-parameter-version \
  --name "parameterName" \
  --parameter-version "version" \
  --labels "label_1" "label_2" "label_3"
```

出力:

```
{
  "RemovedLabels": [
    "label_1"
    "label_2"
    "label_3"
  ],
  "InvalidLabels": []
}
```

詳細については、Systems Manager ユーザーガイドの「[パラメータラベルの削除 \(AWS CLI\)](#)」を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UnlabelParameterVersion](#)」の「」を参照してください。

update-association-status

次の例は、update-association-status を使用する方法を説明しています。

AWS CLI

関連付けステータスを更新するには

次の update-association-status の例では、インスタンスとドキュメント間の関連付けの関連付けステータスを更新します。

```
aws ssm update-association-status \
  --name "AWS-UpdateSSMAgent" \
  --instance-id "i-1234567890abcdef0" \
  --association-
status "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additional
Config-Needed"
```

出力:

```
{
  "AssociationDescription": {
    "Name": "AWS-UpdateSSMAgent",
    "InstanceId": "i-1234567890abcdef0",
    "AssociationVersion": "1",
    "Date": 1550507529.604,
    "LastUpdateAssociationDate": 1550507806.974,
    "Status": {
      "Date": 1424421071.0,
      "Name": "Pending",
      "Message": "temp_status_change",
      "AdditionalInfo": "Additional-Config-Needed"
    },
  },
  "Overview": {
    "Status": "Success",
    "AssociationStatusAggregatedCount": {
      "Success": 1
    }
  },
  "DocumentVersion": "$DEFAULT",
  "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
  "Targets": [
    {
```

```
        "Key": "InstanceIds",
        "Values": [
            "i-1234567890abcdef0"
        ]
    },
    "LastExecutionDate": 1550507808.0,
    "LastSuccessfulExecutionDate": 1550507808.0
}
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[Systems Manager の関連付けの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAssociationStatus](#)」の「」を参照してください。

update-association

次の例は、update-association を使用する方法を説明しています。

AWS CLI

例 1: ドキュメントの関連付けを更新するには

次の update-association の例では、新しいドキュメントバージョンとの関連付けを更新します。

```
aws ssm update-association \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \  
  --document-version "\$LATEST"
```

出力:

```
{  
  "AssociationDescription": {  
    "Name": "AWS-UpdateSSMAgent",  
    "AssociationVersion": "2",  
    "Date": 1550508093.293,  
    "LastUpdateAssociationDate": 1550508106.596,  
    "Overview": {  
      "Status": "Pending",
```

```

        "DetailedStatus": "Creating"
    },
    "DocumentVersion": "$LATEST",
    "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
    "Targets": [
        {
            "Key": "tag:Name",
            "Values": [
                "Linux"
            ]
        }
    ],
    "LastExecutionDate": 1550508094.879,
    "LastSuccessfulExecutionDate": 1550508094.879
}
}

```

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの編集と新しいバージョンの作成](#)」を参照してください。

例 2: 関連付けのスケジュール式を更新するには

次の update-association の例では、指定された関連付けのスケジュール式を更新します。

```

aws ssm update-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --schedule-expression "cron(0 0 0/4 1/1 * ? *)"

```

出力:

```

{
  "AssociationDescription": {
    "Name": "AWS-HelloWorld",
    "AssociationVersion": "2",
    "Date": "2021-02-08T13:54:19.203000-08:00",
    "LastUpdateAssociationDate": "2021-06-29T11:51:07.933000-07:00",
    "Overview": {
      "Status": "Pending",
      "DetailedStatus": "Creating"
    }
  },
  "DocumentVersion": "$DEFAULT",
  "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
  "Targets": [

```

```
    {
      "Key": "aws:NoOpAutomationTag",
      "Values": [
        "AWS-NoOpAutomationTarget-Value"
      ]
    }
  ],
  "ScheduleExpression": "cron(0 0 0/4 1/1 * ? *)",
  "LastExecutionDate": "2021-06-26T19:00:48.110000-07:00",
  "ApplyOnlyAtCronInterval": false
}
}
```

詳細については、「AWS Systems Manager ユーザーガイド」の「[関連付けの編集と新しいバージョンの作成](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateAssociation](#)」の「」を参照してください。

update-document-default-version

次の例は、update-document-default-version を使用する方法を説明しています。

AWS CLI

ドキュメントのデフォルトバージョンを更新するには

次の update-document-default-version の例では、Systems Manager ドキュメントのデフォルトバージョンを更新します。

```
aws ssm update-document-default-version \
  --name "Example" \
  --document-version "2"
```

出力:

```
{
  "Description": {
    "Name": "Example",
    "DefaultVersion": "2"
  }
}
```

詳細については、AWS「[Systems Manager ユーザーガイド](#)」のSSM「[ドキュメントコンテンツの書き込み](#)」を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスUpdateDocumentDefaultVersion](#)」の「」を参照してください。

update-document-metadata

次の例は、update-document-metadata を使用する方法を説明しています。

AWS CLI

例: 変更テンプレートの最新バージョンを承認するには

以下は、レビューのために送信された変更テンプレートの最新バージョンの承認update-document-metadataを提供します。

```
aws ssm update-document-metadata \  
  --name MyChangeManagerTemplate \  
  --document-reviews 'Action=Approve, Comment=[{Type=Comment, Content=Approved!}]'
```

このコマンドでは何も出力されません。

詳細については、「[Systems Manager ユーザーガイド](#)」の「[変更テンプレートの確認と承認または拒否](#)」を参照してください。AWS

- API 詳細については、AWS CLI「[コマンドリファレンスUpdateDocumentMetadata](#)」の「」を参照してください。

update-document

次のコード例は、update-document を使用する方法を示しています。

AWS CLI

ドキュメントの新しいバージョンを作成するには

次の update-document の例では、Windows コンピュータでの実行時に、ドキュメントの新しいバージョンを作成します。で指定されたドキュメントは JSON形式--documentである必要があります。file:// に続くコンテンツファイルのパスを参照する必要があることに注意してください。--document-version パラメータの先頭に \$ があるため、Windows では値を二重引用

符で囲む必要があります。Linux、MacOS、または PowerShell プロンプトでは、値を一重引用符で囲む必要があります。

Windows のバージョン:

```
aws ssm update-document \  
  --name "RunShellScript" \  
  --content "file://RunShellScript.json" \  
  --document-version "$LATEST"
```

Linux/Mac バージョン:

```
aws ssm update-document \  
  --name "RunShellScript" \  
  --content "file://RunShellScript.json" \  
  --document-version '$LATEST'
```

出力:

```
{  
  "DocumentDescription": {  
    "Status": "Updating",  
    "Hash": "f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b",  
    "Name": "RunShellScript",  
    "Parameters": [  
      {  
        "Type": "StringList",  
        "Name": "commands",  
        "Description": "(Required) Specify a shell script or a command to  
run."  
      }  
    ],  
    "DocumentType": "Command",  
    "PlatformTypes": [  
      "Linux"  
    ],  
    "DocumentVersion": "2",  
    "HashType": "Sha256",  
    "CreateDate": 1487899655.152,  
    "Owner": "809632081692",  
    "SchemaVersion": "2.0",  
    "DefaultVersion": "1",
```



```
    "LatestVersion": "2",
    "Description": "Run an updated script"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDocument](#)」の「」を参照してください。

update-maintenance-window-target

次のコード例は、update-maintenance-window-target を使用する方法を示しています。

AWS CLI

メンテナンスウィンドウターゲットを更新するには

次のupdate-maintenance-window-target例では、メンテナンスウィンドウターゲットの名前のみを更新します。

```
aws ssm update-maintenance-window-target \
  --window-id "mw-0c5ed765acEXAMPLE" \
  --window-target-id "57e8344e-fe64-4023-8191-6bf05EXAMPLE" \
  --name "NewName" \
  --no-replace
```

出力:

```
{
  "Description": "",
  "OwnerInformation": "",
  "WindowTargetId": "57e8344e-fe64-4023-8191-6bf05EXAMPLE",
  "WindowId": "mw-0c5ed765acEXAMPLE",
  "Targets": [
    {
      "Values": [
        "i-1234567890EXAMPLE"
      ],
      "Key": "InstanceIds"
    }
  ],
  "Name": "NewName"
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウの更新 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMaintenanceWindowTarget](#)」の「」を参照してください。

update-maintenance-window-task

次のコード例は、update-maintenance-window-task を使用する方法を示しています。

AWS CLI

メンテナンスウィンドウタスクを更新するには

次のupdate-maintenance-window-task例では、メンテナンスウィンドウタスクのサービスロールを更新します。

```
aws ssm update-maintenance-window-task \  
  --window-id "mw-0c5ed765acEXAMPLE" \  
  --window-task-id "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE" \  
  --service-role-arn "arn:aws:iam::111222333444:role/aws-service-role/  
  ssm.amazonaws.com/AWSServiceRoleForAmazonSSM"
```

出力:

```
{  
  "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/  
  ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",  
  "MaxErrors": "1",  
  "TaskArn": "AWS-UpdateEC2Config",  
  "MaxConcurrency": "1",  
  "WindowTaskId": "23d3809e-9fbe-4ddf-b41a-b49d7EXAMPLE",  
  "TaskParameters": {},  
  "Priority": 1,  
  "TaskInvocationParameters": {  
    "RunCommand": {  
      "TimeoutSeconds": 600,  
      "Parameters": {  
        "allowDowngrade": [  
          "false"  
        ]  
      }  
    }  
  }  
}
```

```
    },
    "WindowId": "mw-0c5ed765acEXAMPLE",
    "Description": "UpdateEC2Config",
    "Targets": [
      {
        "Values": [
          "57e8344e-fe64-4023-8191-6bf05EXAMPLE"
        ],
        "Key": "WindowTargetIds"
      }
    ],
    "Name": "UpdateEC2Config"
  }
}
```

詳細については、AWS Systems Manager ユーザーガイドの「[メンテナンスウィンドウの更新 \(AWS CLI\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMaintenanceWindowTask](#)」の「」を参照してください。

update-maintenance-window

次のコード例は、update-maintenance-window を使用する方法を示しています。

AWS CLI

例 1: メンテナンスウィンドウを更新するには

次の update-maintenance-window の例では、メンテナンスウィンドウの名前を更新します。

```
aws ssm update-maintenance-window \
  --window-id "mw-1a2b3c4d5e6f7g8h9" \
  --name "My-Renamed-MW"
```

出力:

```
{
  "Cutoff": 1,
  "Name": "My-Renamed-MW",
  "Schedule": "cron(0 16 ? * TUE *)",
  "Enabled": true,
  "AllowUnassociatedTargets": true,
```

```
"WindowId": "mw-1a2b3c4d5e6f7g8h9",  
"Duration": 4  
}
```

例 2: メンテナンスウィンドウを無効にするには

次の `update-maintenance-window` の例では、メンテナンスウィンドウを無効にします。

```
aws ssm update-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9" \  
  --no-enabled
```

例 3: メンテナンスウィンドウを有効にするには

次の `update-maintenance-window` の例では、メンテナンスウィンドウを有効にします。

```
aws ssm update-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9" \  
  --enabled
```

詳細については、AWS Systems Manager ユーザーガイドの [「メンテナンスウィンドウの更新 \(AWS CLI\)」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateMaintenanceWindow](#)」の「」を参照してください。

update-managed-instance-role

次のコード例は、`update-managed-instance-role` を使用する方法を示しています。

AWS CLI

マネージドインスタンスの IAM ロールを更新するには

次の `update-managed-instance-role` 例では、マネージド IAM インスタンスのインスタンスプロファイルを更新します。

```
aws ssm update-managed-instance-role \  
  --instance-id "mi-08ab247cdfEXAMPLE" \  
  --iam-role "ExampleRole"
```

このコマンドでは何も出力されません。

詳細については、[「Systems Manager ユーザーガイド」の「ステップ 4: Systems Manager の IAM インスタンスプロファイルを作成する」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateManagedInstanceRole](#)」の「」を参照してください。

update-ops-item

次のコード例は、update-ops-item を使用する方法を示しています。

AWS CLI

を更新するには OpsItem

次のupdate-ops-item例では、の説明、優先度、カテゴリを更新します OpsItem。さらに、コマンドは、これが OpsItem 編集または変更されたときに通知が送信されるSNSトピックを指定します。

```
aws ssm update-ops-item \  
  --ops-item-id "oi-287b5EXAMPLE" \  
  --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \  
  --priority 2 \  
  --category "Security" \  
  --notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

出力:

```
This command produces no output.
```

詳細については、AWS 「Systems Manager ユーザーガイド」の[「での作業 OpsItems」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateOpsItem](#)」の「」を参照してください。

update-patch-baseline

次の例は、update-patch-baseline を使用する方法を説明しています。

AWS CLI

例 1: パッチベースラインを更新するには

次の `update-patch-baseline` の例では、指定された 2 つのパッチを拒否済み、1 つのパッチを承認済みとして、指定されたパッチベースラインに追加します。

```
aws ssm update-patch-baseline \  
  --baseline-id "pb-0123456789abcdef0" \  
  --rejected-patches "KB2032276" "MS10-048" \  
  --approved-patches "KB2124261"
```

出力:

```
{  
  "BaselineId": "pb-0123456789abcdef0",  
  "Name": "WindowsPatching",  
  "OperatingSystem": "WINDOWS",  
  "GlobalFilters": {  
    "PatchFilters": []  
  },  
  "ApprovalRules": {  
    "PatchRules": [  
      {  
        "PatchFilterGroup": {  
          "PatchFilters": [  
            {  
              "Key": "PRODUCT",  
              "Values": [  
                "WindowsServer2016"  
              ]  
            }  
          ]  
        },  
        "ComplianceLevel": "CRITICAL",  
        "ApproveAfterDays": 0,  
        "EnableNonSecurity": false  
      }  
    ]  
  },  
  "ApprovedPatches": [  
    "KB2124261"  
  ],  
  "ApprovedPatchesComplianceLevel": "UNSPECIFIED",  
  "ApprovedPatchesEnableNonSecurity": false,  
  "RejectedPatches": [  
    "KB2032276",
```

```
    "MS10-048"  
  ],  
  "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",  
  "CreateDate": 1550244180.465,  
  "ModifiedDate": 1550244180.465,  
  "Description": "Patches for Windows Servers",  
  "Sources": []  
}
```

例 2: パッチベースラインの名前を変更するには

次の update-patch-baseline の例では、指定されたパッチベースラインの名前を変更します。

```
aws ssm update-patch-baseline \  
  --baseline-id "pb-0713accee01234567" \  
  --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

詳細については、AWS 「Systems Manager ユーザーガイド」の「パッチベースラインを更新または削除する」 「< <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-baseline-update-or-delete.html>>`_」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdatePatchBaseline](#)」の「」を参照してください。

update-resource-data-sync

次の例は、update-resource-data-sync を使用する方法を説明しています。

AWS CLI

リソースデータ同期を更新するには

次のupdate-resource-data-sync例では、 SyncFromSource リソースデータ同期を更新します。

```
aws ssm update-resource-data-sync \  
  --sync-name exampleSync \  
  --sync-type SyncFromSource \  
  --sync-source '{"SourceType": "SingleAccountMultiRegions", "SourceRegions": ["us-east-1", "us-west-2"]}'
```

このコマンドでは何も出力されません。

詳細については、[「Systems Manager ユーザーガイド」の「Systems Manager Explorer をセットアップして、複数のアカウントとリージョンからのデータを表示する」](#)を参照してください。

AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResourceDataSync](#)」の「」を参照してください。

update-service-setting

次のコード例は、update-service-setting を使用する方法を示しています。

AWS CLI

Parameter Store スループットのサービス設定を更新するには

次のupdate-service-setting例では、指定したリージョンの Parameter Store スループットの現在のサービス設定を更新して、スループットの向上を使用します。

```
aws ssm update-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
store/high-throughput-enabled \  
  --setting-value true
```

このコマンドでは何も出力されません。

詳細については、AWS Systems Manager ユーザーガイドの[「パラメータストアスループットの増加」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateServiceSetting](#)」の「」を参照してください。

を使用した Amazon Textract の例 AWS CLI

次のコード例は、Amazon Textract AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

analyze-document

次のコード例は、analyze-document を使用する方法を示しています。

AWS CLI

ドキュメント内のテキストを分析するには

次の analyze-document の例は、ドキュメント内のテキストを分析する方法を示しています。

Linux/macOS:

```
aws textract analyze-document \  
  --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \  
  --feature-types ['TABLES','FORMS']
```

Windows :

```
aws textract analyze-document \  
  --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}\" \  
  --feature-types [\"TABLES\",\"FORMS\"] \  
  --region region-name
```

出力:

```
{  
  "Blocks": [  
    {  
      "Geometry": {  
        "BoundingBox": {  
          "Width": 1.0,  
          "Top": 0.0,  
          "Left": 0.0,  
          "Height": 1.0
```

```
    },
    "Polygon": [
      {
        "Y": 0.0,
        "X": 0.0
      },
      {
        "Y": 0.0,
        "X": 1.0
      },
      {
        "Y": 1.0,
        "X": 1.0
      },
      {
        "Y": 1.0,
        "X": 0.0
      }
    ]
  },
  "Relationships": [
    {
      "Type": "CHILD",
      "Ids": [
        "87586964-d50d-43e2-ace5-8a890657b9a0",
        "a1e72126-21d9-44f4-a8d6-5c385f9002ba",
        "e889d012-8a6b-4d2e-b7cd-7a8b327d876a"
      ]
    }
  ],
  "BlockType": "PAGE",
  "Id": "c2227f12-b25d-4e1f-baea-1ee180d926b2"
}
],
"DocumentMetadata": {
  "Pages": 1
}
}
```

詳細については、「Amazon Textract デベロッパーガイド」の「Amazon Textract を使用したドキュメントテキストの分析」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [AnalyzeDocument](#)」の「」を参照してください。

detect-document-text

次の例は、detect-document-text を使用する方法を説明しています。

AWS CLI

ドキュメント内のテキストを検出するには

次の detect-document-text の例は、ドキュメント内のテキストを検出する方法を示しています。

Linux/macOS:

```
aws textract detect-document-text \  
  --document '{"S3Object":{"Bucket":"bucket","Name":"document"}}'
```

Windows :

```
aws textract detect-document-text \  
  --document "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}" \  
  --region region-name
```

出力:

```
{  
  "Blocks": [  
    {  
      "Geometry": {  
        "BoundingBox": {  
          "Width": 1.0,  
          "Top": 0.0,  
          "Left": 0.0,  
          "Height": 1.0  
        },  
        "Polygon": [  
          {  
            "Y": 0.0,  
            "X": 0.0  
          },  
          {  
            "Y": 0.0,  
            "X": 1.0  
          }  
        ]  
      }  
    ]  
  }
```

```
        {
            "Y": 1.0,
            "X": 1.0
        },
        {
            "Y": 1.0,
            "X": 0.0
        }
    ]
},
"Relationships": [
    {
        "Type": "CHILD",
        "Ids": [
            "896a9f10-9e70-4412-81ce-49ead73ed881",
            "0da18623-dc4c-463d-a3d1-9ac050e9e720",
            "167338d7-d38c-4760-91f1-79a8ec457bb2"
        ]
    }
],
"BlockType": "PAGE",
"Id": "21f0535e-60d5-4bc7-adf2-c05dd851fa25"
},
{
    "Relationships": [
        {
            "Type": "CHILD",
            "Ids": [
                "62490c26-37ea-49fa-8034-7a9ff9369c9c",
                "1e4f3f21-05bd-4da9-ba10-15d01e66604c"
            ]
        }
    ],
    "Confidence": 89.11581420898438,
    "Geometry": {
        "BoundingBox": {
            "Width": 0.33642634749412537,
            "Top": 0.17169663310050964,
            "Left": 0.13885067403316498,
            "Height": 0.49159330129623413
        },
        "Polygon": [
            {
                "Y": 0.17169663310050964,
```

```
        "X": 0.13885067403316498
      },
      {
        "Y": 0.17169663310050964,
        "X": 0.47527703642845154
      },
      {
        "Y": 0.6632899641990662,
        "X": 0.47527703642845154
      },
      {
        "Y": 0.6632899641990662,
        "X": 0.13885067403316498
      }
    ]
  },
  "Text": "Hello,",
  "BlockType": "LINE",
  "Id": "896a9f10-9e70-4412-81ce-49ead73ed881"
},
{
  "Relationships": [
    {
      "Type": "CHILD",
      "Ids": [
        "19b28058-9516-4352-b929-64d7cef29daf"
      ]
    }
  ]
},
"Confidence": 85.5694351196289,
"Geometry": {
  "BoundingBox": {
    "Width": 0.33182239532470703,
    "Top": 0.23131252825260162,
    "Left": 0.5091826915740967,
    "Height": 0.3766750991344452
  },
  "Polygon": [
    {
      "Y": 0.23131252825260162,
      "X": 0.5091826915740967
    },
    {
      "Y": 0.23131252825260162,
```

```
        "X": 0.8410050868988037
      },
      {
        "Y": 0.607987642288208,
        "X": 0.8410050868988037
      },
      {
        "Y": 0.607987642288208,
        "X": 0.5091826915740967
      }
    ]
  },
  "Text": "worlc",
  "BlockType": "LINE",
  "Id": "0da18623-dc4c-463d-a3d1-9ac050e9e720"
}
],
"DocumentMetadata": {
  "Pages": 1
}
}
```

詳細については、「Amazon Textract デベロッパーガイド」の「ドキュメントのテキストの検出」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DetectDocumentText](#)」の「」を参照してください。

get-document-analysis

次の例は、get-document-analysis を使用する方法を説明しています。

AWS CLI

複数ページのドキュメントの非同期テキスト分析の結果を取得するには

次の get-document-analysis の例は、複数ページのドキュメントの非同期テキスト分析の結果を取得します。

```
aws textract get-document-analysis \
  --job-id df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b \
  --max-results 1000
```

出力:

```
{
  "Blocks": [
    {
      "Geometry": {
        "BoundingBox": {
          "Width": 1.0,
          "Top": 0.0,
          "Left": 0.0,
          "Height": 1.0
        },
        "Polygon": [
          {
            "Y": 0.0,
            "X": 0.0
          },
          {
            "Y": 0.0,
            "X": 1.0
          },
          {
            "Y": 1.0,
            "X": 1.0
          },
          {
            "Y": 1.0,
            "X": 0.0
          }
        ]
      },
      "Relationships": [
        {
          "Type": "CHILD",
          "Ids": [
            "75966e64-81c2-4540-9649-d66ec341cd8f",
            "bb099c24-8282-464c-a179-8a9fa0a057f0",
            "5ebf522d-f9e4-4dc7-bfae-a288dc094595"
          ]
        }
      ],
      "BlockType": "PAGE",
      "Id": "247c28ee-b63d-4aeb-9af0-5f7ea8ba109e",
      "Page": 1
    }
  ]
}
```

```
    }
  ],
  "NextToken": "cY1W3eTFvoB0cH7YrKVudI4Gb0H8J0xAYLo8xI/JunCIPWCthaKQ+07n/
ElyutsSy0+1V0ImoTRmP1zw4P0RFtaeV9Bzhnfedpx1YqwB4xaGDA==",
  "DocumentMetadata": {
    "Pages": 1
  },
  "JobStatus": "SUCCEEDED"
}
```

詳細については、「Amazon Textract デベロッパーガイド」の「複数ページドキュメント内のテキストの検出または分析」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDocumentAnalysis](#)」の「」を参照してください。

get-document-text-detection

次のコード例は、get-document-text-detection を使用する方法を示しています。

AWS CLI

マルチページドキュメントで非同期テキスト検出の結果を取得するには

次のget-document-text-detection例は、複数ページのドキュメントで非同期テキスト検出の結果を取得する方法を示しています。

```
aws textract get-document-text-detection \
  --job-id 57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9 \
  --max-results 1000
```

出力

```
{
  "Blocks": [
    {
      "Geometry": {
        "BoundingBox": {
          "Width": 1.0,
          "Top": 0.0,
          "Left": 0.0,
          "Height": 1.0
        },

```



```
    "Polygon": [
      {
        "Y": 0.0,
        "X": 0.0
      },
      {
        "Y": 0.0,
        "X": 1.0
      },
      {
        "Y": 1.0,
        "X": 1.0
      },
      {
        "Y": 1.0,
        "X": 0.0
      }
    ],
    "Relationships": [
      {
        "Type": "CHILD",
        "Ids": [
          "1b926a34-0357-407b-ac8f-ec473160c6a9",
          "0c35dc17-3605-4c9d-af1a-d9451059df51",
          "dea3db8a-52c2-41c0-b50c-81f66f4aa758"
        ]
      }
    ],
    "BlockType": "PAGE",
    "Id": "84671a5e-8c99-43be-a9d1-6838965da33e",
    "Page": 1
  },
  "NextToken": "GcqyoAJuZwuj0T35EN4LCI3EUzMtiLq3nKyFFHvU5q1SaIdEBcSty+njNgoWwuMP/
muqc96S4o5NzDqehhXvhkodMyV050JGyms5lSrCxibWJw==",
  "DocumentMetadata": {
    "Pages": 1
  },
  "JobStatus": "SUCCEEDED"
}
```

詳細については、「Amazon Textract デベロッパーガイド」の「複数ページドキュメント内のテキストの検出または分析」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDocumentTextDetection](#)」の「」を参照してください。

start-document-analysis

次の例は、start-document-analysis を使用する方法を説明しています。

AWS CLI

複数ページのドキュメント内のテキストの分析を開始するには

次の start-document-analysis の例は、複数ページのドキュメントの非同期テキスト分析を開始する方法を示しています。

Linux/macOS:

```
aws textract start-document-analysis \  
  --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \  
  --feature-types ['TABLES','FORMS'] \  
  --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Windows :

```
aws textract start-document-analysis \  
  --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}\" \  
  --feature-types [\"TABLES\", \"FORMS\"] \  
  --region region-name \  
  --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

出力:

```
{  
  "JobId": "df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b"  
}
```

詳細については、「Amazon Textract デベロッパーガイド」の「複数ページドキュメント内のテキストの検出または分析」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartDocumentAnalysis](#)」の「」を参照してください。

start-document-text-detection

次の例は、start-document-text-detection を使用する方法を説明しています。

AWS CLI

複数ページのドキュメント内のテキストの検出を開始するには

次の start-document-text-detection の例は、複数ページのドキュメントの非同期テキスト検出を開始する方法を示しています。

Linux/macOS:

```
aws textract start-document-text-detection \  
  --document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \  
  --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleARN"
```

Windows :

```
aws textract start-document-text-detection \  
  --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\"}}\" \  
  --region region-name \  
  --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleARN"
```

出力:

```
{  
  "JobId": "57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9"  
}
```

詳細については、「Amazon Textract デベロッパーガイド」の「複数ページドキュメント内のテキストの検出または分析」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartDocumentTextDetection](#)」の「」を参照してください。

を使用した Amazon Transcribe の例 AWS CLI

次のコード例は、Amazon Transcribe AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-language-model

次のコード例は、create-language-model を使用する方法を示しています。

AWS CLI

例 1: トレーニングデータとチューニングデータの両方を使用してカスタム言語モデルを作成するには。

次のcreate-language-model例では、カスタム言語モデルを作成します。カスタム言語モデルを使用して、法律、厚遇、財務、保険などのドメインの文字起こしパフォーマンスを向上させることができます。言語コードには、有効な言語コードを入力します。では base-model-name、カスタム言語モデルで文字起こしするオーディオのサンプルレートに最適なベースモデルを指定します。model-name には、カスタム言語モデルを呼び出す名前を指定します。

```
aws transcribe create-language-model \
  --language-code language-code \
  --base-model-name base-model-name \
  --model-name cli-clm-example \
  --input-data-config S3Uri="s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-for-training-data",TuningDataS3Uri="s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-for-tuning-data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-with-permissions-to-create-a-custom-language-model"
```

出力:

```
{
  "LanguageCode": "language-code",
  "BaseModelName": "base-model-name",
  "ModelName": "cli-clm-example",
  "InputDataConfig": {
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",
    "TuningDataS3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",
    "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-with-
permissions-create-a-custom-language-model"
  },
  "ModelStatus": "IN_PROGRESS"
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[カスタム言語モデルを使用したドメイン固有のトランスクリプション精度の向上](#)」を参照してください。

例 2: トレーニングデータのみを使用してカスタム言語モデルを作成するには。

次の `create-language-model` の例は、音声ファイルの文字起こしを行います。カスタム言語モデルを使用して、法律、厚遇、財務、保険などのドメインの文字起こしパフォーマンスを向上させることができます。言語コードには、有効な言語コードを入力します。では `base-model-name`、カスタム言語モデルで文字起こしするオーディオのサンプルレートに最適なベースモデルを指定します。 `model-name` には、カスタム言語モデルを呼び出す名前を指定します。

```
aws transcribe create-language-model \
  --language-code en-US \
  --base-model-name base-model-name \
  --model-name cli-clm-example \
  --input-data-config S3Uri="s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-For-
Training-Data",DataAccessRoleArn="arn:aws:iam::AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
```

出力:

```
{
  "LanguageCode": "en-US",
  "BaseModelName": "base-model-name",
  "ModelName": "cli-clm-example",
  "InputDataConfig": {
```

```
"S3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix-For-Training-Data/",
  "DataAccessRoleArn": "arn:aws:iam::your-AWS-account-number:role/IAM-role-
with-permissions-to-create-a-custom-language-model"
},
"ModelStatus": "IN_PROGRESS"
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[カスタム言語モデルを使用したドメイン固有のトランスクリプション精度の向上](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLanguageModel](#)」の「」を参照してください。

create-medical-vocabulary

次のコード例は、create-medical-vocabulary を使用する方法を示しています。

AWS CLI

医療カスタム語彙を作成するには

次の create-medical-vocabulary 例は、カスタム語彙を作成します。カスタム語彙を作成するには、より正確に書き起こすべき用語のすべてを含むテキストファイルを作成しておく必要があります。には vocabulary-file-uri、URIそのテキストファイルの Amazon Simple Storage Service (Amazon S3) を指定します。language-code として、カスタム語彙の言語に対応する言語コードを指定します。vocabulary-name として、カスタムボキャブラリーに付ける名前を指定します。

```
aws transcribe create-medical-vocabulary \
  --vocabulary-name cli-medical-vocab-example \
  --language-code language-code \
  --vocabulary-file-uri https://DOC-EXAMPLE-BUCKET.AWS-Region.amazonaws.com/the-
text-file-for-the-medical-custom-vocabulary.txt
```

出力:

```
{
  "VocabularyName": "cli-medical-vocab-example",
  "LanguageCode": "language-code",
  "VocabularyState": "PENDING"
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[医療カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateMedicalVocabulary](#)」の「」を参照してください。

create-vocabulary-filter

次のコード例は、create-vocabulary-filter を使用する方法を示しています。

AWS CLI

語彙フィルターを作成するには

次のcreate-vocabulary-filter例では、文字起こしには表示しない単語のリストを含むテキストファイルを使用する語彙フィルターを作成します。言語コードには、語彙フィルターの言語に対応する言語コードを指定します。には vocabulary-filter-file-uri、URIテキストファイルの Amazon Simple Storage Service (Amazon S3) を指定します。には vocabulary-filter-name、語彙フィルターの名前を指定します。

```
aws transcribe create-vocabulary-filter \  
  --language-code language-code \  
  --vocabulary-filter-file-uri s3://DOC-EXAMPLE-BUCKET/vocabulary-filter.txt \  
  --vocabulary-filter-name cli-vocabulary-filter-example
```

出力:

```
{  
  "VocabularyFilterName": "cli-vocabulary-filter-example",  
  "LanguageCode": "language-code"  
}
```

詳細については、「[Amazon Transcribe デベロッパーガイド](#)」の「[不要な単語のフィルタリング](#)」を参照してください。 Amazon Transcribe

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateVocabularyFilter](#)」の「」を参照してください。

create-vocabulary

次の例は、create-vocabulary を使用する方法を説明しています。

AWS CLI

カスタム語彙を作成するには

次の `create-vocabulary` 例は、カスタム語彙を作成します。カスタム語彙を作成するには、より正確に書き起こすべき用語のすべてを含むテキストファイルを作成しておく必要があります。には `vocabulary-file-uri`、URI そのテキストファイルの Amazon Simple Storage Service (Amazon S3) を指定します。 `language-code` として、カスタム語彙の言語に対応する言語コードを指定します。 `vocabulary-name` として、カスタムボキャブラリーに付ける名前を指定します。

```
aws transcribe create-vocabulary \  
  --language-code language-code \  
  --vocabulary-name cli-vocab-example \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/the-text-file-for-the-custom-vocabulary.txt
```

出力:

```
{  
  "VocabularyName": "cli-vocab-example",  
  "LanguageCode": "language-code",  
  "VocabularyState": "PENDING"  
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateVocabulary](#)」の「」を参照してください。

`delete-language-model`

次の例は、`delete-language-model` を使用する方法を説明しています。

AWS CLI

カスタム言語モデルを削除するには

次の `delete-language-model` 例では、カスタム言語モデルを削除します。

```
aws transcribe delete-language-model \  
  --language-model-name cli-lang-model-example
```



```
--model-name model-name
```

このコマンドでは何も出力されません。

詳細については、「Amazon Transcribe 開発者ガイド」の「[カスタム言語モデルを使用したドメイン固有のトランスクリプション精度の向上](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLanguageModel](#)」の「」を参照してください。

delete-medical-transcription-job

次のコード例は、delete-medical-transcription-job を使用する方法を示しています。

AWS CLI

医療文字起こしジョブを削除するには

次の delete-medical-transcription-job の例は、医療文字起こしジョブを削除します。

```
aws transcribe delete-medical-transcription-job \  
  --medical-transcription-job-name medical-transcription-job-name
```

このコマンドでは何も出力されません。

詳細については、Amazon Transcribe デベロッパーガイド[DeleteMedicalTranscriptionJob](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMedicalTranscriptionJob](#)」の「」を参照してください。

delete-medical-vocabulary

次のコード例は、delete-medical-vocabulary を使用する方法を示しています。

AWS CLI

医療カスタム語彙を削除するには

次のdelete-medical-vocabulary例では、医療カスタム語彙を削除します。語彙名には、医療カスタム語彙の名前を指定します。

```
aws transcribe delete-vocabulary \  
  --vocabulary-name medical-custom-vocabulary-name
```

このコマンドでは何も出力されません。

詳細については、「Amazon Transcribe 開発者ガイド」の「[医療カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMedicalVocabulary](#)」の「」を参照してください。

delete-transcription-job

次のコード例は、delete-transcription-job を使用する方法を示しています。

AWS CLI

文字起こしジョブの 1 つを削除するには

次の delete-transcription-job 例では、トランスクリプションジョブの 1 つを削除します。

```
aws transcribe delete-transcription-job \  
  --transcription-job-name your-transcription-job
```

このコマンドでは何も出力されません。

詳細については、Amazon Transcribe デベロッパーガイド[DeleteTranscriptionJob](#)」の「」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTranscriptionJob](#)」の「」を参照してください。

delete-vocabulary-filter

次のコード例は、delete-vocabulary-filter を使用する方法を示しています。

AWS CLI

語彙フィルターを削除するには

次のdelete-vocabulary-filter例では、語彙フィルターを削除します。

```
aws transcribe delete-vocabulary-filter \  
  --vocabulary-filter-name vocabulary-filter-name
```

このコマンドでは何も出力されません。

詳細については、[「Amazon Transcribe デベロッパーガイド」の「不要な単語のフィルタリング」](#)を参照してください。 Amazon Transcribe

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVocabularyFilter](#)」の「」を参照してください。

delete-vocabulary

次の例は、delete-vocabulary を使用する方法を説明しています。

AWS CLI

カスタム語彙を削除するには

次の delete-vocabulary の例は、カスタム語彙を削除します。

```
aws transcribe delete-vocabulary \  
  --vocabulary-name vocabulary-name
```

このコマンドでは何も出力されません。

詳細については、「Amazon Transcribe デベロッパーガイド」の「[カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteVocabulary](#)」の「」を参照してください。

describe-language-model

次のコード例は、describe-language-model を使用する方法を示しています。

AWS CLI

特定のカスタム言語モデルに関する情報を取得するには

次のdescribe-language-model例では、特定のカスタム言語モデルに関する情報を取得します。例えば、BaseModelNameの下には、モデルが NarrowBand または WideBand モデルを使用してトレーニングされているかどうかを確認できます。NarrowBand ベースモデルを持つカスタム言語モデルは、サンプルレートが 16 未満の音声を文字起こしできませんkHz。WideBand ベースモデルを使用する言語モデルは、サンプルレートが 16 を超えるオーディオを文字起こしできませんkHz。S3Uri パラメータは、トレーニングデータにアクセスしてカスタム言語モデルを作成するために使用した Amazon S3 プレフィックスを示します。

```
aws transcribe describe-language-model \  
  --model-name cli-clm-example
```

出力:

```
{  
  "LanguageModel": {  
    "ModelName": "cli-clm-example",  
    "CreateTime": "2020-09-25T17:57:38.504000+00:00",  
    "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",  
    "LanguageCode": "language-code",  
    "BaseModelName": "base-model-name",  
    "ModelStatus": "IN_PROGRESS",  
    "UpgradeAvailability": false,  
    "InputDataConfig": {  
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",  
      "TuningDataS3Uri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/",  
      "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-with-permissions-to-create-a-custom-language-model"  
    }  
  }  
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[カスタム言語モデルを使用したドメイン固有のトランスクリプション精度の向上](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeLanguageModel](#)」の「」を参照してください。

get-medical-transcription-job

次のコード例は、get-medical-transcription-job を使用する方法を示しています。

AWS CLI

特定の医療文字起こしジョブに関する情報を取得するには

次のget-medical-transcription-job例では、特定の医療文字起こしジョブに関する情報を取得します。文字起こし結果にアクセスするには、TranscriptFileUri パラメータを使用します。文字起こしジョブの追加機能を有効にしている場合は、設定オブジェクトで確認できます。Specialty パラメータは、プロバイダーの医療専門分野を示します。Type パラメータは、文字起こしジョブの音声か医療会話か医療ディクテーションかを示します。

```
aws transcribe get-medical-transcription-job \  
  --medical-transcription-job-name vocabulary-dictation-medical-transcription-job
```

出力:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-  
job",  
    "TranscriptionJobStatus": "COMPLETED",  
    "LanguageCode": "en-US",  
    "MediaSampleRateHertz": 48000,  
    "MediaFormat": "mp4",  
    "Media": {  
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-audio-file.file-extension"  
    },  
    "Transcript": {  
      "TranscriptFileUri": "https://s3.Region.amazonaws.com/Amazon-S3-Prefix/  
vocabulary-dictation-medical-transcription-job.json"  
    },  
    "StartTime": "2020-09-21T21:17:27.045000+00:00",  
    "CreationTime": "2020-09-21T21:17:27.016000+00:00",  
    "CompletionTime": "2020-09-21T21:17:59.561000+00:00",  
    "Settings": {  
      "ChannelIdentification": false,  
      "ShowAlternatives": false,  
      "VocabularyName": "cli-medical-vocab-example"  
    },  
    "Specialty": "PRIMARYCARE",  
    "Type": "DICTATION"  
  }  
}
```

詳細については、「Amazon [Transcribe デベロッパーガイド](#)」の「[バッチ文字起こし](#)」を参照してください。 Amazon Transcribe

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMedicalTranscriptionJob](#)」の「」を参照してください。

get-medical-vocabulary

次の例は、get-medical-vocabulary を使用する方法を説明しています。

AWS CLI

医療カスタム語彙に関する情報を取得するには

次のget-medical-vocabulary例では、医療カスタム語彙に関する情報を取得します。VocabularyState パラメータを使用して、語彙の処理状態を確認できます。の場合はREADY、StartMedicalTranscriptionJob オペレーションで使用できます。

```
aws transcribe get-medical-vocabulary \  
  --vocabulary-name medical-vocab-example
```

出力:

```
{  
  "VocabularyName": "medical-vocab-example",  
  "LanguageCode": "en-US",  
  "VocabularyState": "READY",  
  "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",  
  "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-  
  medical-custom-vocabulary"  
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[医療カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMedicalVocabulary](#)」の「」を参照してください。

get-transcription-job

次の例は、get-transcription-job を使用する方法を説明しています。

AWS CLI

特定の文字起こしジョブに関する情報を取得するには

次の `get-transcription-job` 例では、特定の文字起こしジョブに関する情報を取得します。文字起こし結果にアクセスするには、`TranscriptFileUri` パラメータを使用します。`MediaFileUri` パラメータを使用して、このジョブで文字起こししたオーディオファイルを確認します。`Settings` オブジェクトを使用して、文字起こしジョブで有効にしたオプション機能を確認できます。

```
aws transcribe get-transcription-job \  
  --transcription-job-name your-transcription-job
```

出力:

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "your-transcription-job",  
    "TranscriptionJobStatus": "COMPLETED",  
    "LanguageCode": "language-code",  
    "MediaSampleRateHertz": 48000,  
    "MediaFormat": "mp4",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"  
    },  
    "Transcript": {  
      "TranscriptFileUri": "https://Amazon-S3-file-location-of-transcription-  
output"  
    },  
    "StartTime": "2020-09-18T22:27:23.970000+00:00",  
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",  
    "CompletionTime": "2020-09-18T22:28:21.197000+00:00",  
    "Settings": {  
      "ChannelIdentification": false,  
      "ShowAlternatives": false  
    },  
    "IdentifyLanguage": true,  
    "IdentifiedLanguageScore": 0.8672199249267578  
  }  
}
```

詳細については、[Amazon Transcribe デベロッパーガイド](#)の「[入門 \(AWS コマンドラインインターフェイス\)](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTranscriptionJob](#)」の「」を参照してください。

get-vocabulary-filter

次のコード例は、get-vocabulary-filter を使用する方法を示しています。

AWS CLI

語彙フィルターに関する情報を取得するには

次のget-vocabulary-filter例では、語彙フィルターに関する情報を取得します。DownloadUri パラメータを使用して、語彙フィルターの作成に使用した単語のリストを取得できます。

```
aws transcribe get-vocabulary-filter \  
  --vocabulary-filter-name testFilter
```

出力:

```
{  
  "VocabularyFilterName": "testFilter",  
  "LanguageCode": "language-code",  
  "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00",  
  "DownloadUri": "https://Amazon-S3-location-to-download-your-vocabulary-filter"  
}
```

詳細については、[Amazon Transcribe デベロッパーガイド](#)の「[不要な単語をフィルタリングする](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVocabularyFilter](#)」の「」を参照してください。

get-vocabulary

次の例は、get-vocabulary を使用する方法を説明しています。

AWS CLI

カスタム語彙に関する情報を取得するには

次の `get-vocabulary` 例では、以前に作成したカスタム語彙に関する情報を取得します。

```
aws transcribe get-vocabulary \  
  --vocabulary-name cli-vocab-1
```

出力:

```
{  
  "VocabularyName": "cli-vocab-1",  
  "LanguageCode": "language-code",  
  "VocabularyState": "READY",  
  "LastModifiedTime": "2020-09-19T23:22:32.836000+00:00",  
  "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-  
  custom-vocabulary"  
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetVocabulary](#)」の「」を参照してください。

list-language-models

次のコード例は、`list-language-models` を使用する方法を示しています。

AWS CLI

カスタム言語モデルを一覧表示するには

次の `list-language-models` 例は、AWS アカウントとリージョンに関連付けられたカスタム言語モデルを一覧表示します。 `S3Uri` および `TuningDataS3Uri` パラメータを使用して、トレーニングデータまたはチューニングデータとして使用した Amazon S3 プレフィックスを検索できます。は、カスタム言語 WideBand モデルを作成するために NarrowBand、または モデルを使用したかどうか `BaseModelName` を示します。 NarrowBand ベースモデルを使用して、カスタム言語モデル kHz でサンプルレートが 16 未満の音声を文字起こしできます。 WideBand ベースモデルを使用して、カスタム言語モデルでオーディオ 16 kHz 以降を文字起こしできま

す。ModelStatus パラメータは、文字起こしジョブでカスタム言語モデルを使用できるかどうかを示します。値が の場合COMPLETED、文字起こしジョブで使用できます。

```
aws transcribe list-language-models
```

出力:

```
{
  "Models": [
    {
      "ModelName": "cli-clm-2",
      "CreateTime": "2020-09-25T17:57:38.504000+00:00",
      "LastModifiedTime": "2020-09-25T17:57:48.585000+00:00",
      "LanguageCode": "language-code",
      "BaseModelName": "WideBand",
      "ModelStatus": "IN_PROGRESS",
      "UpgradeAvailability": false,
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-training-data/",
        "TuningDataS3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-tuning-data/",
        "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-used-to-create-the-custom-language-model"
      }
    },
    {
      "ModelName": "cli-clm-1",
      "CreateTime": "2020-09-25T17:16:01.835000+00:00",
      "LastModifiedTime": "2020-09-25T17:16:15.555000+00:00",
      "LanguageCode": "language-code",
      "BaseModelName": "WideBand",
      "ModelStatus": "IN_PROGRESS",
      "UpgradeAvailability": false,
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-training-data/",
        "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-used-to-create-the-custom-language-model"
      }
    },
    {
      "ModelName": "clm-console-1",
      "CreateTime": "2020-09-24T19:26:28.076000+00:00",
      "LastModifiedTime": "2020-09-25T04:25:22.271000+00:00",
      "LanguageCode": "language-code",
```

```

    "BaseModelName": "NarrowBand",
    "ModelStatus": "COMPLETED",
    "UpgradeAvailability": false,
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-BUCKET/clm-training-data/",
      "DataAccessRoleArn": "arn:aws:iam::AWS-account-number:role/IAM-role-used-to-create-the-custom-language-model"
    }
  }
]
}

```

詳細については、「Amazon Transcribe 開発者ガイド」の「[カスタム言語モデルを使用したドメイン固有のトランスクリプション精度の向上](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLanguageModels](#)」の「」を参照してください。

list-medical-transcription-jobs

次のコード例は、list-medical-transcription-jobs を使用する方法を示しています。

AWS CLI

医療文字起こしジョブを一覧表示するには

次のlist-medical-transcription-jobs例では、AWS アカウントとリージョンに関連付けられた医療文字起こしジョブを一覧表示します。特定の文字起こしジョブに関する詳細情報を取得するには、文字起こし出力の MedicalTranscriptionJobName パラメータの値をコピーし、get-medical-transcription-jobコマンドMedicalTranscriptionJobNameのオプションにその値を指定します。文字起こしジョブをさらに表示するには、NextToken パラメータの値をコピーし、list-medical-transcription-jobsコマンドを再度実行して、--next-tokenオプションでその値を指定します。

```
aws transcribe list-medical-transcription-jobs
```

出力:

```

{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/gsuUSTIkGyanvRE6WMXfd/ZTEc2EZj+P9eii/"
}

```

```

z102FDY1i6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqLZB10Wa9McC+eb01
+LaDtZPC4u6ttoHLRL1EfzqstHXSgapXg3tEBtm9piIaPB6M0M5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD21VIan
+vfUs3gBYA15VTDmXXzQPBOHPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSznzq49Q9xHgHdT2yBADFmvFK7myZBsJ75+2vQZOSVpWUPy3WT/32zFAcoEL
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSjFuwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWm7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDY0X3tTPW1D7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
Wff934aKqfD6UTggm/zV3GA0WiBpFvAZRvEb924i6yGHYMC7y5401ZAwSBupmI
+FFd13CaP04kN1vJlth6aM5vUPXg4BpyUhtbRhWd/KxCvf9K0tLJGyL1A=="
  "MedicalTranscriptionJobSummaries": [
    {
      "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:17:27.016000+00:00",
      "StartTime": "2020-09-21T21:17:27.045000+00:00",
      "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
    {
      "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:01:14.569000+00:00",
      "StartTime": "2020-09-21T21:01:14.592000+00:00",
      "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
    {
      "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
      "CreationTime": "2020-09-21T19:09:18.171000+00:00",
      "StartTime": "2020-09-21T19:09:18.199000+00:00",

```

```

    "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  },
  {
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",
    "StartTime": "2020-09-21T18:43:37.265000+00:00",
    "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  },
  {
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-20T23:46:44.053000+00:00",
    "StartTime": "2020-09-20T23:46:44.081000+00:00",
    "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
]
}

```

詳細については、「Amazon Transcribeデベロッパーガイド」の <https://docs.aws.amazon.com/transcribe/latest/dg/batch「-med-transcription.html」> を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMedicalTranscriptionJobs](#)」の「」を参照してください。

list-medical-vocabularies

次の例は、list-medical-vocabularies を使用する方法を説明しています。

AWS CLI

医療カスタム語彙を一覧表示するには

次のlist-medical-vocabularies例は、AWS アカウントとリージョンに関連付けられた医療カスタム語彙を一覧表示します。特定の文字起こしジョブに関する詳細情報を取得するには、文字起こし出力のMedicalTranscriptionJobNameパラメータの値をコピーし、get-medical-transcription-jobコマンドMedicalTranscriptionJobNameのオプションにその値を指定します。文字起こしジョブをさらに表示するには、NextTokenパラメータの値をコピーし、list-medical-transcription-jobsコマンドを再度実行して、--next-tokenオプションでその値を指定します。

```
aws transcribe list-medical-vocabularies
```

出力:

```
{
  "Vocabularies": [
    {
      "VocabularyName": "cli-medical-vocab-2",
      "LanguageCode": "en-US",
      "LastModifiedTime": "2020-09-21T21:44:59.521000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "cli-medical-vocab-1",
      "LanguageCode": "en-US",
      "LastModifiedTime": "2020-09-19T23:59:04.349000+00:00",
      "VocabularyState": "READY"
    }
  ]
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[医療カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMedicalVocabularies](#)」の「」を参照してください。

list-transcription-jobs

次の例は、list-transcription-jobs を使用する方法を説明しています。

AWS CLI

文字起こしジョブを一覧表示するには

次のlist-transcription-jobs例では、AWS アカウントとリージョンに関連付けられた文字起こしジョブを一覧表示します。

```
aws transcribe list-transcription-jobs
```

出力:

```
{
  "NextToken": "NextToken",
  "TranscriptionJobSummaries": [
    {
      "TranscriptionJobName": "speak-id-job-1",
      "CreationTime": "2020-08-17T21:06:15.391000+00:00",
      "StartTime": "2020-08-17T21:06:15.416000+00:00",
      "CompletionTime": "2020-08-17T21:07:05.098000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    },
    {
      "TranscriptionJobName": "job-1",
      "CreationTime": "2020-08-17T20:50:24.207000+00:00",
      "StartTime": "2020-08-17T20:50:24.230000+00:00",
      "CompletionTime": "2020-08-17T20:52:18.737000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    },
    {
      "TranscriptionJobName": "sdk-test-job-4",
      "CreationTime": "2020-08-17T20:32:27.917000+00:00",
      "StartTime": "2020-08-17T20:32:27.956000+00:00",
      "CompletionTime": "2020-08-17T20:33:15.126000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",

```

```
    "OutputLocationType": "SERVICE_BUCKET"
  },
  {
    "TranscriptionJobName": "Diarization-speak-id",
    "CreationTime": "2020-08-10T22:10:09.066000+00:00",
    "StartTime": "2020-08-10T22:10:09.116000+00:00",
    "CompletionTime": "2020-08-10T22:26:48.172000+00:00",
    "LanguageCode": "language-code",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "SERVICE_BUCKET"
  },
  {
    "TranscriptionJobName": "your-transcription-job-name",
    "CreationTime": "2020-07-29T17:45:09.791000+00:00",
    "StartTime": "2020-07-29T17:45:09.826000+00:00",
    "CompletionTime": "2020-07-29T17:46:20.831000+00:00",
    "LanguageCode": "language-code",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "SERVICE_BUCKET"
  }
]
}
```

詳細については、[Amazon Transcribe デベロッパーガイド](#)の「[入門 \(AWS コマンドラインインターフェイス\)](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスListTranscriptionJobs](#)」の「」を参照してください。

list-vocabularies

次の例は、list-vocabularies を使用する方法を説明しています。

AWS CLI

カスタム語彙を一覧表示するには

次のlist-vocabularies例では、AWS アカウントとリージョンに関連付けられたカスタム語彙を一覧表示します。

```
aws transcribe list-vocabularies
```

出力:


```
{
  "NextToken": "NextToken",
  "Vocabularies": [
    {
      "VocabularyName": "ards-test-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-27T22:00:27.330000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "sample-test",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T23:04:11.044000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "CRLF-to-LF-test-3-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T22:12:22.277000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "CRLF-to-LF-test-2",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T21:53:50.455000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "CRLF-to-LF-1-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T21:39:33.356000+00:00",
      "VocabularyState": "READY"
    }
  ]
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVocabularies](#)」の「」を参照してください。

list-vocabulary-filters

次のコード例は、list-vocabulary-filters を使用する方法を示しています。

AWS CLI

語彙フィルターを一覧表示するには

次のlist-vocabulary-filters例では、AWS アカウントとリージョンに関連付けられた語彙フィルターを一覧表示します。

```
aws transcribe list-vocabulary-filters
```

出力:

```
{
  "NextToken": "NextToken": [
    {
      "VocabularyFilterName": "testFilter",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00"
    },
    {
      "VocabularyFilterName": "testFilter2",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-05-21T23:29:35.174000+00:00"
    },
    {
      "VocabularyFilterName": "filter2",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-05-08T20:18:26.426000+00:00"
    },
    {
      "VocabularyFilterName": "filter-review",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-06-03T18:52:30.448000+00:00"
    },
    {
      "VocabularyFilterName": "crlf-filt",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-05-22T19:42:42.737000+00:00"
    }
  ]
}
```

```
}
```

詳細については、[「Amazon Transcribe デベロッパーガイド」の「不要な単語のフィルタリング」](#)を参照してください。Amazon Transcribe

- API 詳細については、AWS CLI 「コマンドリファレンス[ListVocabularyFilters](#)」の「」を参照してください。

start-medical-transcription-job

次のコード例は、start-medical-transcription-job を使用する方法を示しています。

AWS CLI

例 1: オーディオファイルとして保存されている医療ディクテーションを文字起こしするには

次の start-medical-transcription-job の例は、オーディオファイルの文字起こしを行います。トランスクリプション出力の場所を OutputBucketName パラメータで指定します。

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://myfile.json
```

myfile.json の内容:

```
{  
  "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",  
  "LanguageCode": "language-code",  
  "Specialty": "PRIMARYCARE",  
  "Type": "DICTATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  }  
}
```

出力:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",
```

```
"LanguageCode": "language-code",
"Media": {
  "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
},
"StartTime": "2020-09-20T00:35:22.256000+00:00",
"CreationTime": "2020-09-20T00:35:22.218000+00:00",
"Specialty": "PRIMARYCARE",
"Type": "DICTATION"
}
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[バッチトランスクリプションの概要](#)」を参照してください。

例 2: オーディオファイルとして保存されている臨床医と患者の対話を文字起こしするには

次の `start-medical-transcription-job` 例では、臨床医と患者の対話を含むオーディオファイルの文字起こしを行います。OutputBucketName パラメータで文字起こし出力の場所を指定します。

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysecondfile.json
```

mysecondfile.json の内容:

```
{
  "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  }
}
```

出力:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "simple-conversation-medical-transcription-job",
```

```

    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:19:49.965000+00:00",
    "CreationTime": "2020-09-20T23:19:49.941000+00:00",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}

```

詳細については、「Amazon Transcribe 開発者ガイド」の「[バッチトランスクリプションの概要](#)」を参照してください。

例 3: 臨床医と患者の対話のマルチチャンネルオーディオファイルを書き起こすには

次の `start-medical-transcription-job` 例では、オーディオファイルの各チャンネルの音声の文字起こしを行い、チャンネル別の文字起こし結果を組み合わせ、単一の文字起こし出力にまとめます。文字起こしの出力の場所を `OutputBucketName` パラメータで指定します。

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://mythirdfile.json

```

`mythirdfile.json` の内容:

```

{
  "MedicalTranscriptionJobName": "multichannel-conversation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  },
  "Settings": {
    "ChannelIdentification": true
  }
}

```

出力:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:46:44.081000+00:00",
    "CreationTime": "2020-09-20T23:46:44.053000+00:00",
    "Settings": {
      "ChannelIdentification": true
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[チャンネル識別](#)」を参照してください。

例 4: 臨床医と患者の対話のオーディオファイルを文字起こしして、文字起こし出力の話者を特定するには

次の `start-medical-transcription-job` の例は、オーディオファイルを書き起こしして、文字起こし出力の各話者の発話にラベルを付けます。文字起こしの出力の場所を `OutputBucketName` パラメータで指定します。

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfourthfile.json
```

`myfourthfile.json` の内容:

```
{
  "MedicalTranscriptionJobName": "speaker-id-conversation-medical-transcription-
job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
```

```
"Media": {
  "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
},
"Settings":{
  "ShowSpeakerLabels": true,
  "MaxSpeakerLabels": 2
}
}
```

出力:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T18:43:37.265000+00:00",
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",
    "Settings": {
      "ShowSpeakerLabels": true,
      "MaxSpeakerLabels": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[話者の識別](#)」を参照してください。

例 5: オーディオファイルとして保存されている医療会話を、最大 2 つの代替文字起こし結果に文字起こしするには

次の start-medical-transcription-job の例は、単一のオーディオファイルから最大 2 つの代替文字起こし結果を作成します。文字起こし結果ごとに信頼度レベルが関連付けられます。デフォルトでは、Amazon Transcribe は、信頼度レベルが最も高い文字起こし結果を返します。Amazon Transcribe で他の信頼度レベルがより低いトランスクリプションを返すようにも指定できます。文字起こしの出力の場所を OutputBucketName パラメータで指定します。

```
aws transcribe start-medical-transcription-job \  
--cli-input-json file://myfifthfile.json
```

myfifthfile.json の内容:

```
{  
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-  
job",  
  "LanguageCode": "language-code",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  },  
  "Settings": {  
    "ShowAlternatives": true,  
    "MaxAlternatives": 2  
  }  
}
```

出力:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-  
transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
    "StartTime": "2020-09-21T19:09:18.199000+00:00",  
    "CreationTime": "2020-09-21T19:09:18.171000+00:00",  
    "Settings": {  
      "ShowAlternatives": true,  
      "MaxAlternatives": 2  
    },  
    "Specialty": "PRIMARYCARE",  
    "Type": "CONVERSATION"  
  }  
}
```


詳細については、「Amazon Transcribe デベロッパーガイド」の「[代替文字起こし](#)」を参照してください。

例 6: 医療ディクテーションのオーディオファイルを、最大 2 つの代替文字起こし結果に文字起こしするには

次の `start-medical-transcription-job` の例は、オーディオファイルを文字起こしして、語彙フィルターを使用して不要な単語をマスクします。OutputBucketName パラメータで文字起こし出力の場所を指定します。

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://mysixthfile.json
```

mysixthfile.json の内容:

```
{  
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-job",  
  "LanguageCode": "language-code",  
  "Specialty": "PRIMARYCARE",  
  "Type": "DICTATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  },  
  "Settings": {  
    "ShowAlternatives": true,  
    "MaxAlternatives": 2  
  }  
}
```

出力:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "alternatives-dictation-medical-transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
  },  
}
```

```
"StartTime": "2020-09-21T21:01:14.592000+00:00",
"CreationTime": "2020-09-21T21:01:14.569000+00:00",
"Settings": {
  "ShowAlternatives": true,
  "MaxAlternatives": 2
},
"Specialty": "PRIMARYCARE",
"Type": "DICTATION"
}
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[代替文字起こし](#)」を参照してください。

例 7: カスタムボ語彙を使用して、医療ディクテーションのオーディオファイルをより正確に書き起こすには

次の `start-medical-transcription-job` の例は、オーディオファイルを文字起こしして、以前に作成した医療カスタム語彙を使用して文字起こし結果の精度を高めます。文字起こしの出力の場所を `OutputBucketName` パラメータで指定します。

```
aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json
```

`mysixthfile.json` の内容:

```
{
  "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  },
  "Settings": {
    "VocabularyName": "cli-medical-vocab-1"
  }
}
```

出力:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-
job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T21:17:27.045000+00:00",
    "CreationTime": "2020-09-21T21:17:27.016000+00:00",
    "Settings": {
      "VocabularyName": "cli-medical-vocab-1"
    },
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[医療カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartMedicalTranscriptionJob](#)」の「」を参照してください。

start-transcription-job

次のコード例は、start-transcription-job を使用する方法を示しています。

AWS CLI

例 1: オーディオファイルを文字起こしするには

次の start-transcription-job の例は、音声ファイルの文字起こしを行います。

```
aws transcribe start-transcription-job \
  --cli-input-json file://myfile.json
```

myfile.json の内容:

```
{
  "TranscriptionJobName": "cli-simple-transcription-job",
```

```
"LanguageCode": "the-language-of-your-transcription-job",
"Media": {
  "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
}
}
```

詳細については、[Amazon Transcribe デベロッパーガイド](#)の「[入門 \(AWS コマンドラインインターフェイス\)](#)」を参照してください。

例 2: マルチチャンネルのオーディオファイルを文字起こしするには

次の `start-transcription-job` の例は、マルチチャンネルのオーディオファイルの文字起こしを行います。

```
aws transcribe start-transcription-job \
  --cli-input-json file://mysecondfile.json
```

`mysecondfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-channelid-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
  },
  "Settings":{
    "ChannelIdentification":true
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-channelid-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    }
  }
}
```

```
    },
    "StartTime": "2020-09-17T16:07:56.817000+00:00",
    "CreationTime": "2020-09-17T16:07:56.784000+00:00",
    "Settings": {
      "ChannelIdentification": true
    }
  }
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[マルチチャンネル音声の書き起こし](#)」を参照してください。

例 3: オーディオファイルを文字起こしして、複数の異なる話者を識別するには

次の `start-transcription-job` 例では、オーディオファイルを書き起こし、文字起こし出力の話者を識別します。

```
aws transcribe start-transcription-job \
  --cli-input-json file://mythirdfile.json
```

`mythirdfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-speakerid-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"
  },
  "Settings":{
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-speakerid-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
```

```
"LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
  },
  "StartTime": "2020-09-17T16:22:59.696000+00:00",
  "CreationTime": "2020-09-17T16:22:59.676000+00:00",
  "Settings": {
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
  }
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[話者の識別](#)」を参照してください。

例 4: オーディオファイルを文字起こしして、文字起こし出力内の不要な単語をすべてマスクするには

次の `start-transcription-job` 例では、オーディオファイルを書き起こし、以前に作成した語彙フィルターを使用して不要な単語をマスクします。

```
aws transcribe start-transcription-job \
  --cli-input-json file://myfourthfile.json
```

`myfourthfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-filter-mask-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
  },
  "Settings":{
    "VocabularyFilterName": "your-vocabulary-filter",
    "VocabularyFilterMethod": "mask"
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-filter-mask-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyFilterName": "your-vocabulary-filter",
      "VocabularyFilterMethod": "mask"
    }
  }
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[トランスクリプションのフィルタリング](#)」を参照してください。

例 5: オーディオファイルを文字起こしし、文字起こし出力から不要な単語を削除するには

次の `start-transcription-job` 例では、オーディオファイルを書き起こし、以前に作成した語彙フィルターを使用して不要な単語をマスクします。

```
aws transcribe start-transcription-job \
  --cli-input-json file://myfifthfile.json
```

`myfifthfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-filter-remove-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"
  },
  "Settings": {
    "VocabularyFilterName": "your-vocabulary-filter",
    "VocabularyFilterMethod": "remove"
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-filter-remove-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyFilterName": "your-vocabulary-filter",
      "VocabularyFilterMethod": "remove"
    }
  }
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[トランスクリプションのフィルタリング](#)」を参照してください。

例 6: カスタム語彙を使用して、オーディオファイルをより正確に文字起こしするには

次の `start-transcription-job` 例では、オーディオファイルを書き起こし、以前に作成した語彙フィルターを使用して不要な単語をマスクします。

```
aws transcribe start-transcription-job \
  --cli-input-json file://mysixthfile.json
```

`mysixthfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-vocab-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"
  },
  "Settings": {
    "VocabularyName": "your-vocabulary"
  }
}
```



```
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-vocab-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyName": "your-vocabulary"
    }
  }
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[トランスクリプションのフィルタリング](#)」を参照してください。

例 7: オーディオファイルの言語を識別して文字起こしするには

次の `start-transcription-job` 例では、オーディオファイルを書き起こし、以前に作成した語彙フィルターを使用して不要な単語をマスクします。

```
aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json
```

`myseventhfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-identify-language-transcription-job",
  "IdentifyLanguage": true,
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-identify-language-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"
    },
    "StartTime": "2020-09-18T22:27:23.970000+00:00",
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",
    "IdentifyLanguage": true
  }
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[言語の特定](#)」を参照してください。

例 8: 個人を特定できる情報をマスクしてオーディオファイルを文字起こしするには

次の `start-transcription-job` の例は、オーディオファイルを文字起こしして、文字起こし出力内の個人を特定できる情報をマスクします。

```
aws transcribe start-transcription-job \
  --cli-input-json file://myeighthfile.json
```

`myeighthfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-redaction-job",
  "LanguageCode": "language-code",
  "Media": {
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
  },
  "ContentRedaction": {
    "RedactionOutput": "redacted",
    "RedactionType": "PII"
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-redaction-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "StartTime": "2020-09-25T23:49:13.195000+00:00",
    "CreationTime": "2020-09-25T23:49:13.176000+00:00",
    "ContentRedaction": {
      "RedactionType": "PII",
      "RedactionOutput": "redacted"
    }
  }
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[自動コンテンツリダクション](#)」を参照してください。

例 9: 個人を特定できる情報 (PII) と未編集のトランスクリプトを含むトランスクリプトを生成するには

次の `start-transcription-job` の例は、オーディオファイルの 2 つの文字起こしを生成します。1 つでは個人を特定できる情報をマスクし、別の 1 つではマスクしません。

```
aws transcribe start-transcription-job \
  --cli-input-json file://myninthfile.json
```

`myninthfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",
  "LanguageCode": "language-code",
  "Media": {
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
  },
  "ContentRedaction": {
    "RedactionOutput": "redacted_and_unredacted",
    "RedactionType": "PII"
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "StartTime": "2020-09-25T23:59:47.677000+00:00",
    "CreationTime": "2020-09-25T23:59:47.653000+00:00",
    "ContentRedaction": {
      "RedactionType": "PII",
      "RedactionOutput": "redacted_and_unredacted"
    }
  }
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[自動コンテンツリダクション](#)」を参照してください。

例 10: 以前に作成したカスタム言語モデルを使用してオーディオファイルを文字起こしするには次の `start-transcription-job` の例は、以前に作成したカスタム言語モデルを使用してオーディオファイルを文字起こしします。

```
aws transcribe start-transcription-job \
  --cli-input-json file://mytenthfile.json
```

`mytenthfile.json` の内容:

```
{
  "TranscriptionJobName": "cli-clm-2-job-1",
  "LanguageCode": "language-code",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"
  },
  "ModelSettings": {
    "LanguageModelName": "cli-clm-2"
  }
}
```

出力:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-clm-2-job-1",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"
    },
    "StartTime": "2020-09-28T17:56:01.835000+00:00",
    "CreationTime": "2020-09-28T17:56:01.801000+00:00",
    "ModelSettings": {
      "LanguageModelName": "cli-clm-2"
    }
  }
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[カスタム言語モデルを使用したドメイン固有のトランスクリプション精度の向上](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[StartTranscriptionJob](#)」の「」を参照してください。

update-medical-vocabulary

次の例は、update-medical-vocabulary を使用する方法を説明しています。

AWS CLI

新しい用語で医療カスタム語彙を更新するには。

次のupdate-medical-vocabulary例では、医療カスタム語彙で使用される用語を新しい語彙に置き換えます。前提条件: 医療カスタム語彙の用語を置き換えるには、新しい用語を持つファイルが必要です。

```
aws transcribe update-medical-vocabulary \
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/medical-custom-vocabulary.txt \
  --vocabulary-name medical-custom-vocabulary \
  --language-code language
```

出力:

```
{
  "VocabularyName": "medical-custom-vocabulary",
  "LanguageCode": "en-US",
  "VocabularyState": "PENDING"
}
```

詳細については、「Amazon Transcribe 開発者ガイド」の「[医療カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMedicalVocabulary](#)」の「」を参照してください。

update-vocabulary-filter

次のコード例は、update-vocabulary-filter を使用する方法を示しています。

AWS CLI

語彙フィルターの単語を置き換えるには

次のupdate-vocabulary-filter例では、語彙フィルターの単語を新しい単語に置き換えます。前提条件: 語彙フィルターを新しい単語で更新するには、それらの単語をテキストファイルとして保存する必要があります。

```
aws transcribe update-vocabulary-filter \
  --vocabulary-filter-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/your-text-file-to-update-your-vocabulary-filter.txt \
  --vocabulary-filter-name vocabulary-filter-name
```

出力:

```
{
  "VocabularyFilterName": "vocabulary-filter-name",
  "LanguageCode": "language-code",
  "LastModifiedTime": "2020-09-23T18:40:35.139000+00:00"
}
```

詳細については、「[Amazon Transcribe デベロッパーガイド](#)」の「[不要な単語のフィルタリング](#)」を参照してください。 Amazon Transcribe

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVocabularyFilter](#)」の「」を参照してください。

update-vocabulary

次の例は、update-vocabulary を使用する方法を説明しています。

AWS CLI

カスタム語彙を新しい用語で更新するには

次の update-vocabulary の例は、カスタム語彙の作成に使用した用語を、指定した新しい用語で上書きします。前提条件: カスタム語彙の用語を置き換えるには、新しい用語を含むファイルが必要です。

```
aws transcribe update-vocabulary \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/custom-  
vocabulary.txt \  
  --vocabulary-name custom-vocabulary \  
  --language-code language-code
```

出力:

```
{  
  "VocabularyName": "custom-vocabulary",  
  "LanguageCode": "language",  
  "VocabularyState": "PENDING"  
}
```

詳細については、「Amazon Transcribe デベロッパーガイド」の「[カスタムボキャブラリー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateVocabulary](#)」の「」を参照してください。

を使用した Amazon Translate の例 AWS CLI

次のコード例は、Amazon Translate AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

import-terminology

次の例は、import-terminology を使用する方法を説明しています。

AWS CLI

ファイルからカスタム用語をインポートするには

次のimport-terminology例では、test-terminology.csv ファイルMyTestTerminologyから という用語を作成します。

```
aws translate import-terminology \  
  --name MyTestTerminology \  
  --description "Creating a test terminology in AWS Translate" \  
  --merge-strategy OVERWRITE \  
  --data-file fileb://test-terminology.csv \  
  --terminology-data Format=CSV
```

test-terminology.csv の内容:

```
en,fr,es,zh Hello world!,Bonjour tout le monde!,Hola Mundo!,???  
Amazon、Amazon、Amazon、Amazon
```

出力:

```
{  
  "TerminologyProperties": {
```



```
    "SourceLanguageCode": "en",
    "Name": "MyTestTerminology",
    "TargetLanguageCodes": [
      "fr",
      "es",
      "zh"
    ],
    "SizeBytes": 97,
    "LastUpdatedAt": 1571089500.851,
    "CreatedAt": 1571089500.851,
    "TermCount": 6,
    "Arn": "arn:aws:translate:us-west-2:123456789012:terminology/
MyTestTerminology/LATEST",
    "Description": "Creating a test terminology in AWS Translate"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ImportTerminology](#)」の「」を参照してください。

Trusted Advisor を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています Trusted Advisor。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

get-organization-recommendation

次のコード例は、get-organization-recommendation を使用する方法を示しています。

AWS CLI

組織のレコメンデーションを取得するには

次の`get-organization-recommendation`例では、識別子によって組織のレコメンデーションを取得します。

```
aws trustedadvisor get-organization-recommendation \  
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-  
recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

出力:

```
{  
  "organizationRecommendation": {  
    "arn": "arn:aws:trustedadvisor::organization-recommendation/9534ec9b-  
bf3a-44e8-8213-2ed68b39d9d5",  
    "name": "Lambda Runtime Deprecation Warning",  
    "description": "One or more lambdas are using a deprecated runtime",  
    "awsServices": [  
      "lambda"  
    ],  
    "checkArn": "arn:aws:trustedadvisor::check/L4dfs2Q4C5",  
    "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",  
    "lifecycleStage": "resolved",  
    "pillars": [  
      "security"  
    ],  
    "resourcesAggregates": {  
      "errorCount": 0,  
      "okCount": 0,  
      "warningCount": 0  
    },  
    "source": "ta_check",  
    "status": "warning",  
    "type": "priority"  
  }  
}
```

詳細については、[Trusted Advisorユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[GetOrganizationRecommendation](#)」の「」を参照してください。

get-recommendation

次のコード例は、get-recommendation を使用する方法を示しています。

AWS CLI

レコメンデーションを取得するには

次のget-recommendation例では、識別子によってレコメンデーションを取得します。

```
aws trustedadvisor get-recommendation \  
  --recommendation-  
  identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-  
  bbb7-491a-833b-5773e9589578
```

出力:

```
{  
  "recommendation": {  
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-  
    bbb7-491a-833b-5773e9589578",  
    "name": "MFA Recommendation",  
    "description": "Enable multi-factor authentication",  
    "awsServices": [  
      "iam"  
    ],  
    "checkArn": "arn:aws:trustedadvisor:::check/7DAFEemoDos",  
    "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",  
    "lastUpdatedAt": "2023-11-01T15:57:58.673Z",  
    "pillarSpecificAggregates": {  
      "costOptimizing": {  
        "estimatedMonthlySavings": 0.0,  
        "estimatedPercentMonthlySavings": 0.0  
      }  
    },  
    "pillars": [  
      "security"  
    ],  
    "resourcesAggregates": {  
      "errorCount": 1,  

```

```
        "okCount": 0,  
        "warningCount": 0  
    },  
    "source": "ta_check",  
    "status": "error",  
    "type": "standard"  
  }  
}
```

詳細については、[Trusted Advisor ユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRecommendation](#)」の「」を参照してください。

list-checks

次のコード例は、list-checks を使用する方法を示しています。

AWS CLI

Trusted Advisor チェックを一覧表示するには

次のlist-checks例では、すべての Trusted Advisor チェックを一覧表示します。

```
aws trustedadvisor list-checks
```

出力:

```
{  
  "checkSummaries": [  
    {  
      "arn": "arn:aws:trustedadvisor:::check/1iG5NDGVre",  
      "awsServices": [  
        "EC2"  
      ],  
      "description": "Checks security groups for rules that allow unrestricted  
access to a resource. Unrestricted access increases opportunities for malicious  
activity (hacking, denial-of-service attacks, loss of data)",  
      "id": "1iG5NDGVre",  
      "metadata": {  
        "0": "Region",  
        "1": "Security Group Name",  
      }  
    }  
  ]  
}
```

```

        "2": "Security Group ID",
        "3": "Protocol",
        "4": "Port",
        "5": "Status",
        "6": "IP Range"
    },
    "name": "Security Groups - Unrestricted Access",
    "pillars": [
        "security"
    ],
    "source": "ta_check"
},
{
    "arn": "arn:aws:trustedadvisor:::check/1qazXsw23e",
    "awsServices": [
        "RDS"
    ],
    "description": "Checks your usage of RDS and provides recommendations
on purchase of Reserved Instances to help reduce costs incurred from using RDS
On-Demand. AWS generates these recommendations by analyzing your On-Demand usage
for the past 30 days. We then simulate every combination of reservations in the
generated category of usage in order to identify the best number of each type
of Reserved Instance to purchase to maximize your savings. This check covers
recommendations based on partial upfront payment option with 1-year or 3-year
commitment. This check is not available to accounts linked in Consolidated Billing.
Recommendations are only available for the Paying Account.",
    "id": "1qazXsw23e",
    "metadata": {
        "0": "Region",
        "1": "Family",
        "2": "Instance Type",
        "3": "License Model",
        "4": "Database Edition",
        "5": "Database Engine",
        "6": "Deployment Option",
        "7": "Recommended number of Reserved Instances to purchase",
        "8": "Expected Average Reserved Instance Utilization",
        "9": "Estimated Savings with Recommendation (monthly)",
        "10": "Upfront Cost of Reserved Instances",
        "11": "Estimated cost of Reserved Instances (monthly)",
        "12": "Estimated On-Demand Cost Post Recommended Reserved Instance
Purchase (monthly)",
        "13": "Estimated Break Even (months)",
        "14": "Lookback Period (days)",

```

```

        "15": "Term (years)"
    },
    "name": "Amazon Relational Database Service (RDS) Reserved Instance
Optimization",
    "pillars": [
        "cost_optimizing"
    ],
    "source": "ta_check"
},
{
    "arn": "arn:aws:trustedadvisor:::check/1qw23er45t",
    "awsServices": [
        "Redshift"
    ],
    "description": "Checks your usage of Redshift and provides
recommendations on purchase of Reserved Nodes to help reduce costs incurred from
using Redshift On-Demand. AWS generates these recommendations by analyzing your
On-Demand usage for the past 30 days. We then simulate every combination of
reservations in the generated category of usage in order to identify the best
number of each type of Reserved Nodes to purchase to maximize your savings. This
check covers recommendations based on partial upfront payment option with 1-year or
3-year commitment. This check is not available to accounts linked in Consolidated
Billing. Recommendations are only available for the Paying Account.",
    "id": "1qw23er45t",
    "metadata": {
        "0": "Region",
        "1": "Family",
        "2": "Node Type",
        "3": "Recommended number of Reserved Nodes to purchase",
        "4": "Expected Average Reserved Node Utilization",
        "5": "Estimated Savings with Recommendation (monthly)",
        "6": "Upfront Cost of Reserved Nodes",
        "7": "Estimated cost of Reserved Nodes (monthly)",
        "8": "Estimated On-Demand Cost Post Recommended Reserved Nodes
Purchase (monthly)",
        "9": "Estimated Break Even (months)",
        "10": "Lookback Period (days)",
        "11": "Term (years)",
    },
    "name": "Amazon Redshift Reserved Node Optimization",
    "pillars": [
        "cost_optimizing"
    ],
    "source": "ta_check"
}

```

```

    },
  ],
  "nextToken": "REDACTED"
}

```

詳細については、[Trusted Advisor ユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[ListChecks](#)」の「」を参照してください。

list-organization-recommendation-accounts

次の例は、list-organization-recommendation-accounts を使用する方法を説明しています。

AWS CLI

組織のレコメンデーションアカウントを一覧表示するには

次のlist-organization-recommendation-accounts例では、組織のレコメンデーションのすべてのアカウントレコメンデーションの概要を識別子別に一覧表示します。

```

aws trustedadvisor list-organization-recommendation-accounts \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5

```

出力:

```

{
  "accountRecommendationLifecycleSummaries": [{
    "accountId": "000000000000",
    "accountRecommendationArn":
"arn:aws:trustedadvisor::000000000000:recommendation/9534ec9b-
bf3a-44e8-8213-2ed68b39d9d5",
    "lifecycleStage": "resolved",
    "updateReason": "Resolved issue",
    "updateReasonCode": "valid_business_case",
    "lastUpdatedAt": "2023-01-17T18:25:44.552Z"
  }],
  "nextToken": "REDACTED"
}

```

詳細については、[Trusted Advisor ユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「[コマンドリファレンスListOrganizationRecommendationAccounts](#)」の「」を参照してください。

list-organization-recommendation-resources

次の例は、list-organization-recommendation-resources を使用する方法を説明しています。

AWS CLI

組織のレコメンデーションリソースを一覧表示するには

次のlist-organization-recommendation-resources例では、組織のレコメンデーションのすべてのリソースを識別子別に一覧表示します。

```
aws trustedadvisor list-organization-recommendation-resources \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-recommendation/5a694939-2e54-45a2-ae72-730598fa89d0
```

出力:

```
{
  "organizationRecommendationResourceSummaries": [
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation-resource/5a694939-2e54-45a2-ae72-730598fa89d0/bb38affc0ce0681d9a6cd13f30238ba03a8f63dfe7a379dc403c619119d86af",
      "awsResourceId": "database-1-instance-1",
      "id": "bb38affc0ce0681d9a6cd13f302383ba03a8f63dfe7a379dc403c619119d86af",
      "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
      "metadata": {
        "0": "14",
        "1": "208.79999999999998",
        "2": "database-1-instance-1",
        "3": "db.r5.large",
        "4": "false",
        "5": "us-west-2",
        "6": "arn:aws:rds:us-west-2:000000000000:db:database-1-instance-1",
        "7": "1"
      }
    }
  ]
}
```



```
    },
    "recommendationArn": "arn:aws:trustedadvisor::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
    "regionCode": "us-west-2",
    "status": "warning"
  },
  {
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-
ae72-730598fa89d0/51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
    "awsResourceId": "database-1",
    "id":
"51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
    "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
    "metadata": {
      "0": "14",
      "1": "31.679999999999996",
      "2": "database-1",
      "3": "db.t3.small",
      "4": "false",
      "5": "us-west-2",
      "6": "arn:aws:rds:us-west-2:000000000000:db:database-1",
      "7": "20"
    },
    },
    "recommendationArn": "arn:aws:trustedadvisor::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
    "regionCode": "us-west-2",
    "status": "warning"
  },
  {
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-ae72-730598fa89d0/
f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
    "awsResourceId": "database-2-instance-1-us-west-2a",
    "id":
"f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
    "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
    "metadata": {
      "0": "14",
      "1": "187.200000000000002",
      "2": "database-2-instance-1-us-west-2a",
      "3": "db.r6g.large",
      "4": "true",
      "5": "us-west-2",
```

```
        "6": "arn:aws:rds:us-west-2:000000000000:db:database-2-instance-1-  
us-west-2a",  
        "7": "1"  
    },  
    "recommendationArn": "arn:aws:trustedadvisor:::organization-  
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",  
    "regionCode": "us-west-2",  
    "status": "warning"  
  },  
],  
"nextToken": "REDACTED"  
}
```

詳細については、[Trusted Advisor ユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOrganizationRecommendationResources](#)」の「」を参照してください。

list-organization-recommendations

次の例は、list-organization-recommendations を使用する方法を説明しています。

AWS CLI

例 1: 組織の推奨事項を一覧表示するには

次のlist-organization-recommendations例では、すべての組織のレコメンデーションを一覧表示し、フィルターは含まれません。

```
aws trustedadvisor list-organization-recommendations
```

出力:

```
{  
  "organizationRecommendationSummaries": [  
    {  
      "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-  
bf3a-44e8-8213-2ed68b39d9d5",  
      "name": "Lambda Runtime Deprecation Warning",  
      "awsServices": [  
        "lambda"  
      ],  
    },  
  ],  
}
```

```
    "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
    "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",
    "lifecycleStage": "resolved",
    "pillars": [
      "security"
    ],
    "resourcesAggregates": {
      "errorCount": 0,
      "okCount": 0,
      "warningCount": 0
    },
    "source": "ta_check",
    "status": "warning",
    "type": "priority"
  },
  {
    "arn": "arn:aws:trustedadvisor:::organization-
recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
    "name": "Lambda Runtime Deprecation Warning",
    "awsServices": [
      "lambda"
    ],
    "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",
    "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",
    "lifecycleStage": "resolved",
    "pillars": [
      "security"
    ],
    "resourcesAggregates": {
      "errorCount": 0,
      "okCount": 0,
      "warningCount": 0
    },
    "source": "ta_check",
    "status": "warning",
    "type": "priority"
  },
],
"nextToken": "REDACTED"
}
```

詳細については、[Trusted Advisorユーザーガイド](#)の「Trusted Advisor の使用を開始するAPI」を参照してください。AWS Trusted Advisor

例 2: フィルターを使用して組織のレコメンデーションを一覧表示するには

次のlist-organization-recommendations例では、「セキュリティ」の柱の一部である 1 つの組織レコメンデーションの最大数をフィルタリングして返します。

```
aws trustedadvisor list-organization-recommendations \  
  --pillar security \  
  --max-items 100
```

出力:

```
{  
  "organizationRecommendationSummaries": [{  
    "arn": "arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-  
bf3a-44e8-8213-2ed68b39d9d5",  
    "name": "Lambda Runtime Deprecation Warning",  
    "awsServices": [  
      "lambda"  
    ],  
    "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",  
    "id": "9534ec9b-bf3a-44e8-8213-2ed68b39d9d5",  
    "lifecycleStage": "resolved",  
    "pillars": [  
      "security"  
    ],  
    "resourcesAggregates": {  
      "errorCount": 0,  
      "okCount": 0,  
      "warningCount": 0  
    },  
    "source": "ta_check",  
    "status": "warning",  
    "type": "priority"  
  }],  
  "nextToken": "REDACTED"  
}
```

詳細については、[Trusted Advisorユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

例 3: ページ分割トークンを使用して組織のレコメンデーションを一覧表示するには

次のlist-organization-recommendations例では、前のリクエストから返されたnextToken「」を使用して、組織のレコメンデーションの次のページを取得します。

```
aws trustedadvisor list-organization-recommendations \  
  --pillar security \  
  --max-items 100 \  
  --starting-token <next-token>
```

出力:

```
{  
  "organizationRecommendationSummaries": [{  
    "arn": "arn:aws:trustedadvisor:::organization-  
recommendation/4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",  
    "name": "Lambda Runtime Deprecation Warning",  
    "awsServices": [  
      "lambda"  
    ],  
    "checkArn": "arn:aws:trustedadvisor:::check/L4dfs2Q4C5",  
    "id": "4ecff4d4-1bc1-4c99-a5b8-0fff9ee500d6",  
    "lifecycleStage": "resolved",  
    "pillars": [  
      "security"  
    ],  
    "resourcesAggregates": {  
      "errorCount": 0,  
      "okCount": 0,  
      "warningCount": 0  
    },  
    "source": "ta_check",  
    "status": "warning",  
    "type": "priority"  
  }]  
}
```

詳細については、[Trusted Advisorユーザーガイド](#)の「Trusted Advisor の使用を開始するAPI」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOrganizationRecommendations](#)」の「」を参照してください。

list-recommendation-resources

次の例は、list-recommendation-resources を使用する方法を説明しています。

AWS CLI

レコメンデーションリソースを一覧表示するには

次のlist-recommendation-resources例では、レコメンデーションのすべてのリソースを識別子別に一覧表示します。

```
aws trustedadvisor list-recommendation-resources \
  --recommendation-
  identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
  bbb7-491a-833b-5773e9589578
```

出力:

```
{
  "recommendationResourceSummaries": [
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
      resource/55fa4d2e-
      bbb7-491a-833b-5773e9589578/18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010",
      "id":
      "18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010",
      "awsResourceId": "webcms-dev-01",
      "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
      "metadata": {
        "0": "14",
        "1": "123.120000000000002",
        "2": "webcms-dev-01",
        "3": "db.m6i.large",
        "4": "false",
        "5": "us-east-1",
        "6": "arn:aws:rds:us-east-1:000000000000:db:webcms-dev-01",
        "7": "20"
      },
      "recommendationArn":
      "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
      bbb7-491a-833b-5773e9589578",
      "regionCode": "us-east-1",
      "status": "warning"
    }
  ]
}
```

```

    },
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-bbb7-491a-833b-5773e9589578/
e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
      "id":
"e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
      "awsResourceId": "aws-dev-db-stack-instance-1",
      "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
      "metadata": {
        "0": "14",
        "1": "29.52",
        "2": "aws-dev-db-stack-instance-1",
        "3": "db.t2.small",
        "4": "false",
        "5": "us-east-1",
        "6": "arn:aws:rds:us-east-1:000000000000:db:aws-dev-db-stack-
instance-1",
        "7": "1"
      },
      "recommendationArn":
"arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
      "regionCode": "us-east-1",
      "status": "warning"
    },
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459
",
      "id":
"31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459",
      "awsResourceId": "aws-awesome-apps-stack-db",
      "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
      "metadata": {
        "0": "14",
        "1": "114.48000000000002",
        "2": "aws-awesome-apps-stack-db",
        "3": "db.m6g.large",
        "4": "false",
        "5": "us-east-1",
        "6": "arn:aws:rds:us-east-1:000000000000:db:aws-awesome-apps-stack-
db",
        "7": "100"
      }
    }
  ]
}

```

```
    },
    "recommendationArn":
"arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
    "regionCode": "us-east-1",
    "status": "warning"
  }
],
"nextToken": "REDACTED"
}
```

詳細については、[Trusted Advisorユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRecommendationResources](#)」の「」を参照してください。

list-recommendations

次の例は、list-recommendations を使用する方法を説明しています。

AWS CLI

例 1: レコメンデーションを一覧表示するには

次のlist-recommendations例では、すべてのレコメンデーションを一覧表示し、フィルターは含まれません。

```
aws trustedadvisor list-recommendations
```

出力:

```
{
  "recommendationSummaries": [
    {
      "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
      "name": "MFA Recommendation",
      "awsServices": [
        "iam"
      ],
      "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
      "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
```



```
"lastUpdatedAt": "2023-11-01T15:57:58.673Z",
"pillarSpecificAggregates": {
  "costOptimizing": {
    "estimatedMonthlySavings": 0.0,
    "estimatedPercentMonthlySavings": 0.0
  }
},
"pillars": [
  "security"
],
"resourcesAggregates": {
  "errorCount": 1,
  "okCount": 0,
  "warningCount": 0
},
"source": "ta_check",
"status": "error",
"type": "standard"
},
{
  "arn":
  "arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
c7650955d9cd",
  "name": "RDS clusters quota warning",
  "awsServices": [
    "rds"
  ],
  "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
  "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
  "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
  "pillarSpecificAggregates": {
    "costOptimizing": {
      "estimatedMonthlySavings": 0.0,
      "estimatedPercentMonthlySavings": 0.0
    }
  },
  "pillars": [
    "service_limits"
  ],
  "resourcesAggregates": {
    "errorCount": 0,
    "okCount": 3,
    "warningCount": 6
  },
}
```

```
        "source": "ta_check",
        "status": "warning",
        "type": "standard"
    }
],
"nextToken": "REDACTED"
}
```

詳細については、[Trusted Advisorユーザーガイド](#)の「Trusted Advisor の使用を開始するAPI」を参照してください。AWS Trusted Advisor

例 2: フィルターを使用してレコメンデーションを一覧表示するには

次のlist-recommendations例では、レコメンデーションを一覧表示し、フィルターを含めません。

```
aws trustedadvisor list-recommendations \
  --aws-service iam \
  --max-items 100
```

出力:

```
{
  "recommendationSummaries": [{
    "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-bbb7-491a-833b-5773e9589578",
    "name": "MFA Recommendation",
    "awsServices": [
      "iam"
    ],
    "checkArn": "arn:aws:trustedadvisor:::check/7DAFEemoDos",
    "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
    "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
    "pillarSpecificAggregates": {
      "costOptimizing": {
        "estimatedMonthlySavings": 0.0,
        "estimatedPercentMonthlySavings": 0.0
      }
    },
    "pillars": [
      "security"
    ],
  ],
}
```

```

    "resourcesAggregates": {
      "errorCount": 1,
      "okCount": 0,
      "warningCount": 0
    },
    "source": "ta_check",
    "status": "error",
    "type": "standard"
  }],
  "nextToken": "REDACTED"
}

```

詳細については、[Trusted Advisorユーザーガイド](#)の「Trusted Advisor の使用を開始するAPI」を参照してください。AWS Trusted Advisor

例 3: ページ分割トークンを使用してレコメンデーションを一覧表示するには

次のlist-recommendations例では、前のリクエストから返されたnextToken「」を使用して、フィルタリングされたレコメンデーションの次のページを取得します。

```

aws trustedadvisor list-recommendations \
  --aws-service rds \
  --max-items 100 \
  --starting-token <next-token>

```

出力:

```

{
  "recommendationSummaries": [{
    "arn":
    "arn:aws:trustedadvisor::000000000000:recommendation/8b602b6f-452d-4cb2-8a9e-
    c7650955d9cd",
    "name": "RDS clusters quota warning",
    "awsServices": [
      "rds"
    ],
    "checkArn": "arn:aws:trustedadvisor:::check/gjqMBn6pjz",
    "id": "8b602b6f-452d-4cb2-8a9e-c7650955d9cd",
    "lastUpdatedAt": "2023-11-01T15:58:17.397Z",
    "pillarSpecificAggregates": {
      "costOptimizing": {
        "estimatedMonthlySavings": 0.0,

```

```
        "estimatedPercentMonthlySavings": 0.0
      }
    },
    "pillars": [
      "service_limits"
    ],
    "resourcesAggregates": {
      "errorCount": 0,
      "okCount": 3,
      "warningCount": 6
    },
    "source": "ta_check",
    "status": "warning",
    "type": "standard"
  ]
}
```

詳細については、[Trusted Advisor ユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRecommendations](#)」の「」を参照してください。

update-organization-recommendation-lifecycle

次のコード例は、update-organization-recommendation-lifecycle を使用する方法を示しています。

AWS CLI

組織のレコメンデーションライフサイクルを更新するには

次のupdate-organization-recommendation-lifecycle例では、組織のレコメンデーションのライフサイクルを識別子で更新します。

```
aws trustedadvisor update-organization-recommendation-lifecycle \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-recommendation/96b5e5ca-7930-444c-90c6-06d386128100 \
  --lifecycle-stage dismissed \
  --update-reason-code not_applicable
```

このコマンドでは何も出力されません。

詳細については、[Trusted Advisor ユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateOrganizationRecommendationLifecycle](#)」の「」を参照してください。

update-recommendation-lifecycle

次のコード例は、update-recommendation-lifecycle を使用する方法を示しています。

AWS CLI

レコメンデーションライフサイクルを更新するには

次のupdate-recommendation-lifecycle例では、識別子によってレコメンデーションのライフサイクルを更新します。

```
aws trustedadvisor update-recommendation-lifecycle \  
  --recommendation-  
  identifier arn:aws:trustedadvisor::000000000000:recommendation/861c9c6e-  
  f169-405a-8b59-537a8cacc7a \  
  --lifecycle-stage resolved \  
  --update-reason-code valid_business_case
```

このコマンドでは何も出力されません。

詳細については、[Trusted Advisorユーザーガイド](#)の「[Trusted Advisor の使用を開始するAPI](#)」を参照してください。AWS Trusted Advisor

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRecommendationLifecycle](#)」の「」を参照してください。

を使用した検証済みアクセス許可の例 AWS CLI

次のコード例は、Verified Permissions AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-identity-source

次のコード例は、create-identity-source を使用する方法を示しています。

AWS CLI

ID ソースを作成するには

次のcreate-identity-source例では、指定された Amazon Cognito ユーザープールに保存されている ID を参照できる ID ソースを作成します。これらの ID は、Verified Permissions でタイプのエンティティとして使用できますUser。

```
aws verifiedpermissions create-identity-source \
  --configuration file://config.txt \
  --principal-entity-type "User" \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

config.txt の内容:

```
{
  "cognitoUserPoolConfiguration": {
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
    "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
  }
}
```

出力:

```
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEEabcdefg111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
```

```
"policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

ID ソースの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「ID プロバイダーでの Amazon Verified Permissions の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateIdentitySource](#)」の「」を参照してください。

create-policy-store

次の例は、create-policy-store を使用する方法を説明しています。

AWS CLI

ポリシーストアを作成するには

次のcreate-policy-store例では、現在の AWS リージョンにポリシーストアを作成します。

```
aws verifiedpermissions create-policy-store \  
--validation-settings "mode=STRICT"
```

出力:

```
{  
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/  
PSEXAMPLEEabcdefg111111",  
  "createdDate": "2023-05-16T17:41:29.103459+00:00",  
  "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111"  
}
```

ポリシーストアの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーストア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePolicyStore](#)」の「」を参照してください。

create-policy-template

次の例は、create-policy-template を使用する方法を説明しています。

AWS CLI

例 1: ポリシーテンプレートを作成するには

次のcreate-policy-template例では、プリンシパルのプレースホルダーを含むステートメントを含むポリシーテンプレートを作成します。

```
aws verifiedpermissions create-policy-template \  
  --definition file://template1.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

template1.txt ファイルの内容。

```
permit(  
  principal in ?principal,  
  action == Action::"view",  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

出力:

```
{  
  "createdDate": "2023-06-12T20:47:42.804511+00:00",  
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "policyTemplateId": "PTEXAMPLEEabcdefg111111"  
}
```

ポリシーテンプレートの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーテンプレート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePolicyTemplate](#)」の「」を参照してください。

create-policy

次のコード例は、create-policy を使用する方法を示しています。

AWS CLI

例 1: 静的ポリシーを作成するには

次のcreate-policy例では、プリンシパルとリソースの両方を指定するポリシースコープを持つ静的ポリシーを作成します。

```
aws verifiedpermissions create-policy \  
  --definition file://definition1.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

definition1.txt ファイルの内容。

```
{  
  "static": {  
    "description": "Grant everyone of janeFriends UserGroup access to the  
vacationFolder Album",  
    "statement": "permit(principal in UserGroup::\\"janeFriends\\", action,  
resource in Album::\\"vacationFolder\\" );"  
  }  
}
```

出力:

```
{  
  "createdDate": "2023-06-12T20:33:37.382907+00:00",  
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",  
  "policyId": "SPEXAMPLEEabcdefg111111",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "policyType": "STATIC",  
  "principal": {  
    "entityId": "janeFriends",  
    "entityType": "UserGroup"  
  },  
  "resource": {  
    "entityId": "vacationFolder",  
    "entityType": "Album"  
  }  
}
```

例 2: リソースへのアクセスを全員に許可する静的ポリシーを作成するには

次のcreate-policy例では、リソースのみを指定するポリシースコープを持つ静的ポリシーを作成します。

```
aws verifiedpermissions create-policy \  
  --definition file://definition1.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

```
--definition file://definition2.txt \  
--policy-store-id PSEXAMPLEEabcdefg111111
```

definition2.txt ファイルの内容。

```
{  
  "static": {  
    "description": "Grant everyone access to the publicFolder Album",  
    "statement": "permit(principal, action, resource in Album:\""publicFolder  
  \");"  
  }  
}
```

出力:

```
{  
  "createdDate": "2023-06-12T20:39:44.975897+00:00",  
  "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",  
  "policyId": "PbfR73F8oh5MMfr9uRtFDB",  
  "policyStoreId": "PSEXAMPLEEabcdefg222222",  
  "policyType": "STATIC",  
  "resource": {  
    "entityId": "publicFolder",  
    "entityType": "Album"  
  }  
}
```

例 3: 指定されたテンプレートに関連付けられているテンプレートにリンクされたポリシーを作成するには

次のcreate-policy例では、指定されたポリシーテンプレートを使用してテンプレートリンクポリシーを作成し、使用する指定されたプリンシパルを新しいテンプレートリンクポリシーに関連付けます。

```
aws verifiedpermissions create-policy \  
--definition file://definition.txt \  
--policy-store-id PSEXAMPLEEabcdefg111111
```

definition.txt の内容:

```
{
```

```
"templateLinked": {
  "policyTemplateId": "PTEXAMPLEEabcdefg111111",
  "principal": {
    "entityType": "User",
    "entityId": "alice"
  }
}
}
```

出力:

```
{
  "createdDate": "2023-06-12T20:49:51.490211+00:00",
  "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
  "policyId": "TPEXAMPLEEabcdefg111111",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "policyType": "TEMPLATE_LINKED",
  "principal": {
    "entityId": "alice",
    "entityType": "User"
  },
  "resource": {
    "entityId": "VacationPhoto94.jpg",
    "entityType": "Photo"
  }
}
```

ポリシーの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreatePolicy](#)」の「」を参照してください。

delete-identity-source

次の例は、delete-identity-source を使用する方法を説明しています。

AWS CLI

ID ソースを削除するには

次のdelete-identity-source例では、指定された ID を持つ ID ソースを削除します。

```
aws verifiedpermissions delete-identity-source \  
  --identity-source-id ISEXAMPLEabcdefg111111 \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

このコマンドでは何も出力されません。

ID ソースの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「ID プロバイダーでの Amazon Verified Permissions の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIdentitySource](#)」の「」を参照してください。

delete-policy-store

次のコード例は、delete-policy-store を使用する方法を示しています。

AWS CLI

ポリシーストアを削除するには

次のdelete-policy-store例では、指定された ID を持つポリシーストアを削除します。

```
aws verifiedpermissions delete-policy-store \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

このコマンドでは何も出力されません。

ポリシーストアの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーストア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicyStore](#)」の「」を参照してください。

delete-policy-template

次の例は、delete-policy-template を使用する方法を説明しています。

AWS CLI

ポリシーテンプレートを削除するには

次のdelete-policy-template例では、指定された ID を持つポリシーテンプレートを削除します。

```
aws verifiedpermissions delete-policy \  
  --policy-template-id PTEXAMPLEabcdefgh111111 \  
  --policy-store-id PSEXAMPLEabcdefgh111111
```

このコマンドでは何も出力されません。

ポリシーテンプレートの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーテンプレート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicyTemplate](#)」の「」を参照してください。

delete-policy

次の例は、delete-policy を使用する方法を説明しています。

AWS CLI

静的ポリシーまたはテンプレートにリンクされたポリシーを削除するには

次のdelete-policy例では、指定された ID を持つポリシーを削除します。

```
aws verifiedpermissions delete-policy \  
  --policy-id SPEXAMPLEabcdefgh111111 \  
  --policy-store-id PSEXAMPLEabcdefgh111111
```

このコマンドでは何も出力されません。

ポリシーの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeletePolicy](#)」の「」を参照してください。

get-identity-source

次のコード例は、get-identity-source を使用する方法を示しています。

AWS CLI

ID ソースの詳細を取得するには

次の`get-identity-source`例では、指定された ID を持つ ID ソースの詳細を表示します。

```
aws verifiedpermissions get-identity-source \  
  --identity-source ISEXAMPLEEabcdefg111111 \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

出力:

```
{  
  "createdDate": "2023-06-12T22:27:49.150035+00:00",  
  "details": {  
    "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],  
    "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_1a2b3c4d5",  
    "openIdIssuer": "COGNITO",  
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_1a2b3c4d5"  
  },  
  "identitySourceId": "ISEXAMPLEEabcdefg111111",  
  "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "principalEntityType": "User"  
}
```

ID ソースの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「ID プロバイダーでの Amazon Verified Permissions の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetIdentitySource](#)」の「」を参照してください。

get-policy-store

次の例は、`get-policy-store` を使用する方法を説明しています。

AWS CLI

ポリシーストアの詳細を取得するには

次のget-policy-store例では、指定された ID を持つポリシーストアの詳細を表示します。

```
aws verifiedpermissions get-policy-store \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

出力:

```
{  
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/  
PSEXAMPLEEabcdefg111111",  
  "createdDate": "2023-06-05T20:16:46.225598+00:00",  
  "lastUpdatedDate": "2023-06-08T20:40:23.173691+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "validationSettings": { "mode": "OFF" }  
}
```

ポリシーストアの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーストア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicyStore](#)」の「」を参照してください。

get-policy-template

次の例は、get-policy-template を使用方法を説明しています。

AWS CLI

ポリシーテンプレートの詳細を取得するには

次のget-policy-template例は、指定された ID のポリシーテンプレートの詳細を表示します。

```
aws verifiedpermissions get-policy-template \  
  --policy-template-id PTEXAMPLEEabcdefg111111 \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

出力:

```
{
```

```
"createdDate": "2023-06-12T20:47:42.804511+00:00",
"lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
"policyStoreId": "PSEXAMPLEEabcdefg111111",
"policyTemplateId": "PTEXAMPLEEabcdefg111111",
"statement": "permit(\n    principal in ?principal,\n    action == Action::\n    \"view\", \n    resource == Photo::\"VacationPhoto94.jpg\" )";
}
```

ポリシーテンプレートの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーテンプレート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicyTemplate](#)」の「」を参照してください。

get-policy

次の例は、get-policy を使用する方法を説明しています。

AWS CLI

ポリシーの詳細を取得するには

次のget-policy例では、指定された ID を持つポリシーの詳細を表示します。

```
aws verifiedpermissions get-policy \
  --policy-id PSEXAMPLEEabcdefg111111 \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

出力:

```
{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "definition": {
    "static": {
      "description": "Grant everyone of janeFriends UserGroup access to the vacationFolder Album",
      "statement": "permit(principal in UserGroup::\"janeFriends\", action, resource in Album::\"vacationFolder\" );"
    }
  },
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
  "policyId": "SPEXAMPLEEabcdefg111111",
```



```
"policyStoreId": "PSEXAMPLEabcdefg111111",
"policyType": "STATIC",
"principal": {
  "entityId": "janeFriends",
  "entityType": "UserGroup"
},
"resource": {
  "entityId": "vacationFolder",
  "entityType": "Album"
}
}
```

ポリシーの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetPolicy](#)」の「」を参照してください。

get-schema

次のコード例は、get-schema を使用する方法を示しています。

AWS CLI

ポリシーストアでスキーマを取得するには

次のget-schema例では、指定されたポリシーストアのスキーマの詳細を表示します。

```
aws verifiedpermissions get-schema \
  --policy-store-id PSEXAMPLEabcdefg111111
```

出力:

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "schema": "{\"MySampleNamespace\":{\"entityTypes\":{\"Employee\":{\"shape\":"
  "\":{\"attributes\":{\"jobLevel\":{\"type\":\"Long\"},\"name\":{\"type\":\"String\":"
  "\"}}},\"type\":\"Record\"}}},\"actions\":{\"remoteAccess\":{\"appliesTo\":"
  "{\"principalTypes\":[\"Employee\"]}}}}}",
  "createdDate": "2023-06-14T17:47:13.999885+00:00",
  "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

スキーマの詳細については、「Amazon Verified Permissions ユーザーガイド」の「[ポリシーストアスキーマ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSchema](#)」の「」を参照してください。

is-authorized-with-token

次のコード例は、is-authorized-with-token を使用方法を示しています。

AWS CLI

例 1: ユーザーリクエストの承認決定をリクエストするには (許可)

次のis-authorized-with-token例では、Amazon Cognito によって認証されたユーザーの認証決定をリクエストします。リクエストは、アクセストークンではなく Cognito によって提供される ID トークンを使用します。この例では、指定された情報ストアは、プリンシパルをタイプのエンティティとして返すように設定されていますCognitoUser。

```
aws verifiedpermissions is-authorized-with-token \  
  --action actionId="View",actionType="Action" \  
  --resource entityId="vacationPhoto94.jpg",entityType="Photo" \  
  --policy-store-id PSEXAMPLEEabcdefg111111 \  
  --identity-token "AbCdE12345...long.string...54321EdCbA"
```

ポリシーストアには、指定された Cognito ユーザープールとアプリケーション ID から ID を受け入れる次のステートメントを含むポリシーが含まれています。

```
permit(  
  principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0ka1bmc",  
  action,  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

出力:

```
{  
  "decision": "Allow",  
  "determiningPolicies": [  
    {  
      "determiningPolicyId": "SPEXAMPLEEabcdefg111111"  
    }  
  ]  
}
```

```
    }
  ],
  "errors": []
}
```

Cognito ユーザープールからの ID の使用の詳細については、[「Amazon Verified Permissions ユーザーガイド」の「ID プロバイダーでの Amazon Verified Permissions の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[IsAuthorizedWithToken](#)」の「」を参照してください。

is-authorized

次の例は、is-authorized を使用する方法を説明しています。

AWS CLI

例 1: ユーザーリクエストの承認決定をリクエストするには (許可)

次のis-authorized例では、Userというタイプのリソースに対して AliceupdatePhotoオペレーションを実行する Photoというタイプのプリンシパルの承認決定をリクエストしますVacationPhoto94.jpg。

レスポンスは、リクエストが 1 つのポリシーで許可されていることを示します。

```
aws verifiedpermissions is-authorized \
  --principal entityType=User,entityId=alice \
  --action actionType=Action,actionId=view \
  --resource entityType=Photo,entityId=VactionPhoto94.jpg \
  --policy-store-id PSEXAMPLEabcdefg111111
```

出力:

```
{
  "decision": "ALLOW",
  "determiningPolicies": [
    {
      "policyId": "SPEXAMPLEabcdefg111111"
    }
  ],
  "errors": []
}
```

```
}
```

例 2: ユーザーリクエストの承認決定をリクエストするには (拒否)

次の例は、プリンシパルがである点を除いて、前の例と同じですUser::"Bob"。ポリシーストアには、そのユーザーへのアクセスを許可するポリシーは含まれていませんAlbum::"alice_folder"。

出力は、のリストが空であるため、DeterminingPoliciesが暗黙的Denyであったことを示します。

```
aws verifiedpermissions create-policy \  
  --definition file://definition2.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

出力:

```
{  
  "decision": "DENY",  
  "determiningPolicies": [],  
  "errors": []  
}
```

詳細については、[「Amazon Verified Permissions ユーザーガイド」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[IsAuthorized](#)」の「」を参照してください。

list-identity-sources

次の例は、list-identity-sources を使用する方法を説明しています。

AWS CLI

使用可能な ID ソースを一覧表示するには

次のlist-identity-sources例では、指定されたポリシーストア内のすべての ID ソースを一覧表示します。

```
aws verifiedpermissions list-identity-sources \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

出力:

```
{
  "identitySources": [
    {
      "createdDate": "2023-06-12T22:27:49.150035+00:00",
      "details": {
        "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
        "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_1a2b3c4d5",
        "openIdIssuer": "COGNITO",
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_1a2b3c4d5"
      },
      "identitySourceId": "ISEXAMPLEabcdefg111111",
      "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
      "policyStoreId": "PSEXAMPLEabcdefg111111",
      "principalEntityType": "User"
    }
  ]
}
```

ID ソースの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「ID プロバイダーでの Amazon Verified Permissions の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListIdentitySources](#)」の「」を参照してください。

list-policies

次のコード例は、list-policies を使用する方法を示しています。

AWS CLI

使用可能なポリシーを一覧表示するには

次のlist-policies例では、指定されたポリシーストア内のすべてのポリシーを一覧表示します。

```
aws verifiedpermissions list-policies \
  --policy-store-id PSEXAMPLEabcdefg111111
```

出力:

```
{
  "policies": [
    {
      "createdDate": "2023-06-12T20:33:37.382907+00:00",
      "definition": {
        "static": {
          "description": "Grant everyone of janeFriends UserGroup access
to the vacationFolder Album"
        }
      },
      "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
      "policyId": "SPEXAMPLEabcdefg111111",
      "policyStoreId": "PSEXAMPLEabcdefg111111",
      "policyType": "STATIC",
      "principal": {
        "entityId": "janeFriends",
        "entityType": "UserGroup"
      },
      "resource": {
        "entityId": "vacationFolder",
        "entityType": "Album"
      }
    },
    {
      "createdDate": "2023-06-12T20:39:44.975897+00:00",
      "definition": {
        "static": {
          "description": "Grant everyone access to the publicFolder Album"
        }
      },
      "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
      "policyId": "SPEXAMPLEabcdefg222222",
      "policyStoreId": "PSEXAMPLEabcdefg111111",
      "policyType": "STATIC",
      "resource": {
        "entityId": "publicFolder",
        "entityType": "Album"
      }
    },
    {
      "createdDate": "2023-06-12T20:49:51.490211+00:00",
      "definition": {
```

```
        "templateLinked": {
            "policyTemplateId": "PTEXAMPLEabcdefghijklmnop111111"
        }
    },
    "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
    "policyId": "SPEXAMPLEabcdefghijklmnop333333",
    "policyStoreId": "PSEXAMPLEabcdefghijklmnop111111",
    "policyType": "TEMPLATE_LINKED",
    "principal": {
        "entityId": "alice",
        "entityType": "User"
    },
    "resource": {
        "entityId": "VacationPhoto94.jpg",
        "entityType": "Photo"
    }
}
]
```

ポリシーの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシー」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicies](#)」の「」を参照してください。

list-policy-stores

次の例は、list-policy-stores を使用する方法を説明しています。

AWS CLI

使用可能なポリシーストアを一覧表示するには

次のlist-policy-stores例では、AWS リージョン内のすべてのポリシーストアを一覧表示します。create-policy-store および を除く Verified Permissions のすべてのコマンドでは、操作するポリシーストアの ID を指定list-policy-storesする必要があります。

```
aws verifiedpermissions list-policy-stores
```

出力:

```
{
  "policyStores": [
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg111111",
      "createdDate": "2023-06-05T20:16:46.225598+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg111111"
    },
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg222222",
      "createdDate": "2023-06-08T18:09:37.364356+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg222222"
    },
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg333333",
      "createdDate": "2023-06-08T18:09:46.920600+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg333333"
    }
  ]
}
```

ポリシーストアの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーストア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicyStores](#)」の「」を参照してください。

list-policy-templates

次の例は、list-policy-templates を使用する方法を説明しています。

AWS CLI

使用可能なポリシーテンプレートを一覧表示するには

次のlist-policy-templates例では、指定されたポリシーストア内のすべてのポリシーテンプレートを一覧表示します。

```
aws verifiedpermissions list-policy-templates \
  --policy-store-id PSEXAMPLEEabcdefg111111
```


出力:

```
{
  "policyTemplates": [
    {
      "createdDate": "2023-06-12T20:47:42.804511+00:00",
      "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
      "policyStoreId": "PSEXAMPLEabcdefg111111",
      "policyTemplateId": "PTEXAMPLEabcdefg111111"
    }
  ]
}
```

ポリシーテンプレートの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーテンプレート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListPolicyTemplates](#)」の「」を参照してください。

put-schema

次のコード例は、put-schema を使用する方法を示しています。

AWS CLI

スキーマをポリシーストアに保存するには

次のput-schema例では、指定されたポリシーストアでスキーマを作成または置き換えます。

入力ファイルの cedarJsonパラメータは、JSONオブジェクトの文字列表現を取得します。これには、最も外側の引用符ペア内に埋め込まれた引用符 (「) が含まれます。そのためには、すべての埋め込み引用符にバックスラッシュ文字 (「\」) を付け、すべての行を改行のない 1 つのテキスト行に結合して、JSONを文字列に変換する必要があります。

文字列の例は、読みやすくするためにここで複数の行にラップして表示できますが、オペレーションではパラメータを 1 行の文字列として送信する必要があります。

```
aws verifiedpermissions put-schema --definition file://schema.txt --policy-store-id
PSEXAMPLEabcdefg111111
```

schema.txt の内容:

```
{
  "cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {
    \"appliesTo\": {\"principalTypes\": [\"Employee\"]}},\"entityTypes\": {
    \"Employee\": {\"shape\": {\"attributes\": {\"jobLevel\": {\"type\":
    \"Long\"}},\"name\": {\"type\": \"String\"}},\"type\": \"Record\"}}}}}"
}
```

出力:

```
{
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "namespaces": [
    "MySampleNamespace"
  ],
  "createdDate": "2023-06-14T17:47:13.999885+00:00",
  "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

スキーマの詳細については、「Amazon Verified Permissions ユーザーガイド」の「[ポリシーストアスキーマ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutSchema](#)」の「」を参照してください。

update-identity-source

次の例は、update-identity-source を使用する方法を説明しています。

AWS CLI

ID ソースを更新するには

次のupdate-identity-source例では、新しい Cognito ユーザープール設定を指定し、ID ソースによって返されるエンティティタイプを変更することで、指定された ID ソースを変更します。

```
aws verifiedpermissions update-identity-source
--identity-source-id ISEXAMPLEEabcdefg111111 \
--update-configuration file://config.txt \
--principal-entity-type "Employee" \
--policy-store-id PSEXAMPLEEabcdefg111111
```

config.txt の内容:

```
{
  "cognitoUserPoolConfiguration": {
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_1a2b3c4d5",
    "clientIds": ["a1b2c3d4e5f6g7h8i9j0kalbmc"]
  }
}
```

出力:

```
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefgh111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

ID ソースの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「ID プロバイダーでの Amazon Verified Permissions の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIdentitySource](#)」の「」を参照してください。

update-policy-store

次のコード例は、update-policy-store を使用する方法を示しています。

AWS CLI

ポリシーストアを更新するには

次のupdate-policy-store例では、検証設定を変更してポリシーストアを変更します。

```
aws verifiedpermissions update-policy-store \
  --validation-settings "mode=STRICT" \
  --policy-store-id PSEXAMPLEabcdefgh111111
```

出力:

```
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg111111",
  "createdDate": "2023-05-16T17:41:29.103459+00:00",
  "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",
  "policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

ポリシーストアの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーストア」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePolicyStore](#)」の「」を参照してください。

update-policy-template

次の例は、update-policy-template を使用する方法を説明しています。

AWS CLI

例 1: ポリシーテンプレートを更新するには

次のupdate-policy-template例では、指定されたテンプレートリンクポリシーを変更して、ポリシーステートメントを置き換えます。

```
aws verifiedpermissions update-policy-template \
  --policy-template-id PTEXAMPLEEabcdefg111111 \
  --statement file://template1.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

template1.txt ファイルの内容。

```
permit(
  principal in ?principal,
  action == Action::"view",
  resource == Photo::"VacationPhoto94.jpg"
);
```

出力:

```
{
```

```
"createdDate": "2023-06-12T20:47:42.804511+00:00",
"lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
"policyStoreId": "PSEXAMPLEEabcdefg111111",
"policyTemplateId": "PTEXAMPLEEabcdefg111111"
}
```

ポリシーテンプレートの詳細については、[「Amazon Verified Permissions ユーザーガイド」の「Amazon Verified Permissions ポリシーテンプレート」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePolicyTemplate](#)」の「」を参照してください。

update-policy

次のコード例は、update-policy を使用する方法を示しています。

AWS CLI

例 1: 静的ポリシーを作成するには

次のcreate-policy例では、プリンシパルとリソースの両方を指定するポリシースコープを持つ静的ポリシーを作成します。

```
aws verifiedpermissions create-policy \  
  --definition file://definition.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

statement パラメータは、JSONオブジェクトの文字列表現を取得します。これには、最も外側の引用符ペア内に埋め込まれた引用符 (「) が含まれます。そのためには、すべての埋め込み引用符にバックスラッシュ文字 (「\」) を付け、すべての行を改行のない 1 つのテキスト行に結合して、JSONを文字列に変換する必要があります。

文字列の例は、読みやすくするためにここで複数の行にラップして表示できますが、オペレーションではパラメータを 1 行の文字列として送信する必要があります。

definition.txt ファイルの内容。

```
{  
  "static": {  
    "description": "Grant everyone of janeFriends UserGroup access to the  
vacationFolder Album",
```

```

    "statement": "permit(principal in UserGroup::\\"janeFriends\\", action,
resource in Album::\\"vacationFolder\" );"
  }
}

```

出力:

```

{
  "createdDate": "2023-06-12T20:33:37.382907+00:00",
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",
  "policyId": "SPEXAMPLEabcdefg111111",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyType": "STATIC",
  "principal": {
    "entityId": "janeFriends",
    "entityType": "UserGroup"
  },
  "resource": {
    "entityId": "vacationFolder",
    "entityType": "Album"
  }
}

```

例 2: リソースへのアクセスを全員に許可する静的ポリシーを作成するには

次のcreate-policy例では、リソースのみを指定するポリシースコープを持つ静的ポリシーを作成します。

```

aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEabcdefg111111

```

definition2.txt ファイルの内容。

```

{
  "static": {
    "description": "Grant everyone access to the publicFolder Album",
    "statement": "permit(principal, action, resource in Album::\\"publicFolder
\\");"
  }
}

```

出力:

```
{
  "createdDate": "2023-06-12T20:39:44.975897+00:00",
  "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
  "policyId": "PbfR73F8oh5MMfr9uRtFDB",
  "policyStoreId": "PSEXAMPLEEabcdefg222222",
  "policyType": "STATIC",
  "resource": {
    "entityId": "publicFolder",
    "entityType": "Album"
  }
}
```

例 3: 指定されたテンプレートに関連付けられているテンプレートにリンクされたポリシーを作成するには

次のcreate-policy例では、指定されたポリシーテンプレートを使用してテンプレートリンクポリシーを作成し、使用する指定されたプリンシパルを新しいテンプレートリンクポリシーに関連付けます。

```
aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

definition3.txt の内容 :

```
{
  "templateLinked": {
    "policyTemplateId": "PTEXAMPLEEabcdefg111111",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    }
  }
}
```

出力:

```
{
  "createdDate": "2023-06-12T20:49:51.490211+00:00",
```

```
"lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",
"policyId": "TPEXAMPLEabcdefg111111",
"policyStoreId": "PSEXAMPLEabcdefg111111",
"policyType": "TEMPLATE_LINKED",
"principal": {
  "entityId": "alice",
  "entityType": "User"
},
"resource": {
  "entityId": "VacationPhoto94.jpg",
  "entityType": "Photo"
}
}
```

ポリシーの詳細については、[「Amazon Verified Permissions ユーザーガイド」](#)の「[Amazon Verified Permissions ポリシー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePolicy](#)」の「」を参照してください。

VPC を使用した格子の例 AWS CLI

次のコード例は、Lattice VPC AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-listener

次のコード例は、create-listener を使用する方法を示しています。

AWS CLI

リスナーを作成するには

次のcreate-listener例では、指定した Lattice VPC ターゲットグループにトラフィックを転送するデフォルトのルールを持つHTTPSリスナーを作成します。

```
aws vpc-lattice create-listener \  
  --name my-service-listener \  
  --protocol HTTPS \  
  --port 443 \  
  --service-identifier svc-0285b53b2eEXAMPLE \  
  --default-action file://listener-config.json
```

listener-config.json の内容:

```
{  
  "forward": {  
    "targetGroups": [  
      {  
        "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE"  
      }  
    ]  
  }  
}
```

出力:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/  
svc-0285b53b2eEXAMPLE/listener/listener-07cc7fb0abEXAMPLE",  
  "defaultAction": {  
    "forward": {  
      "targetGroups": [  
        {  
          "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE",  
          "weight": 100  
        }  
      ]  
    }  
  },  
  "id": "listener-07cc7fb0abEXAMPLE",
```

```
"name": "my-service-listener",
"port": 443,
"protocol": "HTTPS",
"serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
"serviceId": "svc-0285b53b2eEXAMPLE"
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[リスナー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateListener](#)」の「」を参照してください。

create-service-network-service-association

次の例は、create-service-network-service-association を使用する方法を説明しています。

AWS CLI

サービス関連付けを作成するには

次のcreate-service-network-service-association例では、指定されたサービスを指定されたサービスネットワークに関連付けます。

```
aws vpc-lattice create-service-network-service-association \
  --service-identifier svc-0285b53b2eEXAMPLE \
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkserviceassociation/snsa-0e16955a8cEXAMPLE",
  "createdBy": "123456789012",
  "dnsEntry": {
    "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-svcs.us-east-2.on.aws",
    "hostedZoneId": "Z09127221KTH2CEXAMPLE"
  },
  "id": "snsa-0e16955a8cEXAMPLE",
  "status": "CREATE_IN_PROGRESS"
}
```

```
}
```

詳細については、[「Amazon Lattice ユーザーガイド」の「サービス関連付けの管理」](#)を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateServiceNetworkServiceAssociation](#)」の「」を参照してください。

create-service-network-vpc-association

次のコード例は、create-service-network-vpc-association を使用する方法を示しています。

AWS CLI

VPC関連付けを作成するには

次のcreate-service-network-vpc-association例では、指定された vpc を指定されたサービスネットワークに関連付けます。指定されたセキュリティグループはVPC、のサービスネットワークとそのサービスにアクセスできるリソースを制御します。

```
aws vpc-lattice create-service-network-vpc-association \  
  --vpc-identifier vpc-0a1b2c3d4eEXAMPLE \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --security-group-ids sg-0aee16bc6cEXAMPLE
```

出力:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/  
snva-0821fc8631EXAMPLE",  
  "createdBy": "123456789012",  
  "id": "snva-0821fc8631EXAMPLE",  
  "securityGroupIds": [  
    "sg-0aee16bc6cEXAMPLE"  
  ],  
  "status": "CREATE_IN_PROGRESS"  
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の[VPC「関連付けの管理」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateServiceNetworkVpcAssociation](#)」の「」を参照してください。

create-service-network

次のコード例は、create-service-network を使用する方法を示しています。

AWS CLI

サービスネットワークを作成するには

次のcreate-service-network例では、指定された名前のサービスネットワークを作成します。

```
aws vpc-lattice create-service-network \  
  --name my-service-network
```

出力:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/  
sn-080ec7dc93EXAMPLE",  
  "authType": "NONE",  
  "id": "sn-080ec7dc93EXAMPLE",  
  "name": "my-service-network"  
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[サービスネットワーク](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateServiceNetwork](#)」の「」を参照してください。

create-service

次のコード例は、create-service を使用する方法を示しています。

AWS CLI

サービスを作成するには

次のcreate-service例では、指定された名前のサービスを作成します。

```
aws vpc-lattice create-service \  
  --name my-lattice-service
```

出力:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/  
svc-0285b53b2eEXAMPLE",  
  "authType": "NONE",  
  "dnsEntry": {  
    "domainName": "my-lattice-service-0285b53b2eEXAMPLE.1a2b3c4.vpc-lattice-  
svcs.us-east-2.on.aws",  
    "hostedZoneId": "Z09127221KTH2CEXAMPLE"  
  },  
  "id": "svc-0285b53b2eEXAMPLE",  
  "name": "my-lattice-service",  
  "status": "CREATE_IN_PROGRESS"  
}
```

詳細については、「Amazon [Lattice VPC ユーザーガイド](#)」の「Lattice のサービス」を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateService](#)」の「」を参照してください。

create-target-group

次のコード例は、create-target-group を使用する方法を示しています。

AWS CLI

例 1: タイプのターゲットグループを作成するには INSTANCE

次のcreate-target-group例では、指定された名前、タイプ、および設定を持つターゲットグループを作成します。

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-instance \  
  --type INSTANCE \  
  --target-arn arn:aws:ec2:us-east-2:123456789012:instance/i-123456789012
```

```
--config file://tg-config.json
```

tg-config.json の内容:

```
{
  "port": 443,
  "protocol": "HTTPS",
  "protocolVersion": "HTTP1",
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
  "config": {
    "healthCheck": {
      "enabled": true,
      "healthCheckIntervalSeconds": 30,
      "healthCheckTimeoutSeconds": 5,
      "healthyThresholdCount": 5,
      "matcher": {
        "httpCode": "200"
      },
      "path": "/",
      "protocol": "HTTPS",
      "protocolVersion": "HTTP1",
      "unhealthyThresholdCount": 2
    },
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
  },
  "id": "tg-0eaa4b9ab4EXAMPLE",
  "name": "my-lattice-target-group-instance",
  "status": "CREATE_IN_PROGRESS",
  "type": "INSTANCE"
}
```

例 2: IP タイプのターゲットグループを作成するには

次のcreate-target-group例では、指定された名前、タイプ、および設定を持つターゲットグループを作成します。

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-ip \  
  --type IP \  
  --config file://tg-config.json
```

tg-config.json の内容:

```
{  
  "ipAddressType": "IPV4",  
  "port": 443,  
  "protocol": "HTTPS",  
  "protocolVersion": "HTTP1",  
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

出力:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "config": {  
    "healthCheck": {  
      "enabled": true,  
      "healthCheckIntervalSeconds": 30,  
      "healthCheckTimeoutSeconds": 5,  
      "healthyThresholdCount": 5,  
      "matcher": {  
        "httpCode": "200"  
      },  
      "path": "/",  
      "protocol": "HTTPS",  
      "protocolVersion": "HTTP1",  
      "unhealthyThresholdCount": 2  
    },  
    "ipAddressType": "IPV4",  
    "port": 443,  
    "protocol": "HTTPS",  
    "protocolVersion": "HTTP1",  
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
  }
```

```
  },  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "name": "my-lattice-target-group-ip",  
  "status": "CREATE_IN_PROGRESS",  
  "type": "IP"  
}
```

例 3: タイプのターゲットグループを作成するには LAMBDA

次のcreate-target-group例では、指定された名前、タイプ、および設定を持つターゲットグループを作成します。

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-lambda \  
  --type LAMBDA
```

出力:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "name": "my-lattice-target-group-lambda",  
  "status": "CREATE_IN_PROGRESS",  
  "type": "LAMBDA"  
}
```

例 4: タイプのターゲットグループを作成するには ALB

次のcreate-target-group例では、指定された名前、タイプ、および設定を持つターゲットグループを作成します。

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-alb \  
  --type ALB \  
  --config file://tg-config.json
```

tg-config.json の内容:

```
{  
  "port": 443,  
}
```



```
"protocol": "HTTPS",
"protocolVersion": "HTTP1",
"vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
  "config": {
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
  },
  "id": "tg-0eaa4b9ab4EXAMPLE",
  "name": "my-lattice-target-group-alb",
  "status": "CREATE_IN_PROGRESS",
  "type": "ALB"
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ターゲットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTargetGroup](#)」の「」を参照してください。

delete-auth-policy

次のコード例は、delete-auth-policy を使用する方法を示しています。

AWS CLI

認証ポリシーを削除するには

次のdelete-auth-policy例では、指定されたサービスの認証ポリシーを削除します。

```
aws vpc-lattice delete-auth-policy \
  --resource-identifier svc-0285b53b2eEXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「Amazon VPCLattice ユーザーガイド」の「[認証ポリシー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAuthPolicy](#)」の「」を参照してください。

delete-listener

次の例は、delete-listener を使用する方法を説明しています。

AWS CLI

リスナーを削除するには

次のdelete-listener例では、指定されたリスナーを削除します。

```
aws vpc-lattice delete-listener \  
  --listener-identifier Listener-07cc7fb0abEXAMPLE \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[リスナー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteListener](#)」の「」を参照してください。

delete-service-network-service-association

次の例は、delete-service-network-service-association を使用する方法を説明しています。

AWS CLI

サービス関連付けを削除するには

次のdelete-service-network-service-association例では、指定されたサービス関連付けの関連付けを解除します。

```
aws vpc-lattice delete-service-network-service-association \  
  --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
  "id": "snsa-031fabb4d8EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の [「サービス関連付けの管理」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteServiceNetworkServiceAssociation](#)」の「」を参照してください。

delete-service-network-vpc-association

次の例は、delete-service-network-vpc-association を使用する方法を説明しています。

AWS CLI

VPC 関連付けを削除するには

次の delete-service-network-vpc-association 例では、指定された VPC 関連付けの関連付けを解除します。

```
aws vpc-lattice delete-service-network-vpc-association \
  --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/snva-0821fc8631EXAMPLE",
  "id": "snva-0821fc8631EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

詳細については、[「Amazon Lattice ユーザーガイド」](#)の VPC [「関連付けの管理」](#) を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteServiceNetworkVpcAssociation](#)」の「」を参照してください。

delete-service-network

次のコード例は、delete-service-network を使用する方法を示しています。

AWS CLI

サービスネットワークを削除するには

次のdelete-service-network例では、指定されたサービスネットワークを削除します。

```
aws vpc-lattice delete-service-network \  
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

このコマンドでは何も出力されません。

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[サービスネットワーク](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteServiceNetwork](#)」の「」を参照してください。

delete-service

次のコード例は、delete-service を使用する方法を示しています。

AWS CLI

サービスを削除するには

次のdelete-service例では、指定されたサービスを削除します。

```
aws vpc-lattice delete-service \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

出力:

```
{  
  "arn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/  
  svc-0285b53b2eEXAMPLE",
```

```
"id": "svc-0285b53b2eEXAMPLE",
"name": "my-lattice-service",
"status": "DELETE_IN_PROGRESS"
}
```

詳細については、「Amazon [Lattice VPC ユーザーガイド](#)」の「Lattice のサービス」を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteService](#)」の「」を参照してください。

delete-target-group

次の例は、delete-target-group を使用する方法を説明しています。

AWS CLI

ターゲットグループを削除するには

次の delete-target-group の例は、指定されたターゲットグループを削除します。

```
aws vpc-lattice delete-target-group \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
  "id": "tg-0eaa4b9ab4EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ターゲットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTargetGroup](#)」の「」を参照してください。

deregister-targets

次のコード例は、deregister-targets を使用する方法を示しています。

AWS CLI

ターゲットの登録を解除するには

次のderegister-targets例では、指定されたターゲットグループから指定されたターゲットを登録解除します。

```
aws vpc-lattice deregister-targets \  
  --targets i-07dd579bc5EXAMPLE \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

出力:

```
{  
  "successful": [  
    {  
      "id": "i-07dd579bc5EXAMPLE",  
      "port": 443  
    }  
  ],  
  "unsuccessful": []  
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ターゲットの登録](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterTargets](#)」の「」を参照してください。

get-auth-policy

次のコード例は、get-auth-policy を使用する方法を示しています。

AWS CLI

認証ポリシーに関する情報を取得するには

次のget-auth-policy例では、指定されたサービスの認証ポリシーに関する情報を取得します。

```
aws vpc-lattice get-auth-policy \  
  --service-id EXAMPLE
```

```
--resource-identifier svc-0285b53b2eEXAMPLE
```

出力:

```
{
  "createdAt": "2023-06-07T03:51:20.266Z",
  "lastUpdatedAt": "2023-06-07T04:39:27.082Z",
  "policy": "{\\"Version\\":\\"2012-10-17\\",\\"Statement\\":[{\\"Effect\\":\\"Allow\\",\\"Principal\\":{\\"AWS\\":\\"arn:aws:iam:123456789012:role/my-clients\\"},\\"Action\\":\\"vpc-lattice-svcs:Invoke\\",\\"Resource\\":\\"arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE\\"}]}",
  "state": "Active"
}
```

詳細については、「Amazon VPCLattice ユーザーガイド」の「[認証ポリシー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetAuthPolicy](#)」の「」を参照してください。

get-listener

次の例は、get-listener を使用する方法を説明しています。

AWS CLI

サービスリスナーに関する情報を取得するには

次のget-listener例では、指定されたサービスの指定されたリスナーに関する情報を取得します。

```
aws vpc-lattice get-listener \
  --listener-identifier listener-0ccf55918cEXAMPLE \
  --service-identifier svc-0285b53b2eEXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
  "createdAt": "2023-05-07T05:08:45.192Z",
  "defaultAction": {
```

```
    "forward": {
      "targetGroups": [
        {
          "targetGroupIdentifier": "tg-0ff213abb6EXAMPLE",
          "weight": 1
        }
      ]
    },
    "id": "listener-0ccf55918cEXAMPLE",
    "lastUpdatedAt": "2023-05-07T05:08:45.192Z",
    "name": "http-80",
    "port": 80,
    "protocol": "HTTP",
    "serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",
    "serviceId": "svc-0285b53b2eEXAMPLE"
  }
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ルーティングの定義](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetListener](#)」の「」を参照してください。

get-service-network-service-association

次の例は、get-service-network-service-association を使用する方法を説明しています。

AWS CLI

サービス関連付けに関する情報を取得するには

次のget-service-network-service-association例では、指定されたサービス関連付けに関する情報を取得します。

```
aws vpc-lattice get-service-network-service-association \
  --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

出力:

```
{
```



```
"arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
"createdAt": "2023-05-05T21:48:16.076Z",
"createdBy": "123456789012",
"dnsEntry": {
  "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-svcs.us-east-2.on.aws",
  "hostedZoneId": "Z09127221KTH2CEXAMPLE"
},
"id": "snsa-031fabb4d8EXAMPLE",
"serviceArn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",
"serviceId": "svc-0285b53b2eEXAMPLE",
"serviceName": "my-lattice-service",
"serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/sn-080ec7dc93EXAMPLE",
"serviceNetworkId": "sn-080ec7dc93EXAMPLE",
"serviceNetworkName": "my-service-network",
"status": "ACTIVE"
}
```

詳細については、[「Amazon Lattice ユーザーガイド」の「サービス関連付けの管理」](#)を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceNetworkServiceAssociation](#)」の「」を参照してください。

get-service-network-vpc-association

次の例は、`get-service-network-vpc-association` を使用する方法を説明しています。

AWS CLI

VPC関連付けに関する情報を取得するには

次の`get-service-network-vpc-association`例では、指定されたVPC関連付けに関する情報を取得します。

```
aws vpc-lattice get-service-network-vpc-association \
  --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
  "createdAt": "2023-06-06T23:41:08.421Z",
  "createdBy": "123456789012",
  "id": "snva-0c5dcb60d6EXAMPLE",
  "lastUpdatedAt": "2023-06-06T23:41:08.421Z",
  "securityGroupIds": [
    "sg-0aee16bc6cEXAMPLE"
  ],
  "serviceNetworkArn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
  "serviceNetworkId": "sn-080ec7dc93EXAMPLE",
  "serviceNetworkName": "my-service-network",
  "status": "ACTIVE",
  "vpcId": "vpc-0a1b2c3d4eEXAMPLE"
}
```

詳細については、[「Amazon Lattice ユーザーガイド」のVPC「関連付けの管理」](#)を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceNetworkVpcAssociation](#)」の「」を参照してください。

get-service-network

次の例は、get-service-network を使用する方法を説明しています。

AWS CLI

サービスネットワークに関する情報を取得するには

次のget-service-network例では、指定されたサービスネットワークに関する情報を取得します。

```
aws vpc-lattice get-service-network \
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

出力:

```
{
```

```
"arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
"authType": "AWS_IAM",
"createdAt": "2023-05-05T15:26:08.417Z",
"id": "sn-080ec7dc93EXAMPLE",
"lastUpdatedAt": "2023-05-05T15:26:08.417Z",
"name": "my-service-network",
"numberOfAssociatedServices": 2,
"numberOfAssociatedVPCs": 3
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[サービスネットワーク](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceNetwork](#)」の「」を参照してください。

get-service

次の例は、get-service を使用する方法を説明しています。

AWS CLI

サービスに関する情報を取得するには

次のget-service例では、指定されたサービスに関する情報を取得します。

```
aws vpc-lattice get-service \
  --service-identifier svc-0285b53b2eEXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
  "authType": "AWS_IAM",
  "createdAt": "2023-05-05T21:35:29.339Z",
  "dnsEntry": {
    "domainName": "my-lattice-service-0285b53b2eEXAMPLE.7d67968.vpc-lattice-
svcs.us-east-2.on.aws",
    "hostedZoneId": "Z09127221KTH2CFU0HIZH"
  },
}
```

```
"id": "svc-0285b53b2eEXAMPLE",
"lastUpdatedAt": "2023-05-05T21:35:29.339Z",
"name": "my-lattice-service",
"status": "ACTIVE"
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[サービス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetService](#)」の「」を参照してください。

get-target-group

次の例は、get-target-group を使用する方法を説明しています。

AWS CLI

ターゲットグループに関する情報を取得するには

次のget-target-group例では、ターゲットタイプが `INSTANCE` である指定されたターゲットグループに関する情報を取得します。

```
aws vpc-lattice get-target-group \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

出力:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
  "config": {
    "healthCheck": {
      "enabled": true,
      "healthCheckIntervalSeconds": 30,
      "healthCheckTimeoutSeconds": 5,
      "healthyThresholdCount": 5,
      "matcher": {
        "httpCode": "200"
      },
      "path": "/",
      "protocol": "HTTPS",
      "protocolVersion": "HTTP1",

```

```
        "unhealthyThresholdCount": 2
      },
      "port": 443,
      "protocol": "HTTPS",
      "protocolVersion": "HTTP1",
      "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
    },
    "createdAt": "2023-05-06T04:41:04.122Z",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "lastUpdatedAt": "2023-05-06T04:41:04.122Z",
    "name": "my-target-group",
    "serviceArns": [
      "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE"
    ],
    "status": "ACTIVE",
    "type": "INSTANCE"
  }
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ターゲットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTargetGroup](#)」の「」を参照してください。

list-listeners

次の例は、list-listeners を使用する方法を説明しています。

AWS CLI

サービスリスナーを一覧表示するには

次のlist-listeners例では、指定されたサービスのリスナーを一覧表示します。

```
aws vpc-lattice list-listeners \
  --service-identifier svc-0285b53b2eEXAMPLE
```

出力:

```
{
  "items": [
    {
```

```
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:service/
svc-0285b53b2eEXAMPLE/listener/listener-0ccf55918cEXAMPLE",
    "createdAt": "2023-05-07T05:08:45.192Z",
    "id": "listener-0ccf55918cEXAMPLE",
    "lastUpdatedAt": "2023-05-07T05:08:45.192Z",
    "name": "http-80",
    "port": 80,
    "protocol": "HTTP"
  }
]
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ルーティングの定義](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListListeners](#)」の「」を参照してください。

list-service-network-service-associations

次の例は、list-service-network-service-associations を使用する方法を説明しています。

AWS CLI

サービス関連付けを一覧表示するには

次のlist-service-network-service-associations例では、指定されたサービスネットワークのサービス関連付けを一覧表示します。--query オプションは、出力をサービス関連付け IDsの にスコープします。

```
aws vpc-lattice list-service-network-service-associations \
  --service-network-identifier sn-080ec7dc93EXAMPLE \
  --query items[*].id
```

出力:

```
[
  "snsa-031fabb4d8EXAMPLE",
  "snsa-0e16955a8cEXAMPLE"
]
```

詳細については、[「Amazon Lattice ユーザーガイド」の「サービス関連付けの管理」](#)を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServiceNetworkServiceAssociations](#)」の「」を参照してください。

list-service-network-vpc-associations

次の例は、list-service-network-vpc-associations を使用する方法を説明しています。

AWS CLI

VPC 関連付けを一覧表示するには

次のlist-service-network-vpc-associations例では、指定されたサービスネットワークのVPC 関連付けを一覧表示します。 --query オプションは、VPC 関連付けIDsの への出力をスコープします。

```
aws vpc-lattice list-service-network-vpc-associations \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --query items[*].id
```

出力:

```
[  
  "snva-0821fc8631EXAMPLE",  
  "snva-0c5dcb60d6EXAMPLE"  
]
```

詳細については、[「Amazon Lattice ユーザーガイド」のVPC「関連付けの管理」](#)を参照してください。 VPC

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServiceNetworkVpcAssociations](#)」の「」を参照してください。

list-service-networks

次のコード例は、list-service-networks を使用する方法を示しています。

AWS CLI

サービスネットワークを一覧表示するには

次のlist-service-networks例では、呼び出し元アカウントを所有または共有しているサービスネットワークを一覧表示します。--query オプションは、結果をサービスネットワークの Amazon リソースネーム (ARN) にスコープします。

```
aws vpc-lattice list-service-networks \  
  --query items[*].arn
```

出力:

```
[  
  "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/  
sn-080ec7dc93EXAMPLE",  
  "arn:aws:vpc-lattice:us-east-2:111122223333:servicenetwork/sn-0ec4d436cfEXAMPLE"  
]
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[サービスネットワーク](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServiceNetworks](#)」の「」を参照してください。

list-services

次の例は、list-services を使用する方法を説明しています。

AWS CLI

サービスを一覧表示するには

次のlist-services例では、呼び出し元アカウントを所有または共有している サービスを一覧表示します。--query オプションは、結果をサービスの Amazon リソースネーム (ARN) にスコープします。

```
aws vpc-lattice list-services \  
  --query items[*].arn
```

出力:

```
[  
  "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",
```



```
"arn:aws:vpc-lattice:us-east-2:111122223333:service/svc-0b8ac96550EXAMPLE"  
]
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[サービス](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListServices](#)」の「」を参照してください。

list-target-groups

次の例は、list-target-groups を使用する方法を説明しています。

AWS CLI

ターゲットグループを一覧表示するには

次のlist-target-groups例では、ターゲットタイプが のターゲットグループを一覧表示し
ますLAMBDA。

```
aws vpc-lattice list-target-groups \  
  --target-group-type LAMBDA
```

出力:

```
{  
  "items": [  
    {  
      "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-045c1b7d9dEXAMPLE",  
      "createdAt": "2023-05-06T05:22:16.637Z",  
      "id": "tg-045c1b7d9dEXAMPLE",  
      "lastUpdatedAt": "2023-05-06T05:22:16.637Z",  
      "name": "my-target-group-lam",  
      "serviceArns": [  
        "arn:aws:vpc-lattice:us-east-2:123456789012:service/  
svc-0285b53b2eEXAMPLE"  
      ],  
      "status": "ACTIVE",  
      "type": "LAMBDA"  
    }  
  ]  
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ターゲットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTargetGroups](#)」の「」を参照してください。

list-targets

次の例は、list-targets を使用する方法を説明しています。

AWS CLI

ターゲットグループのターゲットを一覧表示するには

次のlist-targets例では、指定されたターゲットグループのターゲットを一覧表示します。

```
aws vpc-lattice list-targets \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

出力:

```
{  
  "items": [  
    {  
      "id": "i-07dd579bc5EXAMPLE",  
      "port": 443,  
      "status": "HEALTHY"  
    },  
    {  
      "id": "i-047b3c9078EXAMPLE",  
      "port": 443,  
      "reasonCode": "HealthCheckFailed",  
      "status": "UNHEALTHY"  
    }  
  ]  
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ターゲットグループ](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTargets](#)」の「」を参照してください。

put-auth-policy

次の例は、put-auth-policy を使用する方法を説明しています。

AWS CLI

サービスの認証ポリシーを作成するには

次のput-auth-policy例では、指定されたIAMロールを使用する認証されたプリンシパルからのリクエストへのアクセスを許可します。リソースは、ポリシーARNがアタッチされているサービスのです。

```
aws vpc-lattice put-auth-policy \  
  --resource-identifier svc-0285b53b2eEXAMPLE \  
  --policy file://auth-policy.json
```

auth-policy.json の内容:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::123456789012:role/my-clients"  
      },  
      "Action": "vpc-lattice-svcs:Invoke",  
      "Resource": "arn:aws:vpc-lattice:us-east-2:123456789012:service/  
svc-0285b53b2eEXAMPLE"  
    }  
  ]  
}
```

出力:

```
{  
  "policy": "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Effect\":\"Allow\",  
\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:role/my-clients\"},  
\"Action\":\"vpc-lattice-svcs:Invoke\",\"Resource\":\"arn:aws:vpc-lattice:us-  
east-2:123456789012:service/svc-0285b53b2eEXAMPLE\"}]}",  
  "state": "Active"
```

```
}
```

詳細については、「Amazon VPCLattice ユーザーガイド」の「[認証ポリシー](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutAuthPolicy](#)」の「」を参照してください。

register-targets

次のコード例は、register-targets を使用する方法を示しています。

AWS CLI

ターゲットを登録するには

次のregister-targets例では、指定されたターゲットを指定されたターゲットグループに登録します。

```
aws vpc-lattice register-targets \  
  --targets id=i-047b3c9078EXAMPLE id=i-07dd579bc5EXAMPLE \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

出力:

```
{  
  "successful": [  
    {  
      "id": "i-07dd579bc5EXAMPLE",  
      "port": 443  
    }  
  ],  
  "unsuccessful": [  
    {  
      "failureCode": "UnsupportedTarget",  
      "failureMessage": "Instance targets must be in the same VPC as their  
target group",  
      "id": "i-047b3c9078EXAMPLE",  
      "port": 443  
    }  
  ]  
}
```

詳細については、「Amazon Lattice VPC ユーザーガイド」の「[ターゲットの登録](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterTargets](#)」の「」を参照してください。

AWS WAF Classic を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS WAF Classic。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

put-logging-configuration

次の例は、put-logging-configuration を使用する方法を説明しています。

AWS CLI

指定された Kinesis Firehose ストリームACLARNを使用してウェブのログ記録設定を作成するには ARN

次のput-logging-configuration例は、WAFと のログ記録設定を示しています CloudFront。

```
aws waf put-logging-configuration \  
  --logging-configuration ResourceArn=arn:aws:waf::123456789012:webacl/3bffd3ed-  
fa2e-445e-869f-a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-  
east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[]
```

出力:

```
{
  "LoggingConfiguration": {
    "ResourceArn": "arn:aws:waf::123456789012:webacl/3bffd3ed-fa2e-445e-869f-
a6a7cf153fd3",
    "LogDestinationConfigs": [
      "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-
firehose-stream"
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLoggingConfiguration](#)」の「」を参照してください。

update-byte-match-set

次のコード例は、update-byte-match-set を使用する方法を示しています。

AWS CLI

バイト一致セットを更新するには

次のupdate-byte-match-setコマンドは、内の ByteMatchTuple オブジェクト (フィルター) を削除します ByteMatchSet。

```
aws waf update-byte-match-set --byte-match-set-id a123fae4-
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --
updates
Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="b
```

詳細については、デAWS WAFベロツパーガイドの「文字列一致条件の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateByteMatchSet](#)」の「」を参照してください。

update-ip-set

次のコード例は、update-ip-set を使用する方法を示しています。

AWS CLI

IP セットを更新するには

次のupdate-ip-setコマンドは、IPv4 アドレスIPSetを使用して を更新し、IPv6アドレスを削除します。

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
Action="INSERT",IPSetDescriptor={Type="IPv4",Value="12.34.56.78/16"},Action="DELETE",IPSetD
```

または、JSON ファイルを使用して入力を指定することもできます。例:

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-
token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates file://change.json
```

JSON ファイルの内容が次の場合 :

```
[
{
  "Action": "INSERT",
  "IPSetDescriptor":
  {
    "Type": "IPv4",
    "Value": "12.34.56.78/16"
  }
},
{
  "Action": "DELETE",
  "IPSetDescriptor":
  {
    "Type": "IPv6",
    "Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
  }
}
]
```

詳細については、デAWS WAFベロッパーガイドの「IP マッチ条件の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIpSet](#)」の「」を参照してください。

update-rule

次の例は、update-rule を使用する方法を説明しています。

AWS CLI

ルールを更新するには

次のupdate-ruleコマンドは、ルール内の述語オブジェクトを削除します。

```
aws waf update-rule --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}
```

詳細については、デAWS WAFベロツパーガイドの「ルールの使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRule](#)」の「」を参照してください。

update-size-constraint-set

次のコード例は、update-size-constraint-set を使用する方法を示しています。

AWS CLI

サイズ制約セットを更新するには

次のupdate-size-constraint-setコマンドは、サイズ制約セット内の SizeConstraint オブジェクト (フィルター) を削除します。

```
aws waf update-size-constraint-set --size-constraint-set-id a123fae4-
b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --
updates
Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NONE"}
```

詳細については、デAWS WAFベロツパーガイドの「サイズ制約条件の使用」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSizeConstraintSet](#)」の「」を参照してください。

update-sql-injection-match-set

次のコード例は、update-sql-injection-match-set を使用する方法を示しています。

AWS CLI

SQLインジェクションマッチセットを更新するには

次のupdate-sql-injection-match-setコマンドは、SQLインジェクション一致セット内のSqlInjectionMatchTuple オブジェクト (フィルター) を削除します。

```
aws waf update-sql-injection-match-set --sql-injection-  
match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --  
change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates  
Action="DELETE",SqlInjectionMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation
```

詳細については、デAWS WAFベロッパーガイドのSQL「インジェクションマッチ条件の使用」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateSqlInjectionMatchSet](#)」の「」を参照してください。

update-web-acl

次のコード例は、update-web-acl を使用する方法を示しています。

AWS CLI

ウェブを更新するには ACL

次のupdate-web-aclコマンドは、ウェブ内のActivatedRuleオブジェクトを削除します ACL。

```
aws waf update-web-acl --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90  
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates  
ActionDELETE「Action」, ActivatedRule='{Priority=1, RuleIdTypesWAFRule-1-  
Example「Action={TypeALLOW」},TypeREGULAR」
```

出力:

```
{  
  "ChangeToken": "12cs345-67cd-890b-1cd2-c3a4567d89f1"
```

```
}
```

詳細については、AWS WAF AWS「Firewall Manager」および AWS「Shield Advanced Developer Guide」の「[Web の使用ACLs](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateWebAcl](#)」の「」を参照してください。

update-xss-match-set

次の例は、update-xss-match-set を使用する方法を説明しています。

AWS CLI

を更新するには XssMatchSet

次のupdate-xss-match-setコマンドは、内の XssMatchTuple オブジェクト (フィルター) を削除します XssMatchSet。

```
aws waf update-xss-match-set --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_D
```

詳細については、AWS WAFデベロッパーガイドの「クロスサイトスクリプティング一致条件の使用」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateXssMatchSet](#)」の「」を参照してください。

AWS WAF Classic Regional を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS WAF Classic Regional。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-web-acl

次のコード例は、`associate-web-acl` を使用する方法を示しています。

AWS CLI

ウェブをリソースACLに関連付けるには

次の`associate-web-acl`コマンドはACL、で指定されたウェブ を `web-acl-id`、`resource-arn` で指定されたリソースに関連付けます。リソースは、アプリケーションロードバランサーまたはAPIゲートウェイを参照ARNできます。

```
aws waf-regional associate-web-acl \  
  --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --resource-arn 12cs345-67cd-890b-1cd2-c3a4567d89f1
```

詳細については、「デベロッパーガイド」の「[ウェブの使用ACLs](#)」を参照してください。AWS WAF

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateWebAcl](#)」の「」を参照してください。

put-logging-configuration

次の例は、`put-logging-configuration` を使用する方法を説明しています。

AWS CLI

指定された Kinesis Firehose ストリームACLARNを使用してウェブのログ記録設定を作成するには ARN

次の`put-logging-configuration`例は、リージョン の ALB/APIGateway WAFを持つ のログ記録設定を示していますus-east-1。

```
aws waf-regional put-logging-configuration \  
  --stream-arn arn:aws:kinesis:us-east-1:123456789012:stream/my-stream \  
  --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --resource-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancing/app/elb-123456789012
```

```
--logging-configuration ResourceArn=arn:aws:waf-  
regional:us-east-1:123456789012:webacl/3bffd3ed-fa2e-445e-869f-  
a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-  
east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[] \  
--region us-east-1
```

出力:

```
{  
  "LoggingConfiguration": {  
    "ResourceArn": "arn:aws:waf-regional:us-east-1:123456789012:webacl/3bffd3ed-  
fa2e-445e-869f-a6a7cf153fd3",  
    "LogDestinationConfigs": [  
      "arn:aws:firehose:us-east-1:123456789012:deliverystream/aws-waf-logs-  
firehose-stream"  
    ]  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLoggingConfiguration](#)」の「」を参照してください。

update-byte-match-set

次の例は、update-byte-match-set を使用する方法を説明しています。

AWS CLI

バイト一致セットを更新するには

次のupdate-byte-match-setコマンドは、内のByteMatchTupleオブジェクト (フィルター) を削除しますByteMatchSet。updates 値には二重引用符が埋め込まれているため、値を単一引用符で囲む必要があります。

```
aws waf-regional update-byte-match-set \  
--byte-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
--updates  
'Action="DELETE",ByteMatchTuple={FieldToMatch={Type="HEADER",Data="referer"},TargetString="'
```

詳細については、「AWS WAFデベロッパーガイド」の[「文字列一致条件の使用」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateByteMatchSet](#)」の「」を参照してください。

update-ip-set

次の例は、update-ip-set を使用する方法を説明しています。

AWS CLI

IP セットを更新するには

次のupdate-ip-setコマンドは、IPv4 アドレスIPSetを使用して を更新し、IPv6アドレスを削除します。get-change-token コマンドchange-tokenを実行して、 の値を取得します。更新の値には二重引用符が埋め込まれているため、値を単一引用符で囲む必要があります。

```
aws waf update-ip-set \  
  --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
  --updates  
'Action="INSERT", IPSetDescriptor={Type="IPV4", Value="12.34.56.78/16"}, Action="DELETE", IPSet
```

または、JSON ファイルを使用して入力を指定することもできます。例:

```
aws waf-regional update-ip-set \  
  --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
  --updates file://change.json
```

の内容 change.json

```
[  
  {  
    "Action": "INSERT",  
    "IPSetDescriptor":  
    {  
      "Type": "IPV4",  
      "Value": "12.34.56.78/16"  
    }  
  }  
]
```

```

    },
    {
      "Action": "DELETE",
      "IPSetDescriptor":
      {
        "Type": "IPV6",
        "Value": "1111:0000:0000:0000:0000:0000:0000:0111/128"
      }
    }
  ]

```

詳細については、AWS WAF「デベロッパーガイド」の「[IP マッチ条件の使用](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateIpSet](#)」の「」を参照してください。

update-rule

次の例は、update-rule を使用する方法を説明しています。

AWS CLI

ルールを更新するには

次のupdate-ruleコマンドは、ルール内のPredicateオブジェクトを削除します。updates 値には二重引用符が埋め込まれているため、値全体を単一引用符で囲む必要があります。

```

aws waf-regional update-rule \
  --rule-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
  --updates
'Action="DELETE",Predicate={Negated=false,Type="ByteMatch",DataId="MyByteMatchSetID"}'

```

詳細については、AWS WAF「デベロッパーガイド」の「[ルールの使用](#)」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateRule](#)」の「」を参照してください。

update-size-constraint-set

次のコード例は、update-size-constraint-set を使用する方法を示しています。

AWS CLI

サイズ制約セットを更新するには

次のupdate-size-constraint-setコマンドは、サイズ制約セット内の SizeConstraint オブジェクト (フィルター) を削除します。updates 値には二重引用符が埋め込まれているため、値全体を単一引用符で囲む必要があります。

```
aws waf-regional update-size-constraint-set \  
  --size-constraint-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
  --updates  
  'Action="DELETE",SizeConstraint={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NON'
```

詳細については、「AWS WAFデベロッパーガイド」の[「サイズ制約条件の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSizeConstraintSet](#)」の「」を参照してください。

update-sql-injection-match-set

次の例は、update-sql-injection-match-set を使用する方法を説明しています。

AWS CLI

SQLインジェクションマッチセットを更新するには

次のupdate-sql-injection-match-setコマンドは、SQLインジェクション一致セット内の SqlInjectionMatchTuple オブジェクト (フィルター) を削除します。updates 値には二重引用符が埋め込まれているため、値全体を一重引用符で囲む必要があります。:

```
aws waf-regional update-sql-injection-match-set -sql-injection-match-  
set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token  
12cs345-67cd-890b-1cd2-c3a4567d89f1 ---updates 'ActionDELETE「アクシヨ  
ン」, SqlInjectionMatchTuple={FieldToMatch={TypeQUERY「_STRING」,  
TextTransformationURLDECODE」
```

詳細については、AWS WAF 「デベロッパーガイド」の[SQL「インジェクションマッチ条件の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateSqlInjectionMatchSet](#)」の「」を参照してください。

update-web-acl

次の例は、update-web-acl を使用方法を説明しています。

AWS CLI

ウェブを更新するには ACL

次のupdate-web-aclコマンドは、ウェブ内のActivatedRuleオブジェクトを削除します ACL。updates 値には二重引用符が埋め込まれているため、値全体を一重引用符で囲む必要があります。

```
aws waf-regional update-web-acl \  
  --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
  --updates Action="DELETE",ActivatedRule='{Priority=1,RuleId="WAFRule-1-  
Example",Action={Type="ALLOW"},Type="ALLOW"}'
```

詳細については、「デベロッパーガイド」の「[ウェブの使用ACLs](#)」を参照してください。AWS WAF

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateWebAcl](#)」の「」を参照してください。

update-xss-match-set

次のコード例は、update-xss-match-set を使用方法を示しています。

AWS CLI

を更新するには XSSMatchSet

次のupdate-xss-match-setコマンドは、内のXssMatchTupleオブジェクト (フィルター) を削除しますXssMatchSet。updates 値には二重引用符が埋め込まれているため、値全体を一重引用符で囲む必要があります。

```
aws waf-regional update-xss-match-set \  
  --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --updates Action="DELETE",XssMatchTuple='{Priority=1,RuleId="WAFRule-1-  
Example",Action={Type="ALLOW"},Type="ALLOW"}'
```



```
--change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
--updates
'Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_'
```

詳細については、「AWS WAFデベロッパーガイド」の[「クロスサイトスクリプティング一致条件の使用」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateXssMatchSet](#)」の「」を参照してください。

AWS WAFV2 を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS WAFV2。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-web-acl

次のコード例は、associate-web-acl を使用する方法を示しています。

AWS CLI

ウェブをリージョン AWS リソースACLに関連付けるには

次のassociate-web-acl例では、指定されたウェブを Application Load Balancer ACLに関連付けます。

```
aws wafv2 associate-web-acl \
--web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
```

```
--resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/  
app/waf-cli-alb/1ea17125f8b25a2a \  
--region us-west-2
```

このコマンドでは何も出力されません。

詳細については、[AWS WAF AWS「Firewall Manager AWS」および「Shield Advanced Developer Guide」のAWS「リソースACLとウェブの関連付けまたは関連付け解除」](#)を参照してください。

- API 詳細については、AWS CLI「[コマンドリファレンスAssociateWebAcl](#)」の「」を参照してください。

check-capacity

次の例は、check-capacity を使用する方法を説明しています。

AWS CLI

ルールのセットで使用される容量を取得するには

以下は、レートベースのルールステートメントを含むルールセットと、ネストされたANDルールを含むルールステートメントの容量要件check-capacityを取得します。

```
aws wafv2 check-capacity \  
  --scope REGIONAL \  
  --rules file://waf-rule-list.json \  
  --region us-west-2
```

file://waf-rule-list.json の内容 :

```
[  
  {  
    "Name":"basic-rule",  
    "Priority":0,  
    "Statement":{  
      "AndStatement":{  
        "Statements":[  
          {  
            "ByteMatchStatement":{  
              "SearchString":"example.com",  
              "FieldToMatch":{
```

```
        "SingleHeader":{
            "Name":"host"
        }
    },
    "TextTransformations":[
        {
            "Priority":0,
            "Type":"LOWERCASE"
        }
    ],
    "PositionalConstraint":"EXACTLY"
}
},
{
    "GeoMatchStatement":{
        "CountryCodes":[
            "US",
            "IN"
        ]
    }
}
]
}
},
"Action":{
    "Allow":{

    }
},
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"basic-rule"
}
},
{
    "Name":"rate-rule",
    "Priority":1,
    "Statement":{
        "RateBasedStatement":{
            "Limit":1000,
            "AggregateKeyType":"IP"
        }
    }
},
```

```
    "Action":{
      "Block":{

      }
    },
    "VisibilityConfig":{
      "SampledRequestsEnabled":true,
      "CloudWatchMetricsEnabled":true,
      "MetricName":"rate-rule"
    }
  }
]
```

出力:

```
{
  "Capacity":15
}
```

詳細については、「[Firewall Manager、Shield Advanced デベロッパーガイド](#)」の[AWS WAF「ウェブACLキャパシティユニット \(WCU\)」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスCheckCapacity](#)」の「」を参照してください。

create-ip-set

次の例は、create-ip-set を使用する方法を説明しています。

AWS CLI

ウェブグループACLsとルールグループで使用する IP セットを作成するには

次のcreate-ip-setコマンドは、単一のアドレス範囲仕様を持つ IP セットを作成します。

```
aws wafv2 create-ip-set \  
  --name testip \  
  --scope REGIONAL \  
  --ip-address-version IPV4 \  
  --addresses 198.51.100.0/16
```

出力:

```
{
  "Summary":{
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Description":"",
    "Name":"testip",
    "LockToken":"447e55ac-0000-0000-0000-86b67c17f8b5",
    "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}
```

詳細については、[「Firewall Manager」および「Shield Advanced Developer Guide」の「IP セットと正規表現パターンセット」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateIpSet](#)」の「」を参照してください。

create-regex-pattern-set

次の例は、create-regex-pattern-set を使用する方法を説明しています。

AWS CLI

ウェブグループACLsとルールグループで使用する正規表現パターンセットを作成するには

次のcreate-regex-pattern-setコマンドは、2つの正規表現パターンを指定して正規表現パターンセットを作成します。

```
aws wafv2 create-regex-pattern-set \
  --name regexPatterSet01 \
  --scope REGIONAL \
  --description 'Test web-acl' \
  --regular-expression-list '["RegexString": "[0-9]*"/],{"RegexString": "[a-z]*"/}]'
```

出力:

```
{
  "Summary":{
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Description":"Test web-acl",
```

```

    "Name": "regexPatterSet01",
    "LockToken": "0bc01e21-03c9-4b98-9433-6229cbf1ef1c",
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}

```

詳細については、[「Firewall Manager, Shield Advanced デベロッパーガイド」の「IP セットと正規表現パターンセット」](#)を参照してください。 AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRegexPatternSet](#)」の「」を参照してください。

create-rule-group

次のコード例は、create-rule-group を使用する方法を示しています。

AWS CLI

ウェブで使用するカスタムルールグループを作成するには ACLs

次のcreate-rule-groupコマンドは、リージョンで使用するカスタムルールグループを作成します。グループのルールステートメントは、JSON形式のファイルで提供されます。

```

aws wafv2 create-rule-group \
  --name "TestRuleGroup" \
  --scope REGIONAL \
  --capacity 250 \
  --rules file://waf-rule.json \
  --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestRuleGroupMet
\
  --region us-west-2

```

file://waf-rule.json の内容 :

```

[
  {
    "Name": "basic-rule",
    "Priority": 0,
    "Statement": {
      "AndStatement": {
        "Statements": [
          {

```

```
        "ByteMatchStatement":{
            "SearchString":"example.com",
            "FieldToMatch":{
                "SingleHeader":{
                    "Name":"host"
                }
            },
            "TextTransformations":[
                {
                    "Priority":0,
                    "Type":"LOWERCASE"
                }
            ],
            "PositionalConstraint":"EXACTLY"
        },
        {
            "GeoMatchStatement":{
                "CountryCodes":[
                    "US",
                    "IN"
                ]
            }
        }
    ]
},
"Action":{
    "Allow":{

    }
},
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"basic-rule"
}
}
```

出力:

```
{
```

```

    "Summary":{
      "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Description":"","
      "Name":"TestRuleGroup",
      "LockToken":"7b3bcec2-374e-4c5a-b2b9-563bf47249f0",
      "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
  }
}

```

詳細については、AWS WAF AWS 「、ファイアウォールマネージャー、AWS Shield Advanced デベロッパーガイド」の「[独自のルールグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateRuleGroup](#)」の「」を参照してください。

create-web-acl

次のコード例は、create-web-acl を使用する方法を示しています。

AWS CLI

ウェブを作成するには ACL

次のcreate-web-aclコマンドは、ACLリージョンで使用するウェブを作成します。ウェブのルールステートメントACLは、JSON形式のファイルで提供されます。

```

aws wafv2 create-web-acl \
  --name TestWebAcl \
  --scope REGIONAL \
  --default-action Allow={} \
  --visibility-
config SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestWebAclMetric
\
  --rules file://waf-rule.json \
  --region us-west-2

```

file://waf-rule.json の内容 :

```

[
  {
    "Name":"basic-rule",
    "Priority":0,

```



```
"Statement":{
  "AndStatement":{
    "Statements":[
      {
        "ByteMatchStatement":{
          "SearchString":"example.com",
          "FieldToMatch":{
            "SingleHeader":{
              "Name":"host"
            }
          },
          "TextTransformations":[
            {
              "Priority":0,
              "Type":"LOWERCASE"
            }
          ],
          "PositionalConstraint":"EXACTLY"
        },
        {
          "GeoMatchStatement":{
            "CountryCodes":[
              "US",
              "IN"
            ]
          }
        }
      ]
    },
    "Action":{
      "Allow":{
      }
    },
    "VisibilityConfig":{
      "SampledRequestsEnabled":true,
      "CloudWatchMetricsEnabled":true,
      "MetricName":"basic-rule"
    }
  }
}
```

出力:

```
{
  "Summary":{
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebAcl/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Description":"",
    "Name":"TestWebAcl",
    "LockToken":"2294b3a1-eb60-4aa0-a86f-a3ae04329de9",
    "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  }
}
```

詳細については、[「Firewall Manager」](#) および [「Shield Advanced デベロッパーガイド」](#) の [「ウェブアクセスコントロールリスト \(ウェブ ACL\) の管理と使用」](#) を参照してください。

AWS WAF AWS AWS

- API 詳細については、AWS CLI [「コマンドリファレンス CreateWebAcl」](#) の [「」](#) を参照してください。

delete-ip-set

次の例は、delete-ip-set を使用する方法を説明しています。

AWS CLI

IP セットを削除するには

以下では、指定された IP セット delete-ip-set を削除します。この呼び出しには、呼び出しから取得できる ID、list-ip-sets、および呼び出しから取得できるロックトークン、list-ip-sets および `get-ip-set` が必要です。

```
aws wafv2 delete-ip-set \
  --name test1 \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --lock-token 46851772-db6f-459d-9385-49428812e357
```

このコマンドでは何も出力されません。

詳細については、[「Firewall Manager」](#) および [「Shield Advanced Developer Guide」](#) の [「IP セットと正規表現パターンセット」](#) を参照してください。 AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteIpSet](#)」の「」を参照してください。

delete-logging-configuration

次の例は、delete-logging-configuration を使用する方法を説明しています。

AWS CLI

ウェブのログ記録を無効にするには ACL

以下は、指定されたウェブ からログ記録設定delete-logging-configurationを削除します ACL。

```
aws wafv2 delete-logging-configuration \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

このコマンドでは何も出力されません。

詳細については、「[Firewall Manager、Shield Advanced デベロッパーガイド](#)」の「[ウェブACL トラフィック情報のログ記録](#)」を参照してください。 AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLoggingConfiguration](#)」の「」を参照してください。

delete-regex-pattern-set

次のコード例は、delete-regex-pattern-set を使用する方法を示しています。

AWS CLI

正規表現パターンセットを削除するには

以下は、指定された正規表現パターンセットの設定delete-regex-pattern-setを更新します。この呼び出しには、呼び出しから取得できる ID、list-regex-pattern-sets、および呼び出しlist-regex-pattern-setsまたは呼び出しから取得できるロックトークンが必要ですget-regex-pattern-set。

```
aws wafv2 delete-regex-pattern-set \  
  --name regexPatterSet01 \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

```
--scope REGIONAL \  
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
--lock-token 0bc01e21-03c9-4b98-9433-6229cbf1ef1c
```

このコマンドでは何も出力されません。

詳細については、「[Firewall Manager、Shield Advanced デベロッパーガイド](#)」の「[IP セットと正規表現パターンセット](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteRegexPatternSet](#)」の「」を参照してください。

delete-rule-group

次のコード例は、delete-rule-group を使用する方法を示しています。

AWS CLI

カスタムルールグループを削除するには

以下では、指定されたカスタムルールグループdelete-rule-groupを削除します。この呼び出しには、呼び出しから取得できる ID、list-rule-groups、および呼び出しlist-rule-groupsまたは呼び出しから取得できるロックトークンが必要ですget-rule-group。

```
aws wafv2 delete-rule-group \  
--name TestRuleGroup \  
--scope REGIONAL \  
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
--lock-token 7b3bcec2-0000-0000-0000-563bf47249f0
```

このコマンドでは何も出力されません。

詳細については、AWS WAF AWS 「[ファイアウォールマネージャー、AWS Shield Advanced デベロッパーガイド](#)」の「[独自のルールグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteRuleGroup](#)」の「」を参照してください。

delete-web-acl

次のコード例は、delete-web-acl を使用する方法を示しています。

AWS CLI

ウェブを削除するには ACL

以下では、アカウントACLから指定されたウェブdelete-web-aclを削除します。ウェブは、リソースに関連付けられていない場合にのみ削除ACLできます。この呼び出しには、呼び出しから取得できる ID、list-web-acls、および呼び出しlist-web-aclsまたは呼び出しから取得できるロックトークンが必要ですget-web-acl。

```
aws wafv2 delete-web-acl \  
  --name test \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token ebab4ed2-155e-4c9a-9efb-e4c45665b1f5
```

このコマンドでは何も出力されません。

詳細については、「Firewall Manager」および「Shield Advanced デベロッパーガイド」の「[ウェブアクセスコントロールリスト \(ウェブ ACL\) の管理と使用](#)」を参照してください。

AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteWebAcl](#)」の「」を参照してください。

describe-managed-rule-group

次のコード例は、describe-managed-rule-group を使用する方法を示しています。

AWS CLI

マネージドルールグループの説明を取得するには

以下は、AWS マネージドルールグループの説明describe-managed-rule-groupを取得します。

```
aws wafv2 describe-managed-rule-group \  
  --vendor-name AWS \  
  --name AWSManagedRulesCommonRuleSet \  
  --scope REGIONAL
```

出力:

```
{
  "Capacity": 700,
  "Rules": [
    {
      "Name": "NoUserAgent_HEADER",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "UserAgent_BadBots_HEADER",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "SizeRestrictions_QUERYSTRING",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "SizeRestrictions_Cookie_HEADER",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "SizeRestrictions_BODY",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "SizeRestrictions_URI_PATH",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "EC2MetaDataSSRF_BODY",
      "Action": {
        "Block": {}
      }
    }
  ]
}
```

```
    }
  },
  {
    "Name": "EC2MetaDataSSRF_COOKIE",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2MetaDataSSRF_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2MetaDataSSRF_QUERY_ARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericLFI_QUERY_ARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericLFI_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericLFI_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "RestrictedExtensions_URI_PATH",
    "Action": {
      "Block": {}
    }
  }
}
```

```
  },
  {
    "Name": "RestrictedExtensions_QUERYARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericRFI_QUERYARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericRFI_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericRFI_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_COOKIE",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_QUERYARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
```



```

        "Name": "CrossSiteScripting_URI_PATH",
        "Action": {
            "Block": {}
        }
    ]
}

```

詳細については、「[ファイアウォールマネージャー、Shield Advanced デベロッパーガイド](#)」の「[マネージドルールグループ](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeManagedRuleGroup](#)」の「」を参照してください。

disassociate-web-acl

次のコード例は、disassociate-web-acl を使用する方法を示しています。

AWS CLI

リージョン AWS リソースACLからウェブの関連付けを解除するには

次のdisassociate-web-acl例では、指定された Application Load Balancer から既存のウェブ ACL関連付けを削除します。

```

aws wafv2 disassociate-web-acl \
  --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a \
  --region us-west-2

```

このコマンドでは何も出力されません。

詳細については、[AWS WAF AWS 「Firewall Manager AWS 」および「Shield Advanced Developer Guide 」の「ウェブACLと AWS リソースの関連付けまたは関連付け解除」](#)を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDisassociateWebAcl](#)」の「」を参照してください。

get-ip-set

次のコード例は、get-ip-set を使用する方法を示しています。

AWS CLI

特定の IP セットを取得するには

以下は、指定された名前、スコープ、および ID を持つ IP セット `get-ip-set` を取得します。コマンド `create-ip-set` と から IP セットの ID を取得できます `list-ip-sets`。

```
aws wafv2 get-ip-set \  
  --name testip \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "IPSet":{  
    "Description":"","  
    "Name":"testip",  
    "IPAddressVersion":"IPV4",  
    "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",  
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",  
    "Addresses":[  
      "192.0.2.0/16"  
    ]  
  },  
  "LockToken":"447e55ac-2396-4c6d-b9f9-86b67c17f8b5"  
}
```

詳細については、[「、Firewall Manager、Shield Advanced デベロッパーガイド」の「IP セットと正規表現パターンセット」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [GetIpSet](#)」の「」を参照してください。

get-logging-configuration

次の例は、`get-logging-configuration` を使用する方法を説明しています。

AWS CLI

ウェブのログ記録設定を取得するには ACL

以下は、指定されたウェブ のログ記録設定 `get-logging-configuration` を取得します ACL。

```
aws wafv2 get-logging-configuration \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \  
  --region us-west-2
```

出力:

```
{  
  "LoggingConfiguration":{  
    "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "RedactedFields":[  
      {  
        "Method":{  
          }  
      }  
    ],  
    "LogDestinationConfigs":[  
      "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-  
custom-transformation"  
    ]  
  }  
}
```

詳細については、「[ファイアウォールマネージャー、Shield Advanced デベロッパーガイド](#)」の「[ウェブ ACL トラフィック情報のログ記録](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「[コマンドリファレンス `GetLoggingConfiguration`](#)」の「」を参照してください。

get-rate-based-statement-managed-keys

次の例は、`get-rate-based-statement-managed-keys` を使用する方法を説明しています。

AWS CLI

レートベースのルールによってブロックされている IP アドレスのリストを取得するには

以下では、リージョンアプリケーションに使用されているレートベースのルールによって現在ブロックされている IP アドレス `get-rate-based-statement-managed-keys` を取得します。

```
aws wafv2 get-rate-based-statement-managed-keys \  
  --scope REGIONAL \  
  --web-acl-name testwebacl2 \  
  --web-acl-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --rule-name ratebasedtest
```

出力:

```
{  
  "ManagedKeysIPV4":{  
    "IPAddressVersion":"IPV4",  
    "Addresses":[  
      "198.51.100.0/32"  
    ]  
  },  
  "ManagedKeysIPV6":{  
    "IPAddressVersion":"IPV6",  
    "Addresses":[]  
  }  
}
```

詳細については、「[Firewall Manager、Shield Advanced デベロッパーガイド](#)」の「[レートベースのルールステートメント](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「[コマンドリファレンスGetRateBasedStatementManagedKeys](#)」の「」を参照してください。

get-regex-pattern-set

次のコード例は、`get-regex-pattern-set` を使用する方法を示しています。

AWS CLI

特定の正規表現パターンセットを取得するには

以下は、指定された名前、スコープ、リージョン、および ID を持つ正規表現パターンセット `get-regex-pattern-set` を取得します。コマンド `create-regex-pattern-set` と `list-regex-pattern-sets` から正規表現パターンセットの ID を取得できます。

```
aws wafv2 get-regex-pattern-set \  
  --name regexPatterSet01 \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --region us-west-2
```

出力:

```
{  
  "RegexPatternSet": {  
    "Description": "Test web-acl",  
    "RegularExpressionList": [  
      {  
        "RegexString": "/[0-9]*/"  
      },  
      {  
        "RegexString": "/[a-z]*/"  
      }  
    ],  
    "Name": "regexPatterSet01",  
    "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/  
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  },  
  "LockToken": "c8abf33f-b6fc-46ae-846e-42f994d57b29"  
}
```

詳細については、[「Firewall Manager、Shield Advanced デベロッパーガイド」の「IP セットと正規表現パターンセット」](#)を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [GetRegexPatternSet](#)」の「」を参照してください。

get-rule-group

次のコード例は、`get-rule-group` を使用する方法を示しています。

AWS CLI

特定のカスタムルールグループを取得するには

次に、指定された名前、スコープ、および ID を持つカスタムルールグループ `get-rule-group` を取得します。コマンド `create-rule-group` からルールグループの ID を取得できま
す `list-rule-groups`。

```
aws wafv2 get-rule-group \  
  --name ff \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "RuleGroup":{  
    "Capacity":1,  
    "Description":"","  
    "Rules":[  
      {  
        "Priority":0,  
        "Action":{  
          "Block":{  
  
          }  
        },  
        "VisibilityConfig":{  
          "SampledRequestsEnabled":true,  
          "CloudWatchMetricsEnabled":true,  
          "MetricName":"jj"  
        },  
        "Name":"jj",  
        "Statement":{  
          "SizeConstraintStatement":{  
            "ComparisonOperator":"LE",  
            "TextTransformations":[  
              {  
                "Priority":0,  
                "Type":"NONE"  
              }  
            ],  
            "FieldToMatch":{
```

```

        "UriPath":{
            }
        },
        "Size":7
    }
}
],
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"ff"
},
"Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/ff/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"Name":"ff"
},
"LockToken":"485458c9-1830-4234-af31-ec4d52ced1b3"
}

```

詳細については、AWS WAF AWS 「[ファイアウォールマネージャー](#)、AWS Shield Advanced デベロッパーガイド」の「[独自のルールグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetRuleGroup](#)」の「」を参照してください。

get-sampled-requests

次の例は、get-sampled-requests を使用する方法を説明しています。

AWS CLI

ウェブのウェブリクエストのサンプルを取得するには ACL

以下は、指定されたウェブ、ルールメトリクスACL、および時間枠のサンプルされたウェブリクエストget-sampled-requestsを取得します。

```

aws wafv2 get-sampled-requests \
  --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --rule-metric-name AWS-AWSManagedRulesSQLiRuleSet \

```

```
--scope=REGIONAL \  
--time-window StartTime=2020-02-12T20:00Z,EndTime=2020-02-12T21:10Z \  
--max-items 100
```

出力:

```
{  
  "TimeWindow": {  
    "EndTime": 1581541800.0,  
    "StartTime": 1581537600.0  
  },  
  "SampledRequests": [  
    {  
      "Action": "BLOCK",  
      "Timestamp": 1581541799.564,  
      "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",  
      "Request": {  
        "Country": "US",  
        "URI": "/",  
        "Headers": [  
          {  
            "Name": "Host",  
            "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"  
          },  
          {  
            "Name": "Content-Length",  
            "Value": "7456"  
          },  
          {  
            "Name": "User-Agent",  
            "Value": "curl/7.53.1"  
          },  
          {  
            "Name": "Accept",  
            "Value": "/"  
          },  
          {  
            "Name": "Content-Type",  
            "Value": "application/x-www-form-urlencoded"  
          }  
        ],  
        "ClientIP": "198.51.100.08",  
        "Method": "POST",
```



```
        "HTTPVersion": "HTTP/1.1"
      },
      "Weight": 1
    },
    {
      "Action": "BLOCK",
      "Timestamp": 1581541799.988,
      "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
      "Request": {
        "Country": "US",
        "URI": "/",
        "Headers": [
          {
            "Name": "Host",
            "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
          },
          {
            "Name": "Content-Length",
            "Value": "7456"
          },
          {
            "Name": "User-Agent",
            "Value": "curl/7.53.1"
          },
          {
            "Name": "Accept",
            "Value": "/"
          },
          {
            "Name": "Content-Type",
            "Value": "application/x-www-form-urlencoded"
          }
        ],
        "ClientIP": "198.51.100.08",
        "Method": "POST",
        "HTTPVersion": "HTTP/1.1"
      },
      "Weight": 3
    },
    {
      "Action": "BLOCK",
      "Timestamp": 1581541799.846,
      "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
      "Request": {
```

```
    "Country": "US",
    "URI": "/",
    "Headers": [
      {
        "Name": "Host",
        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
      },
      {
        "Name": "Content-Length",
        "Value": "7456"
      },
      {
        "Name": "User-Agent",
        "Value": "curl/7.53.1"
      },
      {
        "Name": "Accept",
        "Value": "/"
      },
      {
        "Name": "Content-Type",
        "Value": "application/x-www-form-urlencoded"
      }
    ],
    "ClientIP": "198.51.100.08",
    "Method": "POST",
    "HTTPVersion": "HTTP/1.1"
  },
  "Weight": 1
},
{
  "Action": "BLOCK",
  "Timestamp": 1581541799.4,
  "RuleNameWithinRuleGroup": "AWS#AWSManagedRulesSQLiRuleSet#SQLi_BODY",
  "Request": {
    "Country": "US",
    "URI": "/",
    "Headers": [
      {
        "Name": "Host",
        "Value": "alb-test-1EXAMPLE1.us-east-1.elb.amazonaws.com"
      },
      {
        "Name": "Content-Length",
```

```
        "Value": "7456"
      },
      {
        "Name": "User-Agent",
        "Value": "curl/7.53.1"
      },
      {
        "Name": "Accept",
        "Value": "/"
      },
      {
        "Name": "Content-Type",
        "Value": "application/x-www-form-urlencoded"
      }
    ],
    "ClientIP": "198.51.100.08",
    "Method": "POST",
    "HTTPVersion": "HTTP/1.1"
  },
  "Weight": 1
}
],
"PopulationSize": 4
}
```

詳細については、AWS WAF AWS 「、Firewall Manager、AWS Shield Advanced デベロッパ—ガイド」の「[ウェブリクエストのサンプルの表示](#)」を参照してください。

- API 詳細については、「コマンドリファレンス[GetSampledRequests](#)」の「」を参照してください。AWS CLI

get-web-acl-for-resource

次のコード例は、get-web-acl-for-resource を使用する方法を示しています。

AWS CLI

AWS リソースACLに関連付けられているウェブを取得するには

次に、指定されたリソースACLに関連付けられているウェブJSONの get-web-acl-for-resource を取得します。

```
aws wafv2 get-web-acl-for-resource \
```

```
--resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a
```

出力:

```
{
  "WebACL":{
    "Capacity":3,
    "Description":"",
    "Rules":[
      {
        "Priority":1,
        "Action":{
          "Block":{

          }
        },
        "VisibilityConfig":{
          "SampledRequestsEnabled":true,
          "CloudWatchMetricsEnabled":true,
          "MetricName":"testrule01"
        },
        "Name":"testrule01",
        "Statement":{
          "AndStatement":{
            "Statements":[
              {
                "ByteMatchStatement":{
                  "PositionalConstraint":"EXACTLY",
                  "TextTransformations":[
                    {
                      "Priority":0,
                      "Type":"NONE"
                    }
                  ],
                  "SearchString":"dGVzdHN0cm1uZw==",
                  "FieldToMatch":{
                    "UriPath":{

                    }
                  }
                }
              }
            ]
          }
        }
      }
    ]
  },
}
```

```

        {
            "SizeConstraintStatement":{
                "ComparisonOperator":"EQ",
                "TextTransformations":[
                    {
                        "Priority":0,
                        "Type":"NONE"
                    }
                ],
                "FieldToMatch":{
                    "QueryString":{

                    }
                },
                "Size":0
            }
        }
    ]
}
},
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"test01"
},
"DefaultAction":{
    "Allow":{

    }
},
"Id":"9a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ",
"ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  ",
"Name":"test01"
}
}

```

詳細については、[AWS WAF AWS 「Firewall Manager AWS 」および「Shield Advanced Developer Guide」の「ウェブACLと AWS リソースの関連付けまたは関連付け解除」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWebAclForResource](#)」の「」を参照してください。

get-web-acl

次のコード例は、get-web-acl を使用する方法を示しています。

AWS CLI

ウェブを取得するには ACL

以下は、指定された名前、スコープ、および ID ACLを持つウェブget-web-aclを取得します。ウェブの ID は、コマンドcreate-web-aclと ACLから取得できますlist-web-acls。

```
aws wafv2 get-web-acl \  
  --name test01 \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "WebACL":{  
    "Capacity":3,  
    "Description":"","  
    "Rules":[  
      {  
        "Priority":1,  
        "Action":{  
          "Block":{  
  
          }  
        },  
        "VisibilityConfig":{  
          "SampledRequestsEnabled":true,  
          "CloudWatchMetricsEnabled":true,  
          "MetricName":"testrule01"  
        },  
        "Name":"testrule01",  
        "Statement":{  
          "AndStatement":{  
            "Statements":[
```

```

        {
            "ByteMatchStatement":{
                "PositionalConstraint":"EXACTLY",
                "TextTransformations":[
                    {
                        "Priority":0,
                        "Type":"NONE"
                    }
                ],
                "SearchString":"dGVzdHN0cmlyZw==",
                "FieldToMatch":{
                    "UriPath":{

                    }
                }
            }
        },
        {
            "SizeConstraintStatement":{
                "ComparisonOperator":"EQ",
                "TextTransformations":[
                    {
                        "Priority":0,
                        "Type":"NONE"
                    }
                ],
                "FieldToMatch":{
                    "QueryString":{

                    }
                }
            },
            "Size":0
        }
    ]
}
},
"VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"test01"
},

```

```
    "DefaultAction":{
      "Allow":{

      }
    },
    "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name":"test01"
  },
  "LockToken":"e3db7e2c-d58b-4ee6-8346-6aec5511c6fb"
}
```

詳細については、「Firewall Manager」および「Shield Advanced Developer Guide」の「[ウェブアクセスコントロールリスト \(ウェブ ACL\) の管理と使用](#)」を参照してください。AWS WAF
AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetWebAcl](#)」の「」を参照してください。

list-available-managed-rule-groups

次のコード例は、list-available-managed-rule-groups を使用する方法を示しています。

AWS CLI

マネージドルールグループを取得するには

以下は、ウェブ で現在使用可能なすべてのマネージドルールグループのリストlist-available-managed-rule-groupsを返しますACLs。

```
aws wafv2 list-available-managed-rule-groups \
  --scope REGIONAL
```

出力:

```
{
  "ManagedRuleGroups": [
    {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesCommonRuleSet",
```



```
    "Description": "Contains rules that are generally applicable to web
applications. This provides protection against exploitation of a wide range of
vulnerabilities, including those described in OWASP publications and common Common
Vulnerabilities and Exposures (CVE).",
  },
  {
    "VendorName": "AWS",
    "Name": "AWSManagedRulesAdminProtectionRuleSet",
    "Description": "Contains rules that allow you to block external access
to exposed admin pages. This may be useful if you are running third-party software
or would like to reduce the risk of a malicious actor gaining administrative access
to your application.",
  },
  {
    "VendorName": "AWS",
    "Name": "AWSManagedRulesKnownBadInputsRuleSet",
    "Description": "Contains rules that allow you to block request patterns
that are known to be invalid and are associated with exploitation or discovery of
vulnerabilities. This can help reduce the risk of a malicious actor discovering a
vulnerable application.",
  },
  {
    "VendorName": "AWS",
    "Name": "AWSManagedRulesSQLiRuleSet",
    "Description": "Contains rules that allow you to block request patterns
associated with exploitation of SQL databases, like SQL injection attacks. This can
help prevent remote injection of unauthorized queries.",
  },
  {
    "VendorName": "AWS",
    "Name": "AWSManagedRulesLinuxRuleSet",
    "Description": "Contains rules that block request patterns associated
with exploitation of vulnerabilities specific to Linux, including LFI attacks. This
can help prevent attacks that expose file contents or execute code for which the
attacker should not have had access.",
  },
  {
    "VendorName": "AWS",
    "Name": "AWSManagedRulesUnixRuleSet",
    "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to POSIX/POSIX-like OS, including LFI
attacks. This can help prevent attacks that expose file contents or execute code
for which access should not been allowed."
  },
},
```

```
{
  "VendorName": "AWS",
  "Name": "AWSManagedRulesWindowsRuleSet",
  "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to Windows, (e.g., PowerShell commands).
This can help prevent exploits that allow attacker to run unauthorized commands or
execute malicious code."
},
{
  "VendorName": "AWS",
  "Name": "AWSManagedRulesPHPRuleSet",
  "Description": "Contains rules that block request patterns associated
with exploiting vulnerabilities specific to the use of the PHP, including injection
of unsafe PHP functions. This can help prevent exploits that allow an attacker to
remotely execute code or commands."
},
{
  "VendorName": "AWS",
  "Name": "AWSManagedRulesWordPressRuleSet",
  "Description": "The WordPress Applications group contains rules that
block request patterns associated with the exploitation of vulnerabilities specific
to WordPress sites."
},
{
  "VendorName": "AWS",
  "Name": "AWSManagedRulesAmazonIpReputationList",
  "Description": "This group contains rules that are based on Amazon
threat intelligence. This is useful if you would like to block sources associated
with bots or other threats."
}
]
```

詳細については、「Firewall Manager」および「Shield Advanced Developer Guide」の「[Managed Rule Groups](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAvailableManagedRuleGroups](#)」の「」を参照してください。

list-ip-sets

次の例は、list-ip-sets を使用する方法を説明しています。

AWS CLI

IP セットのリストを取得するには

以下は、リージョンスコープを持つアカウントのすべての IP セット `list-ip-sets` を取得します。

```
aws wafv2 list-ip-sets \  
  --scope REGIONAL
```

出力:

```
{  
  "IPSets": [  
    {  
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Description": "",  
      "Name": "testip",  
      "LockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc",  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111  "  
    }  
  ],  
  "NextMarker": "testip"  
}
```

詳細については、[「Firewall Manager」](#) および [「Shield Advanced Developer Guide」](#) の [「IP セットと正規表現パターンセット」](#) を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス [ListIpSets](#)」の「」を参照してください。

list-logging-configurations

次のコード例は、`list-logging-configurations` を使用する方法を示しています。

AWS CLI

リージョンのすべてのログ記録設定のリストを取得するには

以下では、`us-west-2`リージョンでリージョンでの使用が対象とACLsされているウェブのすべてのログ記録設定 `list-logging-configurations` を取得します。

```
aws wafv2 list-logging-configurations \  
  --scope REGIONAL \  
  --region us-west-2
```

出力:

```
{  
  "LoggingConfigurations": [  
    {  
      "ResourceArn": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
test-2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "RedactedFields": [  
        {  
          "QueryString": {  
            }  
        }  
      ],  
      "LogDestinationConfigs": [  
        "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-  
logs-test"  
      ]  
    },  
    {  
      "ResourceArn": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
test/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "RedactedFields": [  
        {  
          "Method": {  
            }  
        }  
      ],  
      "LogDestinationConfigs": [  
        "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-  
logs-custom-transformation"  
      ]  
    }  
  ]  
}
```

詳細については、AWS WAF AWS 「、ファイアウォールマネージャー、AWS Shield Advanced デベロッパーガイド」の「[ウェブACLトラフィック情報のログ記録](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListLoggingConfigurations](#)」の「」を参照してください。

list-regex-pattern-sets

次の例は、list-regex-pattern-sets を使用する方法を説明しています。

AWS CLI

正規表現パターンセットのリストを取得するには

以下は、リージョン で定義されているアカウントのすべての正規表現パターンセットlist-regex-pattern-setsを取得しますus-west-2。

```
aws wafv2 list-regex-pattern-sets \  
--scope REGIONAL \  
--region us-west-2
```

出力:

```
{  
  "NextMarker": "regexPatterSet01",  
  "RegexPatternSets": [  
    {  
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/regexpatternset/  
regexPatterSet01/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Description": "Test web-acl",  
      "Name": "regexPatterSet01",  
      "LockToken": "f17743f7-0000-0000-0000-19a8b93bfb01",  
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
    }  
  ]  
}
```

詳細については、「[Firewall Manager、Shield Advanced デベロッパーガイド](#)」の「[IP セットと正規表現パターンセット](#)」を参照してください。 AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRegexPatternSets](#)」の「」を参照してください。

list-resources-for-web-acl

次の例は、list-resources-for-web-acl を使用する方法を説明しています。

AWS CLI

ウェブに関連付けられたリソースを取得するには ACL

以下では、リージョン ACL 内の指定されたウェブに現在関連付けられている API Gateway RESTAPI リソース list-resources-for-web-acl を取得します us-west-2。

```
aws wafv2 list-resources-for-web-acl \
  --web-acl-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/TestWebACL/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --resource-type API_GATEWAY \
  --region us-west-2
```

出力:

```
{
  "ResourceArns": [
    "arn:aws:apigateway:us-west-2::/restapis/EXAMPLE111/stages/testing"
  ]
}
```

詳細については、AWS WAF AWS 「Firewall Manager AWS 」および「Shield Advanced Developer Guide」の「Web ACLと AWS リソースの関連付けまたは関連付け解除」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [ListResourcesForWebAcl](#)」の「」を参照してください。

list-rule-groups

次の例は、list-rule-groups を使用する方法を説明しています。

AWS CLI

カスタムルールグループのリストを取得するには

以下では、指定されたスコープとリージョンの場所のアカウントに定義されているすべてのカスタムルールグループ list-rule-groups を取得します。

```
aws wafv2 list-rule-groups \  
  --scope REGIONAL \  
  --region us-west-2
```

出力:

```
{  
  "RuleGroups":[  
    {  
      "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/  
TestRuleGroup/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "Description":"","  
      "Name":"TestRuleGroup",  
      "LockToken":"1eb5ec48-0000-0000-0000-ee9b906c541e",  
      "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
    },  
    {  
      "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/rulegroup/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
      "Description":"","  
      "Name":"test",  
      "LockToken":"b0f4583e-998b-4880-9069-3fbe45738b43",  
      "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"  
    }  
  ],  
  "NextMarker":"test"  
}
```

詳細については、AWS WAF AWS 「、ファイアウォールマネージャー、AWS Shield Advanced デベロッパーガイド」の「[独自のルールグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListRuleGroups](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのすべてのタグ AWS WAF を取得するには

以下は、指定されたウェブのすべてのタグキーと値のペアのリスト `list-tags-for-resource` を取得します ACL。

```
aws wafv2 list-tags-for-resource \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

出力:

```
{  
  "NextMarker": "",  
  "TagInfoForResource": {  
    "ResourceARN": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
testwebacl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "TagList": [  
  
    ]  
  }  
}
```

詳細については、[AWS WAF AWS「Firewall Manager AWS」および「Shield Advanced デベロッパーガイド」の「の開始 AWS WAF方法」](#)を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス [ListTagsForResource](#)」の「」を参照してください。

list-web-acls

次のコード例は、`list-web-acls` を使用する方法を示しています。

AWS CLI

ACLs スコープのウェブを取得するには

以下では、指定されたスコープのアカウントに ACLs 定義されているすべてのウェブ `list-web-acls` を取得します。

```
aws wafv2 list-web-acls \  
  --scope REGIONAL
```

出力:


```
{
  "NextMarker": "Testt",
  "WebACLs": [
    {
      "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/Testt/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Description": "sssss",
      "Name": "Testt",
      "LockToken": "7f36cb30-74ef-4cff-8cd4-a77e1aba1746",
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
  ]
}
```

詳細については、「Firewall Manager」および「Shield Advanced デベロッパーガイド」の「[ウェブアクセスコントロールリスト \(ウェブ ACL\) の管理と使用](#)」を参照してください。

AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[ListWebAcls](#)」の「」を参照してください。

put-logging-configuration

次の例は、put-logging-configuration を使用する方法を説明しています。

AWS CLI

ウェブにログ記録設定を追加するには ACL

以下は、Amazon Kinesis Data Firehose ログ記録設定aws-waf-logs-custom-transformationを指定されたウェブ put-logging-configurationに追加します。ログから編集されたフィールドACLはありません。

```
aws wafv2 put-logging-configuration \
  --logging-configuration ResourceArn=arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111,LogDestinationConfigs=arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-custom-transformation \
  --region us-west-2
```

出力:

```
{
  "LoggingConfiguration":{
    "ResourceArn":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test-
cli/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "LogDestinationConfigs":[
      "arn:aws:firehose:us-west-2:123456789012:deliverystream/aws-waf-logs-
custom-transformation"
    ]
  }
}
```

詳細については、「[AWS WAF AWS CLI 「PutLoggingConfiguration」の「」を参照してください。](#)、ファイアウォールマネージャー、Shield Advanced デベロッパガイド」の「[ウェブACLトラフィック情報のログ記録](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[PutLoggingConfiguration](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースに AWS WAF タグを追加するには

次のtag-resource例では、`Key`のキーNameと `Value` に設定された値のタグAWSWAFを指定されたウェブに追加しますACL。

```
aws wafv2 tag-resource \
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --tags Key=Name, Value=AWSWAF
```

このコマンドでは何も出力されません。

詳細については、[AWS WAF AWS CLI 「Firewall Manager AWS」および「Shield Advanced デベロッパガイド」の「の開始 AWS WAF方法」を参照してください。](#)

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次のコード例は、untag-resource を使用する方法を示しています。

AWS CLI

リソースから AWS WAF タグを削除するには

次のuntag-resource例では、指定されたウェブ KeyNameから キーを持つ タグを削除します ACL。

```
aws wafv2 untag-resource \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tag-keys "KeyName"
```

このコマンドでは何も出力されません。

詳細については、AWS WAF AWS 「Firewall Manager AWS 」および「[Shield Advanced デベロッパーガイド](#)」の「[の開始 AWS WAF方法](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスUntagResource](#)」の「」を参照してください。

update-ip-set

次のコード例は、update-ip-set を使用する方法を示しています。

AWS CLI

既存の IP セットの設定を変更するには

以下は、指定された IP セットの設定update-ip-setを更新します。この呼び出しには、呼び出しから取得できる ID、list-ip-sets、呼び出しから取得できるロックトークン、list-ip-setsおよび get-ip-set。この呼び出しは、後続の更新に使用できるロックトークンも返します。

```
aws wafv2 update-ip-set \  
  --name testip \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token lock-token
```

```
--addresses 198.51.100.0/16 \  
--lock-token 447e55ac-2396-4c6d-b9f9-86b67c17f8b5
```

出力:

```
{  
  "NextLockToken": "0674c84b-0304-47fe-8728-c6bfff46af8fc"  
}
```

詳細については、[「ファイアウォールマネージャー、Shield Advanced デベロッパーガイド」](#)の「[IP セットと正規表現パターンセット](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateIpSet](#)」の「」を参照してください。

update-regex-pattern-set

次の例は、update-regex-pattern-set を使用する方法を説明しています。

AWS CLI

既存の正規表現パターンセットの設定を変更するには

以下は、指定された正規表現パターンセットの設定update-regex-pattern-setを更新します。この呼び出しには、呼び出しから取得できる ID、list-regex-pattern-sets、呼び出しから取得できるロックトークン、list-regex-pattern-setsおよび get-regex-pattern-set。この呼び出しは、後続の更新に使用できるロックトークンも返します。

```
aws wafv2 update-regex-pattern-set \  
  --name ExampleRegex \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --regular-expression-list RegexString="^.+ $" \  
  --lock-token ed207e9c-82e9-4a77-aadd-81e6173ab7eb
```

出力:

```
{  
  "NextLockToken": "12ebc73e-fa68-417d-a9b8-2bdd761a4fa5"  
}
```

詳細については、「[Firewall Manager、Shield Advanced デベロッパーガイド](#)」の「[IP セットと正規表現パターンセット](#)」を参照してください。AWS WAF AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRegexPatternSet](#)」の「」を参照してください。

update-rule-group

次の例は、update-rule-group を使用する方法を説明しています。

AWS CLI

カスタムルールグループを更新するには

以下はupdate-rule-group、既存のカスタムルールグループの可視性設定を変更します。この呼び出しには、呼び出しから取得できる ID、list-rule-groups、呼び出しから取得できるロックトークン、list-rule-groupsおよびが必要でget-rule-group。この呼び出しは、後続の更新に使用できるロックトークンも返します。

```
aws wafv2 update-rule-group \  
  --name TestRuleGroup \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0 \  
  --visibility-  
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=TestMetricsFor  
 \  
  --region us-west-2
```

出力:

```
{  
  "NextLockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e"  
}
```

詳細については、AWS WAF AWS 「[ファイアウォールマネージャー、AWS Shield Advanced デベロッパーガイド](#)」の「[独自のルールグループの管理](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateRuleGroup](#)」の「」を参照してください。

update-web-acl

次の例は、update-web-acl を使用する方法を説明しています。

AWS CLI

ウェブを更新するには ACL

以下は、既存のウェブ の設定update-web-aclを変更しますACL。この呼び出しには、呼び出しから取得できる ID、list-web-acls、および呼び出しから取得できるロックトークンやその他の設定が必要ですget-web-acl。この呼び出しは、後続の更新に使用できるロックトークンも返します。

```
aws wafv2 update-web-acl \  
  --name TestWebAcl \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token 2294b3a1-0000-0000-0000-a3ae04329de9 \  
  --default-action Block={} \  
  --visibility-  
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=NewMetricTestw  
 \  
  --rules file://waf-rule.json \  
  --region us-west-2
```

出力:

```
{  
  "NextLockToken": "714a0cfb-0000-0000-0000-2959c8b9a684"  
}
```

詳細については、「Firewall Manager」および「Shield Advanced Developer Guide」の「[ウェブアクセスコントロールリスト \(ウェブ ACL\) の管理と使用](#)」を参照してください。AWS WAF

AWS AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateWebAcl](#)」の「」を参照してください。

を使用した Amazon WorkDocs の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています WorkDocs。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

abort-document-version-upload

次のコード例は、abort-document-version-upload を使用する方法を示しています。

AWS CLI

ドキュメントバージョンのアップロードを停止するには

この例では、以前に開始されたドキュメントバージョンのアップロードを停止します。

コマンド:

```
aws workdocs abort-document-version-upload --document-id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2 --version-id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dccccd417da9313
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AbortDocumentVersionUpload](#)」の「」を参照してください。

activate-user

次の例は、activate-user を使用する方法を説明しています。

AWS CLI

ユーザーをアクティブ化するには

この例では、非アクティブなユーザーをアクティブ化します。

コマンド:

```
aws workdocs activate-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

出力:

```
{
  "User": {
    "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Username": "exampleUser",
    "EmailAddress": "exampleUser@site.awsapps.com",
    "GivenName": "Example",
    "Surname": "User",
    "OrganizationId": "d-926726012c",
    "RootFolderId":
"75f67c183aa1217409ac87576a45c03a5df5e6d8c51c35c01669970538e86cd0",
    "RecycleBinFolderId":
"642b7dd3e60b14204534f3df7b1959e01b5d170f8c2707f410e40a8149120a57",
    "Status": "ACTIVE",
    "Type": "MINIMALUSER",
    "CreatedTimestamp": 1521226107.747,
    "ModifiedTimestamp": 1525297406.462,
    "Storage": {
      "StorageUtilizedInBytes": 0,
      "StorageRule": {
        "StorageAllocatedInBytes": 0,
        "StorageType": "QUOTA"
      }
    }
  }
}
```


- API 詳細については、AWS CLI 「コマンドリファレンス[ActivateUser](#)」の「」を参照してください。

add-resource-permissions

次のコード例は、add-resource-permissions を使用する方法を示しています。

AWS CLI

リソースのアクセス許可を追加するには

この例では、指定されたプリンシパルのリソースにアクセス許可を追加します。

コマンド:

```
aws workdocs add-resource-permissions --resource-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --principals Id=anonymous, Type=ANONYMOUS, Role=VIEWER
```

出力:

```
{
  "ShareResults": [
    {
      "PrincipalId": "anonymous",
      "Role": "VIEWER",
      "Status": "SUCCESS",
      "ShareId":
        "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "StatusMessage": ""
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[AddResourcePermissions](#)」の「」を参照してください。

create-comment

次のコード例は、create-comment を使用する方法を示しています。

AWS CLI

新しいコメントを追加するには

この例では、指定されたドキュメントバージョンに新しいコメントを追加します。

コマンド:

```
aws workdocs create-comment --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --text "This is a comment."
```

出力:

```
{
  "Comment": {
    "CommentId": "1534799058197-c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
    "ThreadId": "1534799058197-c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
    "Text": "This is a comment.",
    "Contributor": {
      "Id": "arn:aws:iam::123456789123:user/exampleUser",
      "Username": "exampleUser",
      "GivenName": "Example",
      "Surname": "User",
      "Status": "ACTIVE"
    },
    "CreatedTimestamp": 1534799058.197,
    "Status": "PUBLISHED",
    "Visibility": "PUBLIC"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateComment](#)」の「」を参照してください。

create-custom-metadata

次のコード例は、create-custom-metadata を使用する方法を示しています。

AWS CLI

カスタムメタデータを作成するには

この例では、指定されたドキュメントのカスタムメタデータを作成します。

コマンド:

```
aws workdocs create-custom-metadata --resource-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-metadata KeyName1=example,KeyName2=example2
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateCustomMetadata](#)」の「」を参照してください。

create-folder

次のコード例は、create-folder を使用する方法を示しています。

AWS CLI

フォルダを作成するには

この例では、フォルダを作成します。

コマンド:

```
aws workdocs create-folder --name documents --parent-folder-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

出力:

```
{
  "Metadata": {
    "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
    "Name": "documents",
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
```

```
    "ParentFolderId":  
    "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",  
    "CreatedTimestamp": 1534450467.622,  
    "ModifiedTimestamp": 1534450467.622,  
    "ResourceState": "ACTIVE",  
    "Signature": "",  
    "Size": 0,  
    "LatestVersionSize": 0  
  }  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateFolder](#)」の「」を参照してください。

create-labels

次のコード例は、create-labels を使用する方法を示しています。

AWS CLI

ラベルを作成するには

この例では、ドキュメントの一連のラベルを作成します。

コマンド:

```
aws workdocs create-labels --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --  
labels "documents" "examples" "my_documents"
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateLabels](#)」の「」を参照してください。

create-notification-subscription

次のコード例は、create-notification-subscription を使用する方法を示しています。

AWS CLI

通知サブスクリプションを作成するには

次のcreate-notification-subscription例では、指定された Amazon WorkDocs 組織の通知サブスクリプションを設定します。

```
aws workdocs create-notification-subscription \  
  --organization-id d-123456789c \  
  --protocol HTTPS \  
  --subscription-type ALL \  
  --notification-endpoint "https://example.com/example"
```

出力:

```
{  
  "Subscription": {  
    "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",  
    "EndPoint": "https://example.com/example",  
    "Protocol": "HTTPS"  
  }  
}
```

詳細については、「[Amazon WorkDocs デベロッパーガイド](#)」の「[通知をサブスクライブする](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスCreateNotificationSubscription](#)」の「」を参照してください。

create-user

次のコード例は、create-user を使用する方法を示しています。

AWS CLI

新しいユーザーを作成するには

この例では、Simple AD または Microsoft AD ディレクトリに新しいユーザーを作成します。

コマンド:

```
aws workdocs create-user --organization-id d-926726012c --username exampleUser2
--email-address exampleUser2@site.awsapps.com --given-name example2Name --
surname example2Surname --password examplePa$$w0rd
```

出力:

```
{
  "User": {
    "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Username": "exampleUser2",
    "EmailAddress": "exampleUser2@site.awsapps.com",
    "GivenName": "example2Name",
    "Surname": "example2Surname",
    "OrganizationId": "d-926726012c",
    "RootFolderId":
"35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
    "RecycleBinFolderId":
"9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
    "Status": "ACTIVE",
    "Type": "MINIMALUSER",
    "CreatedTimestamp": 1535478836.584,
    "ModifiedTimestamp": 1535478836.584,
    "Storage": {
      "StorageUtilizedInBytes": 0,
      "StorageRule": {
        "StorageAllocatedInBytes": 0,
        "StorageType": "QUOTA"
      }
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUser](#)」の「」を参照してください。

deactivate-user

次のコード例は、`deactivate-user` を使用する方法を示しています。

AWS CLI

ユーザーを非アクティブ化するには

この例では、アクティブなユーザーを非アクティブ化します。

コマンド:

```
aws workdocs deactivate-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeactivateUser](#)」の「」を参照してください。

delete-comment

次の例は、delete-comment を使用する方法を説明しています。

AWS CLI

ドキュメントバージョンから指定されたコメントを削除するには

この例では、指定されたドキュメントバージョンから指定されたコメントを削除します。

コマンド:

```
aws workdocs delete-comment --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --comment-id 1534799058197-c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteComment](#)」の「」を参照してください。

delete-custom-metadata

次の例は、delete-custom-metadata を使用する方法を説明しています。

AWS CLI

リソースからカスタムメタデータを削除するには

この例では、指定されたリソースからすべてのカスタムメタデータを削除します。

コマンド:

```
aws workdocs delete-custom-metadata --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteCustomMetadata](#)」の「」を参照してください。

delete-document

次のコード例は、delete-document を使用する方法を示しています。

AWS CLI

ドキュメントを削除するには

この例では、指定されたドキュメントを削除します。

コマンド:

```
aws workdocs delete-document --document-  
id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

出力:

```
None
```


- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteDocument](#)」の「」を参照してください。

delete-folder-contents

次のコード例は、delete-folder-contents を使用する方法を示しています。

AWS CLI

フォルダの内容を削除するには

この例では、指定されたフォルダの内容を削除します。

コマンド:

```
aws workdocs delete-folder-contents --folder-  
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFolderContents](#)」の「」を参照してください。

delete-folder

次のコード例は、delete-folder を使用する方法を示しています。

AWS CLI

フォルダを削除するには

この例では、指定されたフォルダを削除します。

コマンド:

```
aws workdocs delete-folder --folder-  
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

出力:

None

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteFolder](#)」の「」を参照してください。

delete-labels

次の例は、delete-labels を使用する方法を説明しています。

AWS CLI

ラベルを削除するには

この例では、ドキュメントから指定されたラベルを削除します。

コマンド:

```
aws workdocs delete-labels --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --  
labels "documents" "examples"
```

出力:

None

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteLabels](#)」の「」を参照してください。

delete-notification-subscription

次の例は、delete-notification-subscription を使用する方法を説明しています。

AWS CLI

通知サブスクリプションを削除するには

次のdelete-notification-subscription例では、指定された通知サブスクリプションを削除します。

```
aws workdocs delete-notification-subscription \  
--subscription-id 123ab4c5-678d-901e-f23g-45h6789j0123 \  

```

```
--organization-id d-123456789c
```

このコマンドでは何も出力されません。

詳細については、「[Amazon WorkDocs デベロッパーガイド](#)」の「[通知をサブスクライブする](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteNotificationSubscription](#)」の「」を参照してください。

delete-user

次の例は、delete-user を使用する方法を説明しています。

AWS CLI

ユーザーを削除するには

この例では、ユーザーを削除します。

コマンド:

```
aws workdocs delete-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

出力:

```
None
```

- API 詳細については、AWS CLI 「[コマンドリファレンスDeleteUser](#)」の「」を参照してください。

describe-activities

次の例は、describe-activities を使用する方法を説明しています。

AWS CLI

ユーザーアクティビティのリストを取得するには

この例では、指定された組織の最新のユーザーアクティビティのリストを返します。最新の2つのアクティビティには制限が設定されています。

コマンド:

```
aws workdocs describe-activities --organization-id d-926726012c --limit 2
```

出力:

```
{
  "UserActivities": [
    {
      "Type": "DOCUMENT_VERSION_DOWNLOADED",
      "TimeStamp": 1534800122.17,
      "Initiator": {
        "Id": "arn:aws:iam::123456789123:user/exampleUser"
      },
      "ResourceMetadata": {
        "Type": "document",
        "Name": "updatedDoc",
        "Id":
"15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
        "Owner": {
          "Id":
"S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
          "GivenName": "exampleName",
          "Surname": "exampleSurname"
        }
      }
    },
    {
      "Type": "DOCUMENT_VERSION_VIEWED",
      "TimeStamp": 1534799079.207,
      "Initiator": {
        "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
        "GivenName": "exampleName",
        "Surname": "exampleSurname"
      },
      "ResourceMetadata": {
        "Type": "document",
        "Name": "updatedDoc",
        "Id":
"15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3",
        "Owner": {
          "Id":
"S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
```

```

        "GivenName": "exampleName",
        "Surname": "exampleSurname"
    }
}
],
"Marker":
"DnF1ZXJ5VGhlbkZldGNoAgAAAAAAS7Fm1TaU10d1FTU1h1UU00VVFibD1RWhcAAAAAAAJTRY3bWh5eUgzaVF1ZX
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeActivities](#)」の「」を参照してください。

describe-comments

次の例は、describe-comments を使用する方法を説明しています。

AWS CLI

指定されたドキュメントバージョンのすべてのコメントを一覧表示するには

この例では、指定されたドキュメントバージョンのすべてのコメントを一覧表示します。

コマンド:

```

aws workdocs describe-comments --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920

```

出力:

```

{
  "Comments": [
    {
      "CommentId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "ThreadId": "1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5",
      "Text": "This is a comment.",
      "Contributor": {
        "Username": "arn:aws:iam::123456789123:user/exampleUser",
        "Type": "USER"
      }
    }
  ]
}

```

```
    },
    "CreatedTimestamp": 1534799058.197,
    "Status": "PUBLISHED",
    "Visibility": "PUBLIC"
  }
]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeComments](#)」の「」を参照してください。

describe-document-versions

次の例は、describe-document-versions を使用する方法を説明しています。

AWS CLI

ドキュメントのバージョンを取得するには

この例では、初期化されたバージョンやソースドキュメントの など、指定されたドキュメント URL のドキュメントバージョンを取得します。

コマンド:

```
aws workdocs describe-document-versions --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields SOURCE
```

出力:

```
{
  "DocumentVersions": [
    {
      "Id":
      "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
      "Name": "exampleDoc.docx",
      "ContentType": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
      "Size": 13922,
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Status": "ACTIVE",
      "CreatedTimestamp": 1534452029.587,
```

```

    "ModifiedTimestamp": 1534452029.849,
    "CreatorId":
    "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Source": {
        "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1534452029587-15e129dfc1875
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.docx&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
    }
  },
  {
    "Id": "1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59",
    "Name": "exampleDoc.pdf",
    "ContentType": "application/pdf",
    "Size": 425916,
    "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
    "Status": "ACTIVE",
    "CreatedTimestamp": 1529005196.082,
    "ModifiedTimestamp": 1529005196.796,
    "CreatorId":
    "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Source": {
        "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1529005196082-
bb75fa19abc287699cb07147f75816dce43a53a10f28dc001bf61ef2fab01c59?
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc29.pdf&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180816T204149Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180816%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
    }
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeDocumentVersions](#)」の「」を参照してください。

describe-folder-contents

次のコード例は、describe-folder-contents を使用する方法を示しています。

AWS CLI

フォルダの内容を説明するには

この例では、ドキュメントやサブフォルダなど、指定されたフォルダのすべてのアクティブなコンテンツを日付順に昇順でソートします。

コマンド:

```
aws workdocs describe-folder-contents --folder-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --sort DATE --order ASCENDING --type ALL
```

出力:

```
{
  "Folders": [
    {
      "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
      "Name": "testing",
      "CreatorId":
      "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
      "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "CreatedTimestamp": 1534450467.622,
      "ModifiedTimestamp": 1534451113.504,
      "ResourceState": "ACTIVE",
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Size": 23019,
      "LatestVersionSize": 11537
    }
  ],
  "Documents": [
    {
      "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "CreatorId":
      "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "ParentFolderId":
      "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
```



```

    "CreatedTimestamp": 1529005196.082,
    "ModifiedTimestamp": 1534452483.01,
    "LatestVersionMetadata": {
      "Id":
    "1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
      "Name": "exampleDoc.docx",
      "ContentType": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
      "Size": 13922,
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Status": "ACTIVE",
      "CreatedTimestamp": 1534452029.587,
      "ModifiedTimestamp": 1534452029.587,
      "CreatorId":
    "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
    },
    "ResourceState": "ACTIVE"
  }
]
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeFolderContents](#)」の「」を参照してください。

describe-groups

次のコード例は、describe-groups を使用する方法を示しています。

AWS CLI

グループのリストを取得するには

次のdescribe-groups例では、指定された Amazon WorkDocs 組織に関連付けられているグループを一覧表示します。

```

aws workdocs describe-groups \
  --search-query "e" \
  --organization-id d-123456789c

```

出力:

```
{
```

```
"Groups": [
  {
    "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444&d-123456789c",
    "Name": "Example Group 1"
  },
  {
    "Id": "S-1-1-11-1122222222-2222233333-3333334444-5555&d-123456789c",
    "Name": "Example Group 2"
  }
]
```

詳細については、[「Amazon 管理ガイド」の WorkDocs](#)「Amazon WorkDocs の開始方法」を参照してください。

- API 詳細については、AWS CLI「コマンドリファレンス[DescribeGroups](#)」の「」を参照してください。

describe-notification-subscriptions

次の例は、describe-notification-subscriptions を使用する方法を説明しています。

AWS CLI

通知サブスクリプションのリストを取得するには

次のdescribe-notification-subscriptions例では、指定された Amazon WorkDocs 組織の通知サブスクリプションを取得します。

```
aws workdocs describe-notification-subscriptions \
  --organization-id d-123456789c
```

出力:

```
{
  "Subscriptions": [
    {
      "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
      "EndPoint": "https://example.com/example",
      "Protocol": "HTTPS"
    }
  ]
}
```

```
}
```

詳細については、「[Amazon WorkDocs デベロッパーガイド](#)」の「[通知をサブスクライブする](#)」を参照してください。

- API 詳細については、AWS CLI 「[コマンドリファレンスDescribeNotificationSubscriptions](#)」の「」を参照してください。

describe-resource-permissions

次のコード例は、describe-resource-permissions を使用する方法を示しています。

AWS CLI

リソースのアクセス許可のリストを取得するには

この例では、指定されたリソース (ドキュメントまたはフォルダ) のアクセス許可のリストを返します。

コマンド:

```
aws workdocs describe-resource-permissions --resource-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3
```

出力:

```
{
  "Principals": [
    {
      "Id": "anonymous",
      "Type": "ANONYMOUS",
      "Roles": [
        {
          "Role": "VIEWER",
          "Type": "DIRECT"
        }
      ]
    },
    {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Type": "USER",
      "Roles": [
        {
```

```
        "Role": "OWNER",
        "Type": "DIRECT"
      }
    ]
  },
  {
    "Id": "d-926726012c",
    "Type": "ORGANIZATION",
    "Roles": [
      {
        "Role": "VIEWER",
        "Type": "INHERITED"
      }
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeResourcePermissions](#)」の「」を参照してください。

describe-users

次のコード例は、describe-users を使用する方法を示しています。

AWS CLI

指定されたユーザーの詳細を取得するには

この例では、指定された組織内のすべてのユーザーの詳細を取得します。

コマンド:

```
aws workdocs describe-users --organization-id d-926726012c
```

出力:

```
{
  "Users": [
    {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
      "Username": "example1User",
      "OrganizationId": "d-926726012c",
```

```
    "RootFolderId":
      "3c0e3f849dd20a9771d937b9bbcc97e18796150ae56c26d64a4fa0320a2dedc9",
      "RecycleBinFolderId":
        "c277f4c4d647be1f5147b3184ffa96e1e2bf708278b696cacba68ba13b91f4fe",
        "Status": "INACTIVE",
        "Type": "USER",
        "CreatedTimestamp": 1535478999.452,
        "ModifiedTimestamp": 1535478999.452
    },
    {
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-4444&d-926726012c",
      "Username": "example2User",
      "EmailAddress": "example2User@site.awsapps.com",
      "GivenName": "example2Name",
      "Surname": "example2Surname",
      "OrganizationId": "d-926726012c",
      "RootFolderId":
        "35b886cb17198cbd547655e58b025dff0cf34aaed638be52009567e23dc67390",
        "RecycleBinFolderId":
          "9858c3e9ed4c2460dde9aadb4c69fde998070dd46e5e985bd08ec6169ea249ff",
          "Status": "ACTIVE",
          "Type": "MINIMALUSER",
          "CreatedTimestamp": 1535478836.584,
          "ModifiedTimestamp": 1535478836.584
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUsers](#)」の「」を参照してください。

get-document-path

次のコード例は、get-document-path を使用する方法を示しています。

AWS CLI

ドキュメントのパス情報を取得するには

この例では、指定されたドキュメントのパス情報 (ルートフォルダからの階層) を取得し、親フォルダの名前を含めます。

コマンド:

```
aws workdocs get-document-path --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields NAME
```

出力:

```
{
  "Path": {
    "Components": [
      {
        "Id":
        "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
        "Name": "/"
      },
      {
        "Id":
        "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
        "Name": "Top Level Folder"
      },
      {
        "Id":
        "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
        "Name": "exampleDoc.docx"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDocumentPath](#)」の「」を参照してください。

get-document-version

次のコード例は、get-document-version を使用する方法を示しています。

AWS CLI

指定されたドキュメントのバージョンメタデータを取得するには

この例では、ソースメタデータURLやカスタムメタデータなど、指定されたドキュメントのバージョンメタデータを取得します。

コマンド:

```
aws workdocs get-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
fields SOURCE --include-custom-metadata
```

出力:

```
{
  "Metadata": {
    "Id":
    "1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920",
    "Name": "exampleDoc",
    "ContentType": "application/vnd.openxmlformats-
officedocument.wordprocessingml.document",
    "Size": 11537,
    "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
    "Status": "ACTIVE",
    "CreatedTimestamp": 1521672507.741,
    "ModifiedTimestamp": 1534451113.504,
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Source": {
      "ORIGINAL": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3/152167
response-content-disposition=attachment%3B%20filename%2A
%3DUTF-8%27%27exampleDoc&X-Amz-Algorithm=AWS1-ABCD-EFG234&X-Amz-
Date=20180820T212202Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180820%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDocumentVersion](#)」の「」を参照してください。

get-document

次のコード例は、get-document を使用する方法を示しています。

AWS CLI

ドキュメントの詳細を取得するには

この例では、指定されたドキュメントの詳細を取得します。

コマンド:

```
aws workdocs get-document --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65
```

出力:

```
{
  "Metadata": {
    "Id": "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "ParentFolderId":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
    "CreatedTimestamp": 1529005196.082,
    "ModifiedTimestamp": 1534452483.01,
    "LatestVersionMetadata": {
      "Id":
"1534452029587-15e129dfc187505c407588df255be83de2920d733859f1d2762411d22a83e3ef",
      "Name": "exampleDoc.docx",
      "ContentType": "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
      "Size": 13922,
      "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
      "Status": "ACTIVE",
      "CreatedTimestamp": 1534452029.587,
      "ModifiedTimestamp": 1534452029.587,
      "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
    },
    "ResourceState": "ACTIVE"
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetDocument](#)」の「」を参照してください。

get-folder-path

次の例は、get-folder-path を使用する方法を説明しています。

AWS CLI

フォルダのパス情報を取得するには

この例では、指定されたフォルダのパス情報 (ルートフォルダからの階層) を取得し、親フォルダの名前を含めます。

コマンド:

```
aws workdocs get-folder-path --folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --fields NAME
```

出力:

```
{
  "Path": {
    "Components": [
      {
        "Id":
        "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
        "Name": "/"
      },
      {
        "Id":
        "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
        "Name": "Top Level Folder"
      },
      {
        "Id":
        "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
        "Name": "Sublevel Folder"
      }
    ]
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFolderPath](#)」の「」を参照してください。

get-folder

次のコード例は、get-folder を使用する方法を示しています。

AWS CLI

フォルダのメタデータを取得するには

この例では、指定されたフォルダのメタデータを取得します。

コマンド:

```
aws workdocs get-folder --folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

出力:

```
{
  "Metadata": {
    "Id": "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
    "Name": "exampleFolder",
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "ParentFolderId":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
    "CreatedTimestamp": 1534450467.622,
    "ModifiedTimestamp": 1534451113.504,
    "ResourceState": "ACTIVE",
    "Signature": "1a23456b78901c23d4ef56gh7EXAMPLE",
    "Size": 23019,
    "LatestVersionSize": 11537
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetFolder](#)」の「」を参照してください。

get-resources

次の例は、get-resources を使用する方法を説明しています。

AWS CLI

共有リソースを取得するには

次のget-resources例では、指定された Amazon WorkDocs ユーザーと共有されているリソースを取得します。

```
aws workdocs get-resources \  
  --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333" \  
  --collection-type SHARED_WITH_ME
```

出力:

```
{  
  "Folders": [],  
  "Documents": []  
}
```

詳細については、「Amazon WorkDocs ユーザーガイド」の「[ファイルとフォルダの共有](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[GetResources](#)」の「」を参照してください。

initiate-document-version-upload

次の例は、initiate-document-version-upload を使用する方法を説明しています。

AWS CLI

ドキュメントバージョンのアップロードを開始するには

次のinitiate-document-upload例では、新しいドキュメントオブジェクトとバージョンオブジェクトを作成します。

```
aws workdocs initiate-document-version-upload \  
  --name exampledocname \  
  --parent-folder-  
id eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189
```

出力:

```
{  
  "Metadata": {  
    "Id": "feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2",  
    "CreatorId": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",  
    "ParentFolderId":  
    "eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189",
```

```

    "CreatedTimestamp": 1536773972.914,
    "ModifiedTimestamp": 1536773972.914,
    "LatestVersionMetadata": {
      "Id": "1536773972914-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccc417da9313",
      "Name": "exampledocname",
      "ContentType": "application/octet-stream",
      "Size": 0,
      "Status": "INITIALIZED",
      "CreatedTimestamp": 1536773972.914,
      "ModifiedTimestamp": 1536773972.914,
      "CreatorId": "arn:aws:iam::123456789123:user/EXAMPLE"
    },
    "ResourceState": "ACTIVE"
  },
  "UploadMetadata": {
    "UploadUrl": "https://gb-us-west-2-prod-doc-source.s3.us-
west-2.amazonaws.com/
feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2/1536773972914-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccc417da9313?X-Amz-
Algorithm=AWS1-ABCD-EFG234&X-Amz-Date=20180912T173932Z&X-Amz-SignedHeaders=content-
type%3Bhost%3Bx-amz-server-side-encryption&X-Amz-Expires=899&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20180912%2Fus-west-2%2Fs3%2Faws1_request&X-Amz-
Signature=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE",
    "SignedHeaders": {
      "Content-Type": "application/octet-stream",
      "x-amz-server-side-encryption": "ABC123"
    }
  }
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[InitiateDocumentVersionUpload](#)」の「」を参照してください。

remove-all-resource-permissions

次のコード例は、remove-all-resource-permissions を使用する方法を示しています。

AWS CLI

指定されたリソースからすべてのアクセス許可を削除するには

この例では、指定されたリソースからすべてのアクセス許可を削除します。

コマンド:

```
aws workdocs remove-all-resource-permissions --resource-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveAllResourcePermissions](#)」の「」を参照してください。

remove-resource-permission

次の例は、remove-resource-permission を使用する方法を説明しています。

AWS CLI

リソースからアクセス許可を削除するには

この例では、指定されたプリンシパルのリソースからアクセス許可を削除します。

コマンド:

```
aws workdocs remove-resource-permission --resource-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-id anonymous
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RemoveResourcePermission](#)」の「」を参照してください。

update-document-version

次のコード例は、update-document-version を使用する方法を示しています。

AWS CLI

ドキュメントのバージョンステータスをアクティブに変更するには

この例では、ドキュメントバージョンのステータスをアクティブに変更します。

コマンド:

```
aws workdocs update-document-version --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --version-status ACTIVE
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDocumentVersion](#)」の「」を参照してください。

update-document

次の例は、update-document を使用する方法を説明しています。

AWS CLI

ドキュメントを更新するには

この例では、ドキュメントの名前と親フォルダを更新します。

コマンド:

```
aws workdocs update-document --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --name updatedDoc --parent-folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateDocument](#)」の「」を参照してください。

update-folder

次の例は、update-folder を使用する方法を説明しています。

AWS CLI

フォルダを更新するには

この例では、フォルダの名前と親フォルダを更新します。

コマンド:

```
aws workdocs update-folder --folder-  
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --  
name exampleFolder1 --parent-folder-  
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

出力:

```
None
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateFolder](#)」の「」を参照してください。

update-user

次のコード例は、update-user を使用する方法を示しています。

AWS CLI

ユーザーを更新するには

この例では、指定されたユーザーのタイムゾーンを更新します。

コマンド:

```
aws workdocs update-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c" --time-zone-id "America/Los_Angeles"
```

出力:

```
{
  "User": {
    "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c",
    "Username": "exampleUser",
    "EmailAddress": "exampleUser@site.awsapps.com",
    "GivenName": "Example",
    "Surname": "User",
    "OrganizationId": "d-926726012c",
    "RootFolderId":
    "c5eceb5e1a2d1d460c9d1af8330ae117fc8d39bb1d3ed6acd0992d5ff192d986",
    "RecycleBinFolderId":
    "6ca20102926ad15f04b1d248d6d6e44f2449944eda5c758f9a1e9df6a6b7fa66",
    "Status": "ACTIVE",
    "Type": "USER",
    "TimeZoneId": "America/Los_Angeles",
    "Storage": {
      "StorageUtilizedInBytes": 0,
      "StorageRule": {
        "StorageAllocatedInBytes": 53687091200,
        "StorageType": "QUOTA"
      }
    }
  }
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateUser](#)」の「」を参照してください。

を使用した Amazon WorkMail の例 AWS CLI

次のコード例は、Amazon AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています WorkMail。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

associate-delegate-to-resource

次の例は、`associate-delegate-to-resource` を使用する方法を説明しています。

AWS CLI

リソースに委任を追加するには

次の`associate-delegate-to-resource` コマンドは、リソースに委任を追加します。

```
aws workmail associate-delegate-to-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateDelegateToResource](#)」の「」を参照してください。

associate-member-to-group

次のコード例は、`associate-member-to-group` を使用する方法を示しています。

AWS CLI

グループにメンバーを追加するには

次の`associate-member-to-group` コマンドは、指定されたメンバーをグループに追加します。

```
aws workmail associate-member-to-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[AssociateMemberToGroup](#)」の「」を参照してください。

create-alias

次の例は、create-alias を使用する方法を説明しています。

AWS CLI

エイリアスを作成するには

次のcreate-aliasコマンドは、指定されたエンティティ (ユーザーまたはグループ) のエイリアスを作成します。

```
aws workmail create-alias \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --alias exampleAlias@site.awsapps.com
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateAlias](#)」の「」を参照してください。

create-group

次の例は、create-group を使用する方法を説明しています。

AWS CLI

新しいグループを作成するには

次のcreate-groupコマンドは、指定された組織の新しいグループを作成します。

```
aws workmail create-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-name ExampleGroup \  
  --group-type Standard
```

```
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
--name exampleGroup1
```

出力:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateGroup](#)」の「」を参照してください。

create-resource

次のコード例は、create-resource を使用する方法を示しています。

AWS CLI

新しいリソースを作成するには

次のcreate-resourceコマンドは、指定された組織の新しいリソース (会議室) を作成します。

```
aws workmail create-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleRoom1 \  
  --type ROOM
```

出力:

```
{  
  "ResourceId": "r-7afe0efbade843a58cdc10251fce992c"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateResource](#)」の「」を参照してください。

create-user

次のコード例は、create-user を使用する方法を示しています。

AWS CLI

新しいユーザーを作成するには

次のcreate-userコマンドは、新しいユーザーを作成します。

```
aws workmail create-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleName \  
  --display-name exampleDisplayName \  
  --password examplePa$$w0rd
```

出力:

```
{  
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateUser](#)」の「」を参照してください。

delete-access-control-rule

次のコード例は、delete-access-control-rule を使用する方法を示しています。

AWS CLI

アクセスコントロールルールを削除するには

次のdelete-access-control-rule例では、指定された Amazon WorkMail 組織から指定されたアクセスコントロールルールを削除します。

```
aws workmail delete-access-control-rule \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --name "myRule"
```

このコマンドでは何も出力されません。

詳細については、「Amazon WorkMail 管理者ガイド」の「[アクセスコントロールルールの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAccessControlRule](#)」の「」を参照してください。

delete-alias

次のコード例は、delete-alias を使用する方法を示しています。

AWS CLI

エイリアスを削除するには

次のdelete-aliasコマンドは、指定されたエンティティ (ユーザーまたはグループ) のエイリアスを削除します。

```
aws workmail delete-alias \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --alias exampleAlias@site.awsapps.com
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteAlias](#)」の「」を参照してください。

delete-group

次のコード例は、delete-group を使用する方法を示しています。

AWS CLI

既存のグループを削除するには

次のdelete-groupコマンドは、Amazon から既存のグループを削除します WorkMail。

```
aws workmail delete-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteGroup](#)」の「」を参照してください。

delete-mailbox-permissions

次の例は、delete-mailbox-permissions を使用する方法を説明しています。

AWS CLI

メールボックスのアクセス許可を削除するには

次のdelete-mailbox-permissionsコマンドは、以前にユーザーまたはグループに付与されたメールボックスアクセス許可を削除します。エンティティはメールボックスを所有するユーザーを表し、被付与者はアクセス許可を削除するユーザーまたはグループを表します。

```
aws workmail delete-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --grantee-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteMailboxPermissions](#)」の「」を参照してください。

delete-resource

次の例は、delete-resource を使用する方法を説明しています。

AWS CLI

既存のリソースを削除するには

次のdelete-resourceコマンドは、Amazon から既存のリソースを削除します WorkMail。

```
aws workmail delete-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-7afe0efbade843a58cdc10251fce992c
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteResource](#)」の「」を参照してください。

delete-user

次の例は、delete-user を使用する方法を説明しています。

AWS CLI

ユーザーを削除するには

次のdelete-userコマンドは、Amazon WorkMail および後続のすべてのシステムから指定されたユーザーを削除します。

```
aws workmail delete-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUser](#)」の「」を参照してください。

deregister-from-work-mail

次の例は、deregister-from-work-mail を使用する方法を説明しています。

AWS CLI

既存のエンティティを無効にするには

次のderegister-from-work-mailコマンドは、既存のエンティティ (ユーザー、グループ、またはリソース) が Amazon を使用するのを無効にします WorkMail。

```
aws workmail deregister-from-work-mail \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterFromWorkMail](#)」の「」を参照してください。

describe-group

次のコード例は、describe-group を使用する方法を示しています。

AWS CLI

グループの情報を取得するには

次のdescribe-groupコマンドは、指定されたグループに関する情報を取得します。

```
aws workmail describe-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

出力:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444",  
  "Name": "exampleGroup1",  
  "State": "ENABLED"  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeGroup](#)」の「」を参照してください。

describe-organization

次のコード例は、describe-organization を使用する方法を示しています。

AWS CLI

組織の情報を取得するには

次のdescribe-organizationコマンドは、指定された Amazon WorkMail 組織の情報を取得します。

```
aws workmail describe-organization \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

出力:

```
{
```



```
"OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
"Alias": "alias",
"State": "Active",
"DirectoryId": "d-926726012c",
"DirectoryType": "VpcDirectory",
"DefaultMailDomain": "site.awsapps.com",
"CompletedDate": 1522693605.468,
"ARN": "arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza"
}
```

詳細については、「[Amazon WorkMail 管理者ガイド](#)」の「[組織の使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeOrganization](#)」の「」を参照してください。

describe-resource

次のコード例は、describe-resource を使用する方法を示しています。

AWS CLI

リソースの情報を取得するには

次のdescribe-resourceコマンドは、指定されたリソースに関する情報を取得します。

```
aws workmail describe-resource \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --resource-id r-7afe0efbade843a58cdc10251fce992c
```

出力:

```
{
  "ResourceId": "r-7afe0efbade843a58cdc10251fce992c",
  "Name": "exampleRoom1",
  "Type": "ROOM",
  "BookingOptions": {
    "AutoAcceptRequests": true,
    "AutoDeclineRecurringRequests": false,
    "AutoDeclineConflictingRequests": true
  },
  "State": "ENABLED"
```

```
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeResource](#)」の「」を参照してください。

describe-user

次の例は、describe-user を使用する方法を説明しています。

AWS CLI

ユーザー情報を取得するには

次のdescribe-userコマンドは、指定されたユーザーに関する情報を取得します。

```
aws workmail describe-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

出力:

```
{  
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
  "Name": "exampleUser1",  
  "Email": "exampleUser1@site.awsapps.com",  
  "DisplayName": "",  
  "State": "ENABLED",  
  "UserRole": "USER",  
  "EnabledDate": 1532459261.827  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeUser](#)」の「」を参照してください。

disassociate-delegate-from-resource

次のコード例は、disassociate-delegate-from-resource を使用する方法を示しています。

AWS CLI

リソースからメンバーを削除するには

次のdisassociate-delegate-from-resourceコマンドは、リソースから指定されたメンバーを削除します。

```
ws workmail disassociate-delegate-from-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateDelegateFromResource](#)」の「」を参照してください。

disassociate-member-from-group

次の例は、disassociate-member-from-group を使用する方法を説明しています。

AWS CLI

グループからメンバーを削除するには

次のdisassociate-member-from-groupコマンドは、指定されたメンバーをグループから削除します。

```
aws workmail disassociate-member-from-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[DisassociateMemberFromGroup](#)」の「」を参照してください。

get-access-control-effect

次の例は、get-access-control-effect を使用する方法を説明しています。

AWS CLI

アクセスコントロールルールの効果を得るには

次のget-access-control-effect例では、指定された IP アドレス、アクセスプロトコルアクション、およびユーザー ID に対する指定された Amazon WorkMail 組織のアクセスコントロールルールの効果を取得します。

```
aws workmail get-access-control-effect \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --ip-address "192.0.2.0" \  
  --action "WindowsOutlook" \  
  --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333"
```

出力:

```
{  
  "Effect": "DENY",  
  "MatchedRules": [  
    "myRule"  
  ]  
}
```

詳細については、「Amazon WorkMail 管理者ガイド」の [「アクセスコントロールルールの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetAccessControlEffect](#)」の「」を参照してください。

get-mailbox-details

次のコード例は、get-mailbox-details を使用する方法を示しています。

AWS CLI

ユーザーのメールボックスの詳細を取得するには

次のget-mailbox-detailsコマンドは、指定されたユーザーのメールボックスに関する詳細を取得します。

```
aws workmail get-mailbox-details \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

出力:

```
{
  "MailboxQuota": 51200,
  "MailboxSize": 0.03890800476074219
}
```

詳細については、「Amazon 管理者ガイド」の「[ユーザーアカウントの管理](#)」を参照してください。 WorkMail

- API 詳細については、AWS CLI 「コマンドリファレンス[GetMailboxDetails](#)」の「」を参照してください。

list-access-control-rules

次の例は、list-access-control-rules を使用する方法を説明しています。

AWS CLI

アクセスコントロールルールを一覧表示するには

次のlist-access-control-rules例では、指定された Amazon WorkMail 組織のアクセスコントロールルールを一覧表示します。

```
aws workmail list-access-control-rules \
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

出力:

```
{
  "Rules": [
    {
      "Name": "default",
      "Effect": "ALLOW",
      "Description": "Default WorkMail Rule",
      "DateCreated": 0.0,
      "DateModified": 0.0
    },
    {
      "Name": "myRule",
      "Effect": "DENY",
      "Description": "my rule",
      "UserIds": [
```

```
        "S-1-1-11-1111111111-2222222222-3333333333-3333"  
    ],  
    "DateCreated": 1581635628.0,  
    "DateModified": 1581635628.0  
  }  
]  
}
```

詳細については、「Amazon WorkMail 管理者ガイド」の「[アクセスコントロールルールの使用](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAccessControlRules](#)」の「」を参照してください。

list-aliases

次のコード例は、list-aliases を使用する方法を示しています。

AWS CLI

メンバーのエイリアスを一覧表示するには

次のlist-aliasesコマンドは、指定されたメンバー(ユーザーまたはグループ)のエイリアスを一覧表示します。

```
aws workmail list-aliases \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

出力:

```
{  
  "Aliases": [  
    "exampleAlias@site.awsapps.com",  
    "exampleAlias1@site.awsapps.com"  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListAliases](#)」の「」を参照してください。

list-group-members

次のコード例は、list-group-members を使用する方法を示しています。

AWS CLI

グループメンバーを一覧表示するには

次のlist-group-membersコマンドは、指定されたグループのメンバーを一覧表示します。

```
aws workmail list-group-members \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

出力:

```
{  
  "Members": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Name": "exampleUser1",  
      "Type": "USER",  
      "State": "ENABLED",  
      "EnabledDate": 1532459261.827  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGroupMembers](#)」の「」を参照してください。

list-groups

次の例は、list-groups を使用する方法を説明しています。

AWS CLI

グループのリストを取得するには

次のlist-groupsコマンドは、指定された組織内のグループの概要を取得します。

```
aws workmail list-groups \  

```

```
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

出力:

```
{
  "Groups": [
    {
      "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",
      "Name": "exampleGroup1",
      "State": "DISABLED"
    },
    {
      "Id": "S-4-4-44-1122222222-2222233333-3333334444-4444",
      "Name": "exampleGroup2",
      "State": "ENABLED"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListGroups](#)」の「」を参照してください。

list-mailbox-permissions

次のコード例は、list-mailbox-permissions を使用する方法を示しています。

AWS CLI

メールボックスのアクセス許可を取得するには

次のlist-mailbox-permissionsコマンドは、指定されたエンティティのメールボックスに関連付けられたメールボックスのアクセス許可を取得します。

```
aws workmail list-mailbox-permissions \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

出力:

```
{
```



```
"Permissions": [
  {
    "GranteeId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
    "GranteeType": "USER",
    "PermissionValues": [
      "FULL_ACCESS"
    ]
  }
]
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListMailboxPermissions](#)」の「」を参照してください。

list-organizations

次のコード例は、list-organizations を使用する方法を示しています。

AWS CLI

組織のリストを取得するには

次のlist-organizationsコマンドは、削除されていない組織の概要を取得します。

```
aws workmail list-organizations
```

出力:

```
{
  "OrganizationSummaries": [
    {
      "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
      "Alias": "exampleAlias",
      "State": "Active"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListOrganizations](#)」の「」を参照してください。

list-resource-delegates

次のコード例は、list-resource-delegates を使用する方法を示しています。

AWS CLI

リソースの委任を一覧表示するには

次のlist-resource-delegatesコマンドは、指定されたリソースに関連付けられている委任を取得します。

```
aws workmail list-resource-delegates \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443
```

出力:

```
{  
  "Delegates": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Type": "USER"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResourceDelegates](#)」の「」を参照してください。

list-resources

次の例は、list-resources を使用する方法を説明しています。

AWS CLI

リソースのリストを取得するには

次のlist-resourcesコマンドは、指定された組織のリソースの概要を取得します。

```
aws workmail list-resources \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

出力:

```
{
  "Resources": [
    {
      "Id": "r-7afe0efbade843a58cdc10251fce992c",
      "Name": "exampleRoom1",
      "Type": "ROOM",
      "State": "ENABLED"
    }
  ]
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListResources](#)」の「」を参照してください。

list-tags-for-resource

次の例は、list-tags-for-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを一覧表示するには

次のlist-tags-for-resource例では、指定された Amazon WorkMail 組織のタグを一覧表示します。

```
aws workmail list-tags-for-resource \
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-
  n1pq2345678r901st2u3vx45x6789yza
```

出力:

```
{
  "Tags": [
    {
      "Key": "priority",
      "Value": "1"
    }
  ]
}
```

詳細については、「Amazon WorkMail 管理者ガイド」の「[組織のタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTagsForResource](#)」の「」を参照してください。

list-users

次のコード例は、list-users を使用する方法を示しています。

AWS CLI

ユーザーのリストを取得するには

次のlist-usersコマンドは、指定された組織内のユーザーの概要を取得します。

```
aws workmail list-users \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

出力:

```
{  
  "Users": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Email": "exampleUser1@site.awsapps.com",  
      "Name": "exampleUser1",  
      "State": "ENABLED",  
      "UserRole": "USER",  
      "EnabledDate": 1532459261.827  
    },  
    {  
      "Id": "S-1-1-11-1122222222-2222233333-3333334444-4444",  
      "Name": "exampleGuestUser",  
      "State": "DISABLED",  
      "UserRole": "SYSTEM_USER"  
    }  
  ]  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListUsers](#)」の「」を参照してください。

put-access-control-rule

次のコード例は、put-access-control-rule を使用する方法を示しています。

AWS CLI

新しいアクセスコントロールルールを設定するには

次のput-access-control-rule例では、指定された Amazon WorkMail 組織への指定されたユーザーアクセスを拒否します。

```
aws workmail put-access-control-rule \  
  --name "myRule" \  
  --effect "DENY" \  
  --description "my rule" \  
  --user-ids "S-1-1-11-1111111111-2222222222-3333333333-3333" \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

このコマンドでは何も出力されません。

詳細については、「Amazon WorkMail 管理者ガイド」の [「アクセスコントロールルールの使用」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [PutAccessControlRule](#)」の「」を参照してください。

put-mailbox-permissions

次の例は、put-mailbox-permissions を使用する方法を説明しています。

AWS CLI

メールボックスのアクセス許可を設定するには

次のput-mailbox-permissionsコマンドは、指定された被付与者 (ユーザーまたはグループ) にフルアクセス許可を設定します。エンティティはメールボックスの所有者を表します。

```
aws workmail put-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --grantee-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

```
--permission-values FULL_ACCESS
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[PutMailboxPermissions](#)」の「」を参照してください。

register-to-work-mail

次のコード例は、register-to-work-mail を使用する方法を示しています。

AWS CLI

既存のエンティティまたは無効化されたエンティティを登録するには

次のregister-to-work-mailコマンドは、指定された既存のエンティティ (ユーザー、グループ、またはリソース) が Amazon を使用できるようにします WorkMail。

```
aws workmail register-to-work-mail \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --email exampleGroup1@site.awsapps.com
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterToWorkMail](#)」の「」を参照してください。

reset-password

次のコード例は、reset-password を使用する方法を示しています。

AWS CLI

ユーザーのパスワードをリセットするには

次のreset-passwordコマンドは、指定されたユーザーのパスワードをリセットします。

```
aws workmail reset-password \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --password ExamplePassword123!
```

```
--password examplePa$$w0rd
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[ResetPassword](#)」の「」を参照してください。

tag-resource

次のコード例は、tag-resource を使用する方法を示しています。

AWS CLI

リソースにタグを適用するには

次のtag-resource例では、キー「priority」と値「1」のタグを指定された Amazon WorkMail 組織に適用します。

```
aws workmail tag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tags "Key=priority, Value=1"
```

このコマンドでは何も出力されません。

詳細については、「Amazon WorkMail 管理者ガイド」の「[組織のタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[TagResource](#)」の「」を参照してください。

untag-resource

次の例は、untag-resource を使用する方法を説明しています。

AWS CLI

リソースのタグを解除するには

次のuntag-resource例では、指定された Amazon WorkMail 組織から指定されたタグを削除します。

```
aws workmail untag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tag-keys "priority"
```

このコマンドでは何も出力されません。

詳細については、「Amazon WorkMail 管理者ガイド」の「[組織のタグ付け](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[UntagResource](#)」の「」を参照してください。

update-mailbox-quota

次のコード例は、update-mailbox-quota を使用する方法を示しています。

AWS CLI

ユーザーのメールボックスクォータを更新するには

次のupdate-mailbox-quotaコマンドは、指定されたユーザーのメールボックスクォータを変更します。

```
aws workmail update-mailbox-quota \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --mailbox-quota 40000
```

このコマンドでは何も出力されません。

詳細については、「Amazon 管理者ガイド」の「[ユーザーアカウントの管理](#)」を参照してください。 WorkMail

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateMailboxQuota](#)」の「」を参照してください。

update-primary-email-address

次のコード例は、update-primary-email-address を使用する方法を示しています。

AWS CLI

プライマリ E メールアドレスを更新するには

次のupdate-primary-email-addressコマンドは、指定されたエンティティ (ユーザー、グループ、またはリソース) のプライマリ E メールアドレスを更新します。

```
aws workmail update-primary-email-address \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --email exampleUser2@site.awsapps.com
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdatePrimaryEmailAddress](#)」の「」を参照してください。

update-resource

次の例は、update-resource を使用する方法を説明しています。

AWS CLI

リソースを更新するには

次のupdate-resourceコマンドは、指定されたりソースの名前を更新します。

```
aws workmail update-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-7afe0efbade843a58cdc10251fce992c \  
  --name exampleRoom2
```

このコマンドでは何も出力されません。

- API 詳細については、AWS CLI 「コマンドリファレンス[UpdateResource](#)」の「」を参照してください。

を使用した Amazon WorkMail メッセージフローの例 AWS CLI

次のコード例は、Amazon WorkMail Message Flow AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

`get-raw-message-content`

次のコード例は、`get-raw-message-content` を使用する方法を示しています。

AWS CLI

E メールメッセージの raw コンテンツを取得するには

次の`get-raw-message-content`例では、転送中の E メールメッセージの未加工コンテンツを取得し、`test` という名前のテキストファイルに送信します。

```
aws workmailmessageflow get-raw-message-content \  
  --message-id a1b2cd34-ef5g-6h7j-k18m-npq9012345rs \  
  test
```

コマンド実行`test`後のファイルの内容：

```
Subject: Hello World  
From: =?UTF-8?Q?marymajor_marymajor?= <marymajor@example.com>  
To: =?UTF-8?Q?mateojackson=40example=2Enet?= <mateojackson@example.net>  
Date: Thu, 7 Nov 2019 19:22:46 +0000  
Mime-Version: 1.0  
Content-Type: multipart/alternative;  
  boundary="=_EXAMPLE+"  
References: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>  
X-Priority: 3 (Normal)  
X-Mailer: Amazon WorkMail  
Thread-Index: EXAMPLE  
Thread-Topic: Hello World  
Message-Id: <mail.1ab23c45.5de6.7f890g123hj45678@storage.wm.amazon.com>
```

```
This is a multi-part message in MIME format. Your mail reader does not
understand MIME message format.
--=_EXAMPLE+
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

hello world

--=_EXAMPLE+
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE HTML><html>
<head>
<meta name=3D"Generator" content=3D"Amazon WorkMail v3.0-4510">
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3Dutf-8">=

<title>testing</title>
</head>
<body>
<p style=3D"margin: 0px; font-family: Arial, Tahoma, Helvetica, sans-seri=
f; font-size: small;">hello world</p>
</body>
</html>
--=_EXAMPLE+--
```

詳細については、「[Amazon WorkMail 管理者ガイド](#)」の [AWS 「Lambda を使用したメッセージコンテンツの取得」](#) を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [GetRawMessageContent](#)」の「」を参照してください。

WorkSpaces を使用した例 AWS CLI

次のコード例は、AWS Command Line Interface でを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています WorkSpaces。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

create-tags

次の例は、create-tags を使用する方法を説明しています。

AWS CLI

にタグを追加するには Workspace

次のcreate-tags例では、指定されたタグを指定された に追加します Workspace。

```
aws workspaces create-tags \  
  --resource-id ws-dk1xzz417 \  
  --tags Key=Department,Value=Finance
```

このコマンドでは何も出力されません。

詳細については、「Amazon 管理ガイド」の「[タグ WorkSpaces リソース](#)」を参照してください。 WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[CreateTags](#)」の「」を参照してください。

create-workspaces

次の例は、create-workspaces を使用する方法を説明しています。

AWS CLI

例 1: を作成するには AlwaysOn Workspace

次のcreate-workspaces例では、 AlwaysOn Workspace 指定されたディレクトリとバンドルを使用して、指定されたユーザーの を作成します。

```
aws workspaces create-workspaces \  
--workspaces DirectoryId=d-926722edaf,UserName=Mateo,BundleId=wsb-0zsvgp8fc
```

出力:

```
{  
  "FailedRequests": [],  
  "PendingRequests": [  
    {  
      "WorkspaceId": "ws-kcqms853t",  
      "DirectoryId": "d-926722edaf",  
      "UserName": "Mateo",  
      "State": "PENDING",  
      "BundleId": "wsb-0zsvgp8fc"  
    }  
  ]  
}
```

例 2: を作成するには AutoStop WorkSpace

次のcreate-workspaces例では、AutoStop WorkSpace 指定されたディレクトリとバンドルを使用して、指定されたユーザーのを作成します。

```
aws workspaces create-workspaces \  
--  
workspaces DirectoryId=d-926722edaf,UserName=Mary,BundleId=wsb-0zsvgp8fc,WorkspaceProperties
```

出力:

```
{  
  "FailedRequests": [],  
  "PendingRequests": [  
    {  
      "WorkspaceId": "ws-dk1x zr417",  
      "DirectoryId": "d-926722edaf",  
      "UserName": "Mary",  
      "State": "PENDING",  
      "BundleId": "wsb-0zsvgp8fc"  
    }  
  ]  
}
```

例 3: ユーザーデカップリングを作成するには WorkSpace

次のcreate-workspaces例では、ユーザー名を に設定し [UNDEFINED]、WorkSpace 名前、ディレクトリ ID、バンドル ID を指定 WorkSpace して、ユーザーデカップリングを作成します。

```
aws workspaces create-workspaces \
  --workspaces
  DirectoryId=d-926722edaf,UserName='"[UNDEFINED]"',WorkspaceName=MaryWorkspace1,BundleId=wsb-
```

出力:

```
{
  "FailedRequests": [],
  "PendingRequests": [
    {
      "WorkspaceId": "ws-abcd1234",
      "DirectoryId": "d-926722edaf",
      "UserName": "[UNDEFINED]",
      "State": "PENDING",
      "BundleId": "wsb-0zsvgp8fc",
      "WorkspaceName": "MaryWorkspace1"
    }
  ]
}
```

詳細については、「Amazon 管理ガイド」の [「仮想デスクトップの起動」](#) を参照してください。
WorkSpaces

- API 詳細については、AWS CLI 「 コマンドリファレンス [CreateWorkspaces](#) 」の「 」を参照してください。

delete-tags

次の例は、delete-tags を使用する方法を説明しています。

AWS CLI

からタグを削除するには WorkSpace

次のdelete-tags例では、指定された から指定されたタグを削除します WorkSpace。

```
aws workspaces delete-tags \  
  --resource-id ws-dk1xzt417 \  
  --tag-keys Department
```

このコマンドでは何も出力されません。

詳細については、「Amazon 管理ガイド」の「[タグ WorkSpaces リソース](#)」を参照してください。 WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteTags](#)」の「」を参照してください。

deregister-workspace-directory

次のコード例は、deregister-workspace-directory を使用する方法を示しています。

AWS CLI

ディレクトリの登録を解除するには

次のderegister-workspace-directory例では、指定されたディレクトリの登録を解除します。

```
aws workspaces deregister-workspace-directory \  
  --directory-id d-926722edaf
```

このコマンドでは何も出力されません。

詳細については、「[Amazon 管理ガイド](#)」の「[でディレクトリを登録する WorkSpaces](#)」を参照してください。 WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[DeregisterWorkspaceDirectory](#)」の「」を参照してください。

describe-tags

次の例は、describe-tags を使用する方法を説明しています。

AWS CLI

のタグを記述するには WorkSpace

次のdescribe-tags例では、指定された のタグについて説明します Workspace。

```
aws workspaces describe-tags \  
  --resource-id ws-dk1xzzr417
```

出力:

```
{  
  "TagList": [  
    {  
      "Key": "Department",  
      "Value": "Finance"  
    }  
  ]  
}
```

詳細については、「Amazon 管理ガイド」の「[タグ WorkSpaces リソース](#)」を参照してください。 WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeTags](#)」の「」を参照してください。

describe-workspace-bundles

次の例は、describe-workspace-bundles を使用する方法を説明しています。

AWS CLI

Amazon が提供するバンドルを一覧表示するには

次のdescribe-workspace-bundles例では、Amazon が提供するバンドルの名前と IDsを、テーブル形式で名前でソートして一覧表示します。

```
aws workspaces describe-workspace-bundles \  
  --owner AMAZON \  
  --query "Bundles[*].[Name, BundleId]"
```

出力:

```
[
```



```
[
  "Standard with Amazon Linux 2",
  "wsb-clj85qzj1"
],
[
  "Performance with Windows 10 (Server 2016 based)",
  "wsb-gm4d5tx2v"
],
[
  "PowerPro with Windows 7",
  "wsb-1pzkp0bx4"
],
[
  "Power with Amazon Linux 2",
  "wsb-2bs6k5lgn"
],
[
  "Graphics with Windows 10 (Server 2019 based)",
  "wsb-03gyjnfyy"
],
  ...
]
```

詳細については、「Amazon WorkSpaces 管理ガイド」の[WorkSpaces 「バンドルとイメージ」](#)を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeWorkspaceBundles](#)」の「」を参照してください。

describe-workspace-directories

次の例は、describe-workspace-directories を使用する方法を説明しています。

AWS CLI

登録済みディレクトリを記述するには

次のdescribe-workspace-directories例では、指定された登録ディレクトリについて説明します。

```
aws workspaces describe-workspace-directories \
  --directory-ids d-926722edaf
```

出力:

```
{
  "Directories": [
    {
      "DirectoryId": "d-926722edaf",
      "Alias": "d-926722edaf",
      "DirectoryName": "example.com",
      "RegistrationCode": "WSpdx+9RJ8JT",
      "SubnetIds": [
        "subnet-9d19c4c6",
        "subnet-500d5819"
      ],
      "DnsIpAddresses": [
        "172.16.1.140",
        "172.16.0.30"
      ],
      "CustomerUserName": "Administrator",
      "IamRoleId": "arn:aws:iam::123456789012:role/workspaces_DefaultRole",
      "DirectoryType": "SIMPLE_AD",
      "WorkspaceSecurityGroupId": "sg-0d89e927e5645d7c5",
      "State": "REGISTERED",
      "WorkspaceCreationProperties": {
        "EnableWorkDocs": false,
        "EnableInternetAccess": false,
        "UserEnabledAsLocalAdministrator": true,
        "EnableMaintenanceMode": true
      },
      "WorkspaceAccessProperties": {
        "DeviceTypeWindows": "ALLOW",
        "DeviceTypeOsx": "ALLOW",
        "DeviceTypeWeb": "DENY",
        "DeviceTypeIos": "ALLOW",
        "DeviceTypeAndroid": "ALLOW",
        "DeviceTypeChromeOs": "ALLOW",
        "DeviceTypeZeroClient": "ALLOW",
        "DeviceTypeLinux": "DENY"
      },
      "Tenancy": "SHARED",
      "SelfservicePermissions": {
        "RestartWorkspace": "ENABLED",
        "IncreaseVolumeSize": "DISABLED",
        "ChangeComputeType": "DISABLED",
        "SwitchRunningMode": "DISABLED",

```

```
        "RebuildWorkspace": "DISABLED"
      }
    }
  ]
}
```

詳細については、「Amazon WorkSpaces 管理ガイド」の「[のディレクトリの管理 WorkSpaces](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeWorkspaceDirectories](#)」の「」を参照してください。

describe-workspaces-connection-status

次のコード例は、describe-workspaces-connection-status を使用する方法を示しています。

AWS CLI

の接続ステータスを記述するには Workspace

次のdescribe-workspaces-connection-status例では、指定された の接続ステータスについて説明します Workspace。

```
aws workspaces describe-workspaces-connection-status \
  --workspace-ids ws-dk1xzr417
```

出力:

```
{
  "WorkspacesConnectionStatus": [
    {
      "WorkspaceId": "ws-dk1xzr417",
      "ConnectionState": "CONNECTED",
      "ConnectionStateCheckTimestamp": 1662526214.744
    }
  ]
}
```

詳細については、「Amazon WorkSpaces [管理ガイド](#)」の [WorkSpaces](#) 「の管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeWorkspacesConnectionStatus](#)」の「」を参照してください。

describe-workspaces

次のコード例は、describe-workspaces を使用する方法を示しています。

AWS CLI

を記述するには Workspace

次のdescribe-workspaces例では、指定された Workspace について説明します。

```
aws workspaces describe-workspaces \
  --workspace-ids ws-dk1xzr417
```

出力:

```
{
  "Workspaces": [
    {
      "WorkspaceId": "ws-dk1xzr417",
      "DirectoryId": "d-926722edaf",
      "UserName": "Mary",
      "IpAddress": "172.16.0.175",
      "State": "STOPPED",
      "BundleId": "wsb-0zsvgp8fc",
      "SubnetId": "subnet-500d5819",
      "ComputerName": "WSAMZN-RBSLTDD9",
      "WorkspaceProperties": {
        "RunningMode": "AUTO_STOP",
        "RunningModeAutoStopTimeoutInMinutes": 60,
        "RootVolumeSizeGib": 80,
        "UserVolumeSizeGib": 10,
        "ComputeTypeName": "VALUE"
      },
      "ModificationStates": []
    }
  ]
}
```

詳細については、「Amazon WorkSpaces [管理ガイド](#)」の [WorkSpaces](#) 「の管理」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeWorkspaces](#)」の「」を参照してください。

migrate-workspace

次のコード例は、migrate-workspace を使用する方法を示しています。

AWS CLI

を移行するには Workspace

次のmigrate-workspace例では、指定された を指定されたバンドルに移行 Workspace します。

```
aws workspaces migrate-workspace \  
  --source-workspace-id ws-dk1xzr417 \  
  --bundle-id wsb-j4dky1gs4
```

出力:

```
{  
  "SourceWorkspaceId": "ws-dk1xzr417",  
  "TargetWorkspaceId": "ws-x5h11bkp5"  
}
```

詳細については、「Amazon WorkSpaces [管理ガイド](#)」の「[の移行 Workspace](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[MigrateWorkspace](#)」の「」を参照してください。

modify-workspace-creation-properties

次のコード例は、modify-workspace-creation-properties を使用する方法を示しています。

AWS CLI

ディレクトリ Workspace の作成プロパティを変更するには

次のmodify-workspace-creation-properties例では、指定されたディレクトリのEnableInternetAccessプロパティを有効にします。これにより、ディレクトリ用にWorkSpaces 作成された のパブリック IP アドレスを自動的に割り当てることができます。

```
aws workspaces modify-workspace-creation-properties \  
  --resource-id d-926722edaf \  
  --workspace-creation-properties EnableInternetAccess=true
```

このコマンドでは何も出力されません。

詳細については、「Amazon WorkSpaces 管理ガイド」の「[のディレクトリの詳細を更新する WorkSpaces](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyWorkspaceCreationProperties](#)」の「」を参照してください。

modify-workspace-properties

次のコード例は、modify-workspace-properties を使用する方法を示しています。

AWS CLI

の実行モードを変更するには WorkSpace

次のmodify-workspace-properties例では、指定された の実行モード WorkSpace を に設定しますAUTO_STOP。

```
aws workspaces modify-workspace-properties \  
  --workspace-id ws-dk1xzr417 \  
  --workspace-properties RunningMode=AUTO_STOP
```

このコマンドでは何も出力されません。

詳細については、「Amazon WorkSpaces 管理ガイド」の「[の変更 WorkSpace](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyWorkspaceProperties](#)」の「」を参照してください。

modify-workspace-state

次の例は、modify-workspace-state を使用する方法を説明しています。

AWS CLI

の状態を変更するには Workspace

次のmodify-workspace-state例では、指定された の状態を Workspace に設定しますADMIN_MAINTENANCE。

```
aws workspaces modify-workspace-state \  
  --workspace-id ws-dk1xzt417 \  
  --workspace-state ADMIN_MAINTENANCE
```

このコマンドでは何も出力されません。

詳細については、「Amazon 管理ガイド」の[Workspace 「メンテナンス」](#)を参照してください。
WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[ModifyWorkspaceState](#)」の「」を参照してください。

reboot-workspaces

次のコード例は、reboot-workspaces を使用する方法を示しています。

AWS CLI

を再起動するには Workspace

次のreboot-workspaces例では、指定された を再起動します Workspace。

```
aws workspaces reboot-workspaces \  
  --reboot-workspace-requests ws-dk1xzt417
```

出力:

```
{  
  "FailedRequests": []  
}
```

詳細については、「Amazon WorkSpaces 管理ガイド」の「[の再起動 Workspace](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RebootWorkspaces](#)」の「」を参照してください。

rebuild-workspaces

次のコード例は、rebuild-workspaces を使用する方法を示しています。

AWS CLI

を再構築するには Workspace

次のrebuild-workspaces例では、指定された を再構築します Workspace。

```
aws workspaces rebuild-workspaces \  
  --rebuild-workspace-requests ws-dk1xzr417
```

出力:

```
{  
  "FailedRequests": []  
}
```

詳細については、「Amazon WorkSpaces 管理ガイド」の「[の再構築 Workspace](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RebuildWorkspaces](#)」の「」を参照してください。

register-workspace-directory

次の例は、register-workspace-directory を使用する方法を説明しています。

AWS CLI

ディレクトリを登録するには

次のregister-workspace-directory例では、Amazon で使用する指定されたディレクトリを登録します WorkSpaces。

```
aws workspaces register-workspace-directory \  
  --directory-id d-926722edaf \  
  --
```



```
--no-enable-work-docs
```

このコマンドでは何も出力されません。

詳細については、「[Amazon 管理ガイド](#)」の「[でディレクトリを登録する WorkSpaces](#)」を参照してください。 WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[RegisterWorkspaceDirectory](#)」の「」を参照してください。

restore-workspace

次の例は、restore-workspace を使用する方法を説明しています。

AWS CLI

を復元するには Workspace

次のrestore-workspace例では、指定された を復元します Workspace。

```
aws workspaces restore-workspace \  
  --workspace-id ws-dk1xzz417
```

このコマンドでは何も出力されません。

詳細については、「[Amazon WorkSpaces 管理ガイド](#)」の「[の復元 Workspace](#)」を参照してください。

- API 詳細については、AWS CLI 「コマンドリファレンス[RestoreWorkspace](#)」の「」を参照してください。

start-workspaces

次のコード例は、start-workspaces を使用する方法を示しています。

AWS CLI

を起動するには AutoStop Workspace

次のstart-workspaces例では、指定された を起動します Workspace。には の実行モード Workspace が必要ですAutoStop。

```
aws workspaces start-workspaces \  
--start-workspace-requests WorkspaceId=ws-dk1xzr417
```

出力:

```
{  
  "FailedRequests": []  
}
```

詳細については、「Amazon 管理ガイド」の「[を停止して開始 AutoStop WorkSpace](#)する」を参照してください。 WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[StartWorkspaces](#)」の「」を参照してください。

stop-workspaces

次の例は、stop-workspaces を使用する方法を説明しています。

AWS CLI

を停止するには AutoStop WorkSpace

次のstop-workspaces例では、指定された を停止しています WorkSpace。には の実行モード WorkSpace が必要ですAutoStop。

```
aws workspaces stop-workspaces \  
--stop-workspace-requests WorkspaceId=ws-dk1xzr417
```

出力:

```
{  
  "FailedRequests": []  
}
```

詳細については、「Amazon 管理ガイド」の「[を停止して開始 AutoStop WorkSpace](#)する」を参照してください。 WorkSpaces

- API 詳細については、AWS CLI 「コマンドリファレンス[StopWorkspaces](#)」の「」を参照してください。

terminate-workspaces

次の例は、`terminate-workspaces` を使用する方法を説明しています。

AWS CLI

を終了するには `WorkSpace`

次の`terminate-workspaces`例では、指定されたワークスペースを終了します。

```
aws workspaces terminate-workspaces \  
  --terminate-workspace-requests ws-dk1xzr417
```

出力:

```
{  
  "FailedRequests": []  
}
```

詳細については、「[Amazon 管理ガイド](#)」の「[を削除する WorkSpace](#)」を参照してください。

WorkSpaces

- API 詳細については、AWS CLI 「[コマンドリファレンス TerminateWorkspaces](#)」の「」を参照してください。

を使用した X-Ray の例 AWS CLI

次のコード例は、X-Ray AWS Command Line Interface で を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

batch-traces-get

次のコード例は、batch-traces-get を使用する方法を示しています。

AWS CLI

トレースのリストを取得するには

次のbatch-get-traces例では、ID で指定されたトレースのリストを取得します。トレース全体には、同一のトレース ID を使用して取得されるすべてのセグメントドキュメントからコンパイルされた、各セグメントのドキュメントが含まれます。

```
aws xray batch-get-traces \  
  --trace-ids 1-5d82881a-0a9126e92a73e971eed891b9
```

出力:

```
{  
  "Traces": [  
    {  
      "Id": "1-5d82881a-0a9126e92a73e971eed891b9",  
      "Duration": 0.232,  
      "Segments": [  
        {  
          "Id": "54aff5735b12dd28",  
          "Document": {"\id\":"54aff5735b12dd28","\name\":"  
          \Scorekeep","\start_time\":"1.568835610432E9","\end_time\":"1.568835610664E9,  
          \http\":{"request\":{"url\":"http://scorekeep-env-1.m4fg2pfzpv.us-  
          east-2.elasticbeanstalk.com/api/user","\method\":"POST","\user_agent\":"  
          \curl/7.59.0","\client_ip\":"52.95.4.28","\x_forwarded_for\":"true},  
          \response\":{"status\":"200}},\aws\":{"elastic_beanstalk\":{"version_label  
          \":"Sample Application-1","\deployment_id\":"3","\environment_name\":"Scorekeep-  
          env-1","\ec2\":{"availability_zone\":"us-east-2b","\instance_id\":"  
          \i-0e3cf4d2de0f3f37a"},"xray\":{"sdk_version\":"1.1.0","\sdk\":"X-Ray for  
          Java"},"service\":{"runtime\":"OpenJDK 64-Bit Server VM","\runtime_version  
          \":"1.8.0_222"},"trace_id\":"1-5d82881a-0a9126e92a73e971eed891b9",  
          \origin\":"AWS::ElasticBeanstalk::Environment","\subsegments\":[{"id\":"  
          \2d6900034ccfe558","\name\":"DynamoDB","\start_time\":"1.568835610658E9,  
          \end_time\":"1.568835610664E9","\http\":{"response\":{"status\":"200,  
          \content_length\":"61}},\aws\":{"table_name\":"scorekeep-user","\operation\":"
```

```

\ "UpdateItem", \ "request_id": \ "TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG
\ , \ "resource_names": [ \ "scorekeep-user" ], \ "namespace": \ "aws" ] ] }"
    },
    {
      "Id": "0f278b6334c34e6b",
      "Document": "{ \ "id": \ "0f278b6334c34e6b", \ "name":
\ "DynamoDB", \ "start_time": 1.568835610658E9, \ "end_time": 1.568835610664E9,
\ "parent_id": \ "2d6900034ccfe558", \ "inferred": true, \ "http": { \ "response
\ ": { \ "status": 200, \ "content_length": 61 } }, \ "aws": { \ "table_name
\ ": \ "scorekeep-user", \ "operation": \ "UpdateItem", \ "request_id":
\ "TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG", \ "resource_names":
[ \ "scorekeep-user" ], \ "trace_id": \ "1-5d82881a-0a9126e92a73e971eed891b9", \ "origin
\ ": \ "AWS::DynamoDB::Table" }"
    }
  ]
}
],
"UnprocessedTraceIds": []
}

```

詳細については、[AWS 「X-Ray デベロッパーガイド」の「APIでのAWS CLI X-Rayの使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchTracesGet](#)」の「」を参照してください。

create-group

次のコード例は、create-group を使用する方法を示しています。

AWS CLI

グループを作成するには

次のcreate-group例では、 という名前のグループリソースを作成しますAdminGroup。グループは、グループの基準を、障害またはエラーの原因となる特定のサービスに関連するセグメントとして定義するフィルター式を取得します。

```

aws xray create-group \
  --group-name "AdminGroup" \
  --filter-expression "service(\ "mydomain.com" ) {fault OR error}"

```

出力:

```
{
  "GroupName": "AdminGroup",
  "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",
  "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[CreateGroup](#)」の「」を参照してください。

create-sampling-rule

次のコード例は、create-sampling-rule を使用する方法を示しています。

AWS CLI

サンプリングルールを作成するには

次のcreate-sampling-rule例では、計測されたアプリケーションのサンプリング動作を制御するルールを作成します。ルールは JSON ファイルによって提供されます。ルールを作成するには、サンプリングルールフィールドの大部分が必要です。

```
aws xray create-sampling-rule \
  --cli-input-json file://9000-base-scorekeep.json
```

9000-base-scorekeep.json の内容:

```
{
  "SamplingRule": {
    "RuleName": "base-scorekeep",
    "ResourceARN": "*",
    "Priority": 9000,
    "FixedRate": 0.1,
    "ReservoirSize": 5,
    "ServiceName": "Scorekeep",
    "ServiceType": "*",
    "Host": "*",
    "HTTPMethod": "*"
  }
}
```

```
    "URLPath": "*",
    "Version": 1
  }
}
```

出力:

```
{
  "SamplingRuleRecord": {
    "SamplingRule": {
      "RuleName": "base-scorekeep",
      "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/base-
scorekeep",
      "ResourceARN": "*",
      "Priority": 9000,
      "FixedRate": 0.1,
      "ReservoirSize": 5,
      "ServiceName": "Scorekeep",
      "ServiceType": "*",
      "Host": "*",
      "HTTPMethod": "*",
      "URLPath": "*",
      "Version": 1,
      "Attributes": {}
    },
    "CreatedAt": 1530574410.0,
    "ModifiedAt": 1530574410.0
  }
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[CreateSamplingRule](#)」の「」を参照してください。

delete-group

次の例は、delete-group を使用する方法を説明しています。

AWS CLI

グループを削除するには

次のdelete-group例では、指定されたグループリソースを削除します。

```
aws xray delete-group \  
  --group-name "AdminGroup" \  
  --group-arn "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789"
```

このコマンドでは何も出力されません。

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteGroup](#)」の「」を参照してください。

delete-sampling-rule

次の例は、delete-sampling-rule を使用する方法を説明しています。

AWS CLI

サンプリングルールを削除するには

次のdelete-sampling-rule例では、指定されたサンプリングルールを削除します。グループ名またはグループを使用して、グループを指定できますARN。

```
aws xray delete-sampling-rule \  
  --rule-name polling-scorekeep
```

出力:

```
{  
  "SamplingRuleRecord": {  
    "SamplingRule": {  
      "RuleName": "polling-scorekeep",  
      "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/polling-scorekeep",  
      "ResourceARN": "*",  
      "Priority": 5000,  
      "FixedRate": 0.003,  
      "ReservoirSize": 0,  
      "ServiceName": "Scorekeep",  
      "ServiceType": "*",
```



```
        "Host": "*",
        "HTTPMethod": "GET",
        "URLPath": "/api/state/*",
        "Version": 1,
        "Attributes": {}
    },
    "CreatedAt": 1530574399.0,
    "ModifiedAt": 1530574399.0
}
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[DeleteSamplingRule](#)」の「」を参照してください。

get-encryption-config

次の例は、get-encryption-config を使用する方法を説明しています。

AWS CLI

暗号化設定を取得するには

次のget-encryption-config例では、AWS X-Ray データの現在の暗号化設定を取得します。

```
aws xray get-encryption-config
```

出力:

```
{
  "EncryptionConfig": {
    "KeyId": "ae4aa6d49-a4d8-9df9-a475-4ff6d7898456",
    "Status": "ACTIVE",
    "Type": "NONE"
  }
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetEncryptionConfig](#)」の「」を参照してください。

get-group

次の例は、get-group を使用する方法を説明しています。

AWS CLI

グループを取得するには

次のget-group例では、指定されたグループリソースの詳細を表示します。詳細には、グループ名、グループ ARN、およびそのグループの条件を定義するフィルター式が含まれます。グループは、によって取得することもできますARN。

```
aws xray get-group \  
  --group-name "AdminGroup"
```

出力:

```
{  
  "Group": [  
    {  
      "GroupName": "AdminGroup",  
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/  
AdminGroup/123456789",  
      "FilterExpression": "service(\"mydomain.com\") {fault OR error}"  
    }  
  ]  
}
```

詳細については、[AWS 「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetGroup](#)」の「」を参照してください。

get-groups

次のコード例は、get-groups を使用する方法を示しています。

AWS CLI

すべてのグループを取得するには

次の例では、すべてのアクティブなグループの詳細を表示します。

```
aws xray get-groups
```

出力:

```
{
  "Groups": [
    {
      "GroupName": "AdminGroup",
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",
      "FilterExpression": "service(\"example.com\") {fault OR error}"
    },
    {
      "GroupName": "SDETGroup",
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/SDETGroup/987654321",
      "FilterExpression": "responsetime > 2"
    }
  ]
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[GetGroups](#)」の「」を参照してください。

get-sampling-rules

次のコード例は、get-sampling-rules を使用する方法を示しています。

AWS CLI

すべてのサンプリングルールを取得するには

次のget-sampling-rules例では、使用可能なすべてのサンプリングルールの詳細を表示します。

aws xray get-sampling-rules

出力:

```
{
  "SamplingRuleRecords": [
    {
      "SamplingRule": {
        "RuleName": "Default",
        "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/Default",
        "ResourceARN": "*",
        "Priority": 10000,
        "FixedRate": 0.01,
        "ReservoirSize": 0,
        "ServiceName": "*",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1,
        "Attributes": {}
      },
      "CreatedAt": 0.0,
      "ModifiedAt": 1530558121.0
    },
    {
      "SamplingRule": {
        "RuleName": "base-scorekeep",
        "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/base-scorekeep",
        "ResourceARN": "*",
        "Priority": 9000,
        "FixedRate": 0.1,
        "ReservoirSize": 2,
        "ServiceName": "Scorekeep",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1,
        "Attributes": {}
      },
      "CreatedAt": 1530573954.0,
      "ModifiedAt": 1530920505.0
    }
  ]
}
```

```
    },
    {
      "SamplingRule": {
        "RuleName": "polling-scorekeep",
        "RuleARN": "arn:aws:xray:us-east-1::sampling-rule/polling-
scorekeep",
        "ResourceARN": "*",
        "Priority": 5000,
        "FixedRate": 0.003,
        "ReservoirSize": 0,
        "ServiceName": "Scorekeep",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "GET",
        "URLPath": "/api/state/*",
        "Version": 1,
        "Attributes": {}
      },
      "CreatedAt": 1530918163.0,
      "ModifiedAt": 1530918163.0
    }
  ]
}
```

詳細については、[「X-Ray デベロッパーガイド」の「X-Ray でのサンプリングルールの使用 API」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSamplingRules](#)」の「」を参照してください。

get-sampling-targets

次のコード例は、get-sampling-targets を使用する方法を示しています。

AWS CLI

サンプリングクォータをリクエストするには

次のget-sampling-targets例では、サービスがリクエストのサンプリングに使用しているルールのサンプリングクォータをリクエストします。AWS X-Ray からのレスポンスには、リザーバから借りる代わりに使用できるクォータが含まれています。

```
aws xray get-sampling-targets \
```

```
--sampling-statistics-documents '[ { "RuleName": "base-scorekeep", "ClientID": "ABCDEF1234567890ABCDEF10", "Timestamp": "2018-07-07T00:20:06", "RequestCount": 110, "SampledCount": 20, "BorrowCount": 10 }, { "RuleName": "polling-scorekeep", 31, "BorrowCount": 0 } ]'
```

出力:

```
{
  "SamplingTargetDocuments": [
    {
      "RuleName": "base-scorekeep",
      "FixedRate": 0.1,
      "ReservoirQuota": 2,
      "ReservoirQuotaTTL": 1530923107.0,
      "Interval": 10
    },
    {
      "RuleName": "polling-scorekeep",
      "FixedRate": 0.003,
      "ReservoirQuota": 0,
      "ReservoirQuotaTTL": 1530923107.0,
      "Interval": 10
    }
  ],
  "LastRuleModification": 1530920505.0,
  "UnprocessedStatistics": []
}
```

詳細については、[「X-Ray デベロッパーガイド」の「X-Ray でのサンプリングルールの使用 API」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetSamplingTargets](#)」の「」を参照してください。

get-service-graph

次の例は、get-service-graph を使用する方法を説明しています。

AWS CLI

サービスグラフを取得するには

次の例は、受信リクエストを処理するサービスと、その結果呼び出したダウンストリームサービスを説明する、指定された期間内にドキュメントを表示します。

```
aws xray get-service-graph \  
  --start-time 1568835392.0 \  
  --end-time 1568835446.0
```

出力:

```
{  
  "Services": [  
    {  
      "ReferenceId": 0,  
      "Name": "Scorekeep",  
      "Names": [  
        "Scorekeep"  
      ],  
      "Root": true,  
      "Type": "AWS::ElasticBeanstalk::Environment",  
      "State": "active",  
      "StartTime": 1568835392.0,  
      "EndTime": 1568835446.0,  
      "Edges": [  
        {  
          "ReferenceId": 1,  
          "StartTime": 1568835392.0,  
          "EndTime": 1568835446.0,  
          "SummaryStatistics": {  
            "OkCount": 14,  
            "ErrorStatistics": {  
              "ThrottleCount": 0,  
              "OtherCount": 0,  
              "TotalCount": 0  
            },  
            "FaultStatistics": {  
              "OtherCount": 0,  
              "TotalCount": 0  
            },  
            "TotalCount": 14,  
            "TotalResponseTime": 0.13  
          },  
          "ResponseTimeHistogram": [  
            {
```

```
        "Value": 0.008,
        "Count": 1
      },
      {
        "Value": 0.005,
        "Count": 7
      },
      {
        "Value": 0.009,
        "Count": 1
      },
      {
        "Value": 0.021,
        "Count": 1
      },
      {
        "Value": 0.038,
        "Count": 1
      },
      {
        "Value": 0.007,
        "Count": 1
      },
      {
        "Value": 0.006,
        "Count": 2
      }
    ],
    "Aliases": []
  },
  ... TRUNCATED FOR BREVITY ...

]
}
],
"StartTime": 1568835392.0,
"EndTime": 1568835446.0,
"ContainsOldGroupVersions": false
}
```

詳細については、[AWS 「X-Ray デベロッパーガイド」の「APIでの AWS CLI X-Ray の使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetServiceGraph](#)」の「」を参照してください。

get-trace-summaries

次のコード例は、get-trace-summaries を使用する方法を示しています。

AWS CLI

トレースの概要を取得するには

次のget-trace-summaries例では、指定された時間枠内で使用可能なトレースの IDs と メタデータを取得します。

```
aws xray get-trace-summaries \  
  --start-time 1568835392.0 \  
  --end-time 1568835446.0
```

出力:

```
[  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/  
VSAE93HF/GSSD2NTB/DP0PCC09",  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/move/  
GCQ2B35P/FREELDFT/4LRE643M",  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/game/  
VSAE93HF/GSSD2NTB/starttime/1568835513",  
  "http://scorekeep-env-1.123456789.us-east-2.elasticbeanstalk.com/api/  
move/4MQNA5NN/L99KK2RF/null"  
]
```

詳細については、[AWS 「X-Ray デベロッパーガイド」の「APIでの AWS CLI X-Ray の使用」](#)を参照してください。AWS

- API 詳細については、AWS CLI 「コマンドリファレンス[GetTraceSummaries](#)」の「」を参照してください。

put-encryption-config

次の例は、put-encryption-config を使用する方法を説明しています。

AWS CLI

暗号化設定を更新するには

次の `put-encryption-config` `example` updates the encryption configuration for AWS X-Ray data to use the default AWS managed KMS key `aws/xray`.

```
aws xray put-encryption-config \  
  --type KMS \  
  --key-id alias/aws/xray
```

出力:

```
{  
  "EncryptionConfig": {  
    "KeyId": "arn:aws:kms:us-west-2:123456789012:key/c234g4e8-39e9-4gb0-84e2-  
b0ea215cbba5",  
    "Status": "UPDATING",  
    "Type": "KMS"  
  }  
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、および暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[PutEncryptionConfig](#)」の「」を参照してください。

put-trace-segments

次のコード例は、`put-trace-segments` を使用する方法を示しています。

AWS CLI

セグメントをアップロードするには

次の `put-trace-segments` 例では、セグメントドキュメントを AWS X-Ray にアップロードします。セグメントドキュメントは、JSONセグメントドキュメントのリストとして消費されません。

```
aws xray put-trace-segments \  
  \
```

```
--trace-segment-documents "{\"id\":\"20312a0e2b8809f4\",\"name
\": \"DynamoDB\", \"trace_id\":\"1-5832862d-a43aafded3334a971fe312db\",
\": \"start_time\":1.479706157195E9, \"end_time\":1.479706157202E9, \"parent_id\":
\": \"79736b962fe3239e\", \"http\": {\"response\": {\"content_length\":60, \"status
\":200}}, \"inferred\":true, \"aws\": {\"consistent_read\":false, \"table_name
\": \"scorekeep-session-xray\", \"operation\": \"GetItem\", \"request_id\":
\": \"SCAU230M6M8F038UASGC7785ARVV4KQNS05AEMVJF66Q9ASUAAJG\", \"resource_names\":
\": [\"scorekeep-session-xray\"]}, \"origin\": \"AWS::DynamoDB::Table\"}"
```

出力:

```
{
  "UnprocessedTraceSegments": []
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray へのトレースデータの送信」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[PutTraceSegments](#)」の「」を参照してください。

update-group

次の例は、update-group を使用する方法を説明しています。

AWS CLI

グループを更新するには

次のupdate-group例では、 という名前のグループへのトレースを受け入れる基準を更新しますAdminGroup。グループ名またはグループ を使用して、目的のグループを指定できますARN。

```
aws xray update-group \
  --group-name "AdminGroup" \
  --group-arn "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789" \
  --filter-expression "service(\"mydomain.com\") {fault}"
```

出力:

```
{
  "GroupName": "AdminGroup",
  "GroupARN": "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789",
```

```
"FilterExpression": "service(\"mydomain.com\") {fault}"
}
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateGroup](#)」の「」を参照してください。

update-sampling-rule

次のコード例は、update-sampling-rule を使用する方法を示しています。

AWS CLI

サンプリングルールを更新するには

次のupdate-sampling-rule例では、サンプリングルールの設定を変更します。ルールはJSON ファイルから消費されます。更新されるフィールドのみが必要です。

```
aws xray update-sampling-rule \  
  --cli-input-json file://1000-default.json
```

1000-default.json の内容:

```
{  
  "SamplingRuleUpdate": {  
    "RuleName": "Default",  
    "FixedRate": 0.01,  
    "ReservoirSize": 0  
  }  
}
```

出力:

```
{  
  "SamplingRuleRecords": [  
    {  
      "SamplingRule": {  
        "RuleName": "Default",  
        "RuleARN": "arn:aws:xray:us-west-2:123456789012:sampling-rule/  
Default",
```

```
        "ResourceARN": "*",
        "Priority": 10000,
        "FixedRate": 0.01,
        "ReservoirSize": 0,
        "ServiceName": "*",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1,
        "Attributes": {}
    },
    "CreatedAt": 0.0,
    "ModifiedAt": 1529959993.0
}
]
```

詳細については、[AWS「X-Ray デベロッパーガイド」の「X-Ray を使用したサンプリング、グループ、暗号化設定の設定API」](#)を参照してください。AWS

- API 詳細については、AWS CLI「コマンドリファレンス[UpdateSamplingRule](#)」の「」を参照してください。

AWS CLI Bash スクリプトコードの例

このトピックのコード例は、を AWS Command Line Interface で Bash スクリプトで使用方法を示しています AWS。

「基本」は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

「シナリオ」は、1つのサービス内から、または他の AWS のサービスと組み合わせて複数の関数を呼び出し、特定のタスクを実行する方法を示すコード例です。

サービス

- [Bash スクリプト AWS CLI を使用した DynamoDB の例](#)
- [Bash スクリプト AWS CLI で を使用する Amazon EC2 の例](#)

- [HealthImaging Bash スクリプト AWS CLI で を使用する例](#)
- [IAM Bash スクリプト AWS CLI で を使用する例](#)
- [Bash スクリプトを使用した Amazon S3 の例 AWS CLI](#)
- [AWS STS Bash スクリプト AWS CLI で を使用する例](#)

Bash スクリプト AWS CLI を使用した DynamoDB の例

次のコード例は、DynamoDB AWS Command Line Interface で Bash スクリプト を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

「基本」は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [基本](#)
- [アクション](#)

基本

基本を学ぶ

次のコードサンプルは、以下の操作方法を示しています。

- 映画データを保持できるテーブルを作成する。
- テーブルに 1 つの映画を入れ、取得して更新する。
- サンプルJSONファイルからテーブルに映画データを書き込みます。
- 特定の年にリリースされた映画を照会する。
- 何年もの間にリリースされた映画をスキャンする。
- テーブルからムービーを削除し、テーブルを削除します。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

DynamoDB の開始シナリオ。

```
#####
# function dynamodb_getting_started_movies
#
# Scenario to create an Amazon DynamoDB table and perform a series of operations on
the table.
#
# Returns:
#     0 - If successful.
#     1 - If an error occurred.
#####
function dynamodb_getting_started_movies() {

    source ./dynamodb_operations.sh

    key_schema_json_file="dynamodb_key_schema.json"
    attribute_definitions_json_file="dynamodb_attr_def.json"
    item_json_file="movie_item.json"
    key_json_file="movie_key.json"
    batch_json_file="batch.json"
    attribute_names_json_file="attribute_names.json"
    attributes_values_json_file="attribute_values.json"

    echo_repeat "*" 88
    echo
    echo "Welcome to the Amazon DynamoDB getting started demo."
    echo
    echo_repeat "*" 88
    echo

    local table_name
    echo -n "Enter a name for a new DynamoDB table: "
    get_input
    table_name=$get_input_result
}
```

```
local provisioned_throughput="ReadCapacityUnits=5,WriteCapacityUnits=5"

echo '['
{"AttributeName": "year", "KeyType": "HASH"},
{"AttributeName": "title", "KeyType": "RANGE"}
]' >"$key_schema_json_file"

echo '['
{"AttributeName": "year", "AttributeType": "N"},
{"AttributeName": "title", "AttributeType": "S"}
]' >"$attribute_definitions_json_file"

if dynamodb_create_table -n "$table_name" -a "$attribute_definitions_json_file" \
-k "$key_schema_json_file" -p "$provisioned_throughput" 1>/dev/null; then
  echo "Created a DynamoDB table named $table_name"
else
  errecho "The table failed to create. This demo will exit."
  clean_up
  return 1
fi

echo "Waiting for the table to become active...."

if dynamodb_wait_table_active -n "$table_name"; then
  echo "The table is now active."
else
  errecho "The table failed to become active. This demo will exit."
  cleanup "$table_name"
  return 1
fi

echo
echo_repeat "*" 88
echo

echo -n "Enter the title of a movie you want to add to the table: "
get_input
local added_title
added_title=$get_input_result

local added_year
get_int_input "What year was it released? "
added_year=$get_input_result
```



```
local rating
get_float_input "On a scale of 1 - 10, how do you rate it? " "1" "10"
rating=$get_input_result

local plot
echo -n "Summarize the plot for me: "
get_input
plot=$get_input_result

echo '{
  "year": {"N" : ""$added_year""},
  "title": {"S" : ""$added_title""},
  "info": {"M" : {"plot": {"S" : ""$plot""}, "rating": {"N" : ""$rating""} } }
}' >"$item_json_file"

if dynamodb_put_item -n "$table_name" -i "$item_json_file"; then
  echo "The movie '$added_title' was successfully added to the table
'$table_name'."
else
  errecho "Put item failed. This demo will exit."
  clean_up "$table_name"
  return 1
fi

echo
echo_repeat "*" 88
echo

echo "Let's update your movie '$added_title'."
get_float_input "You rated it $rating, what new rating would you give it? " "1"
"10"
rating=$get_input_result

echo -n "You summarized the plot as '$plot'."
echo "What would you say now? "
get_input
plot=$get_input_result

echo '{
  "year": {"N" : ""$added_year""},
  "title": {"S" : ""$added_title""}
}' >"$key_json_file"
```

```
echo '{
  "r": {"N" : ""$rating""},
  "p": {"S" : ""$plot""}
}' >"$item_json_file"

local update_expression="SET info.rating = :r, info.plot = :p"

if dynamodb_update_item -n "$table_name" -k "$key_json_file" -e
"$update_expression" -v "$item_json_file"; then
  echo "Updated '$added_title' with new attributes."
else
  errecho "Update item failed. This demo will exit."
  clean_up "$table_name"
  return 1
fi

echo
echo_repeat "*" 88
echo

echo "We will now use batch write to upload 150 movie entries into the table."

local batch_json
for batch_json in movie_files/movies_*.json; do
  echo "{ \"$table_name\" : $(<"$batch_json") }" >"$batch_json_file"
  if dynamodb_batch_write_item -i "$batch_json_file" 1>/dev/null; then
    echo "Entries in $batch_json added to table."
  else
    errecho "Batch write failed. This demo will exit."
    clean_up "$table_name"
    return 1
  fi
done

local title="The Lord of the Rings: The Fellowship of the Ring"
local year="2001"

if get_yes_no_input "Let's move on...do you want to get info about '$title'? (y/n)
"; then
  echo '{
"year": {"N" : ""$year""},
"title": {"S" : ""$title""}
}' >"$key_json_file"
  local info
```

```
info=$(dynamodb_get_item -n "$table_name" -k "$key_json_file")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "Get item failed. This demo will exit."
    clean_up "$table_name"
    return 1
fi

echo "Here is what I found:"
echo "$info"
fi

local ask_for_year=true
while [[ "$ask_for_year" == true ]]; do
    echo "Let's get a list of movies released in a given year."
    get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"
    year=$get_input_result
    echo '{
"#n": "year"
}' >"$attribute_names_json_file"

    echo '{
":v": {"N" :"""$year"""}
}' >"$attributes_values_json_file"

    response=$(dynamodb_query -n "$table_name" -k "#n=:v" -a
"$attribute_names_json_file" -v "$attributes_values_json_file")

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "Query table failed. This demo will exit."
    clean_up "$table_name"
    return 1
fi

echo "Here is what I found:"
echo "$response"

if ! get_yes_no_input "Try another year? (y/n) "; then
    ask_for_year=false
fi
done
```

```
echo "Now let's scan for movies released in a range of years. Enter a year: "  
get_int_input "Enter a year between 1972 and 2018: " "1972" "2018"  
local start=$get_input_result  
  
get_int_input "Enter another year: " "1972" "2018"  
local end=$get_input_result  
  
echo '{  
  "#n": "year"  
}' >"$attribute_names_json_file"  
  
echo '{  
  ":v1": {"N" : ""$start""},  
  ":v2": {"N" : ""$end""}  
}' >"$attributes_values_json_file"  
  
response=$(dynamodb_scan -n "$table_name" -f "#n BETWEEN :v1 AND :v2" -a  
"$attribute_names_json_file" -v "$attributes_values_json_file")  
  
# shellcheck disable=SC2181  
if [[ ${?} -ne 0 ]]; then  
  errecho "Scan table failed. This demo will exit."  
  clean_up "$table_name"  
  return 1  
fi  
  
echo "Here is what I found:"  
echo "$response"  
  
echo  
echo_repeat "*" 88  
echo  
  
echo "Let's remove your movie '$added_title' from the table."  
  
if get_yes_no_input "Do you want to remove '$added_title'? (y/n) "; then  
  echo '{  
"year": {"N" : ""$added_year""},  
"title": {"S" : ""$added_title""}  
}' >"$key_json_file"  
  
  if ! dynamodb_delete_item -n "$table_name" -k "$key_json_file"; then  
    errecho "Delete item failed. This demo will exit."  
    clean_up "$table_name"
```

```

        return 1
    fi
fi

if get_yes_no_input "Do you want to delete the table '$table_name'? (y/n) "; then
    if ! clean_up "$table_name"; then
        return 1
    fi
else
    if ! clean_up; then
        return 1
    fi
fi

return 0
}

```

このシナリオで使用される DynamoDB 関数。

```

#####
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table to create.
#     -a attribute_definitions -- JSON file path of a list of attributes and their
types.
#     -k key_schema -- JSON file path of a list of attributes and their key types.
#     -p provisioned_throughput -- Provisioned throughput settings for the table.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_create_table() {
    local table_name attribute_definitions key_schema provisioned_throughput response
    local option OPTARG # Required to use getopt command in a function.

#####
# Function usage explanation
#####

```

```
function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table."
    echo " -n table_name -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
types."
    echo " -p provisioned_throughput -- Provisioned throughput settings for the
table."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:a:k:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        a) attribute_definitions="${OPTARG}" ;;
        k) key_schema="${OPTARG}" ;;
        p) provisioned_throughput="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$attribute_definitions" ]]; then
    errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
    usage
    return 1
fi
```

```
fi

if [[ -z "$key_schema" ]]; then
    errecho "ERROR: You must provide a key schema json file path the -k parameter."
    usage
    return 1
fi

if [[ -z "$provisioned_throughput" ]]; then
    errecho "ERROR: You must provide a provisioned throughput json file path the -p
parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:    $table_name"
iecho "    attribute_definitions:  $attribute_definitions"
iecho "    key_schema:    $key_schema"
iecho "    provisioned_throughput:  $provisioned_throughput"
iecho ""

response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"${attribute_definitions}" \
    --key-schema file://"${key_schema}" \
    --provisioned-throughput "$provisioned_throughput")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
fi

return 0
}

#####
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
```

```

# Parameters:
#     -n table_name  -- The name of the table.
#
# Response:
#     - TableStatus:
#     And:
#     0 - Table is active.
#     1 - If it fails.
#####
function dynamodb_describe_table {
    local table_name
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_describe_table"
        echo "Describe the status of a DynamoDB table."
        echo "  -n table_name  -- The name of the table."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$table_name" ]]; then
        errecho "ERROR: You must provide a table name with the -n parameter."
        usage
        return 1
    fi
}

```



```

fi

local table_status
table_status=$(
  aws dynamodb describe-table \
    --table-name "$table_name" \
    --output text \
    --query 'Table.TableStatus'
)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log "$error_code"
  errecho "ERROR: AWS reports describe-table operation failed.$table_status"
  return 1
fi

echo "$table_status"

return 0
}

#####
# function dynamodb_put_item
#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#   -n table_name -- The name of the table.
#   -i item -- Path to json file containing the item values.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_put_item() {
  local table_name item response
  local option OPTARG # Required to use getopt command in a function.

  #####
  # Function usage explanation
  #####
  function usage() {

```

```
    echo "function dynamodb_put_item"
    echo "Put an item into a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -i item -- Path to json file containing the item values."
    echo ""
}

while getopts "n:i:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        i) item="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  item:       $item"
iecho ""
iecho ""

response=$(aws dynamodb put-item \
    --table-name "$table_name" \
```

```

    --item file://"${item}")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports put-item operation failed.${response}"
    return 1
fi

return 0

}

#####
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
# Parameters:
#   -n table_name -- The name of the table.
#   -k keys -- Path to json file containing the keys that identify the item to
#   update.
#   -e update expression -- An expression that defines one or more attributes
#   to be updated.
#   -v values -- Path to json file containing the update values.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_update_item() {
    local table_name keys update_expression values response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_update_item"
        echo "Update an item in a DynamoDB table."
        echo " -n table_name -- The name of the table."
    }
}

```

```
    echo " -k keys -- Path to json file containing the keys that identify the item
to update."
    echo " -e update expression -- An expression that defines one or more
attributes to be updated."
    echo " -v values -- Path to json file containing the update values."
    echo ""
}

while getopts "n:k:e:v:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) keys="${OPTARG}" ;;
        e) update_expression="${OPTARG}" ;;
        v) values="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

if [[ -z "$update_expression" ]]; then
    errecho "ERROR: You must provide an update expression with the -e parameter."
    usage
    return 1
fi
```

```

if [[ -z "$values" ]]; then
    errecho "ERROR: You must provide a values json file path the -v parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  keys:       $keys"
iecho "  update_expression:  $update_expression"
iecho "  values:     $values"

response=$(aws dynamodb update-item \
    --table-name "$table_name" \
    --key file://" $keys" \
    --update-expression "$update_expression" \
    --expression-attribute-values file://" $values")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports update-item operation failed.$response"
    return 1
fi

return 0
}

#####
# function dynamodb_batch_write_item
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#   -i item -- Path to json file containing the items to write.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_batch_write_item() {
    local item response

```

```
local option OPTARG # Required to use getopt command in a function.

#####
# Function usage explanation
#####
function usage() {
    echo "function dynamodb_batch_write_item"
    echo "Write a batch of items into a DynamoDB table."
    echo " -i item -- Path to json file containing the items to write."
    echo ""
}
while getopt "i:h" option; do
    case "${option}" in
        i) item="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  item:       $item"
iecho ""

response=$(aws dynamodb batch-write-item \
    --request-items file://"${item}")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
```

```

    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
fi

return 0
}

#####
# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#   -n table_name  -- The name of the table.
#   -k keys        -- Path to json file containing the keys that identify the item to
#   get.
#   [-q query]    -- Optional JMESPath query expression.
#
# Returns:
#   The item as text output.
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_get_item() {
    local table_name keys query response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    # #####
    function usage() {
        echo "function dynamodb_get_item"
        echo "Get an item from a DynamoDB table."
        echo " -n table_name  -- The name of the table."
        echo " -k keys        -- Path to json file containing the keys that identify the item
to get."
        echo " [-q query]    -- Optional JMESPath query expression."
        echo ""
    }
    query=""
    while getopt "n:k:q:h" option; do
        case "${option}" in

```

```
n) table_name="${OPTARG}" ;;
k) keys="${OPTARG}" ;;
q) query="${OPTARG}" ;;
h)
    usage
    return 0
    ;;
\?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

if [[ -n "$query" ]]; then
    response=$(aws dynamodb get-item \
        --table-name "$table_name" \
        --key file://"${keys}" \
        --output text \
        --query "$query")
else
    response=$(
        aws dynamodb get-item \
            --table-name "$table_name" \
            --key file://"${keys}" \
            --output text
    )
fi

local error_code=${?}
```



```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
fi

if [[ -n "$query" ]]; then
    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
query inserts on some strings.
else
    echo "$response"
fi

return 0
}

#####
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -k key_condition_expression -- The key condition expression.
#     -a attribute_names -- Path to JSON file containing the attribute names.
#     -v attribute_values -- Path to JSON file containing the attribute values.
#     [-p projection_expression] -- Optional projection expression.
#
# Returns:
#     The items as json output.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_query() {
    local table_name key_condition_expression attribute_names attribute_values
projection_expression response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####
function usage() {

```

```
    echo "function dynamodb_query"
    echo "Query a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -k key_condition_expression -- The key condition expression."
    echo " -a attribute_names -- Path to JSON file containing the attribute names."
    echo " -v attribute_values -- Path to JSON file containing the attribute
values."
    echo " [-p projection_expression] -- Optional projection expression."
    echo ""
}

while getopts "n:k:a:v:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) key_condition_expression="${OPTARG}" ;;
        a) attribute_names="${OPTARG}" ;;
        v) attribute_values="${OPTARG}" ;;
        p) projection_expression="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$key_condition_expression" ]]; then
    errecho "ERROR: You must provide a key condition expression with the -k
parameter."
    usage
    return 1
fi
```

```
if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
fi

if [[ -z "$attribute_values" ]]; then
    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
fi

if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"${attribute_names}" \
        --expression-attribute-values file://"${attribute_values}")
else
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"${attribute_names}" \
        --expression-attribute-values file://"${attribute_values}" \
        --projection-expression "$projection_expression")
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function dynamodb_scan
#
# This function scans a DynamoDB table.
```

```

#
# Parameters:
#   -n table_name  -- The name of the table.
#   -f filter_expression  -- The filter expression.
#   -a expression_attribute_names  -- Path to JSON file containing the expression
#   attribute names.
#   -v expression_attribute_values  -- Path to JSON file containing the
#   expression attribute values.
#   [-p projection_expression]  -- Optional projection expression.
#
# Returns:
#   The items as json output.
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_scan() {
    local table_name filter_expression expression_attribute_names
    expression_attribute_values projection_expression response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    # #####
    function usage() {
        echo "function dynamodb_scan"
        echo "Scan a DynamoDB table."
        echo " -n table_name  -- The name of the table."
        echo " -f filter_expression  -- The filter expression."
        echo " -a expression_attribute_names  -- Path to JSON file containing the
#   expression attribute names."
        echo " -v expression_attribute_values  -- Path to JSON file containing the
#   expression attribute values."
        echo " [-p projection_expression]  -- Optional projection expression."
        echo ""
    }
}

while getopt "n:f:a:v:p:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        f) filter_expression="${OPTARG}" ;;
        a) expression_attribute_names="${OPTARG}" ;;
        v) expression_attribute_values="${OPTARG}" ;;
        p) projection_expression="${OPTARG}" ;;
    esac
done

```

```
h)
  usage
  return 0
  ;;
\?)
  echo "Invalid parameter"
  usage
  return 1
  ;;
esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$filter_expression" ]]; then
  errecho "ERROR: You must provide a filter expression with the -f parameter."
  usage
  return 1
fi

if [[ -z "$expression_attribute_names" ]]; then
  errecho "ERROR: You must provide expression attribute names with the -a
parameter."
  usage
  return 1
fi

if [[ -z "$expression_attribute_values" ]]; then
  errecho "ERROR: You must provide expression attribute values with the -v
parameter."
  usage
  return 1
fi

if [[ -z "$projection_expression" ]]; then
  response=$(aws dynamodb scan \
    --table-name "$table_name" \
    --filter-expression "$filter_expression" \
    --expression-attribute-names file://"${expression_attribute_names} \
```

```

    --expression-attribute-values file://"$expression_attribute_values")
else
    response=$(aws dynamodb scan \
        --table-name "$table_name" \
        --filter-expression "$filter_expression" \
        --expression-attribute-names file://"$expression_attribute_names" \
        --expression-attribute-values file://"$expression_attribute_values" \
        --projection-expression "$projection_expression")
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports scan operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -k keys -- Path to json file containing the keys that identify the item to
# delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_delete_item() {
    local table_name keys response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####

```

```
function usage() {
    echo "function dynamodb_delete_item"
    echo "Delete an item from a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -k keys -- Path to json file containing the keys that identify the item
to delete."
    echo ""
}
while getopts "n:k:h" option; do
    case "${option}" in
        n) table_name="${OPTARG}" ;;
        k) keys="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  keys:       $keys"
iecho ""

response=$(aws dynamodb delete-item \
    --table-name "$table_name" \
```

```
--key file://"keys")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
fi

return 0

}

#####
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#     -n table_name  -- The name of the table to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_delete_table() {
    local table_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function dynamodb_delete_table"
        echo "Deletes an Amazon DynamoDB table."
        echo " -n table_name  -- The name of the table to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            h)
                usage

```



```

        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho ""

response=$(aws dynamodb delete-table \
    --table-name "$table_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-table operation failed.$response"
    return 1
fi

return 0
}

```

このシナリオで使用されるユーティリティ関数。

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.

```

```
#####
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  fi
}
```

```
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- API 詳細については、AWS CLI コマンドリファレンスの以下のトピックを参照してください。
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [Query](#)
 - [Scan](#)
 - [UpdateItem](#)

アクション

BatchGetItem

次のコード例は、BatchGetItem を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function dynamodb_batch_get_item
#
# This function gets a batch of items from a DynamoDB table.
#
# Parameters:
#     -i item -- Path to json file containing the keys of the items to get.
#
# Returns:
#     The items as json output.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_batch_get_item() {
    local item response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_batch_get_item"
        echo "Get a batch of items from a DynamoDB table."
        echo " -i item -- Path to json file containing the keys of the items to get."
        echo ""
    }

    while getopt "i:h" option; do
        case "${option}" in
            i) item="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1
}
```

```

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

response=$(aws dynamodb batch-get-item \
    --request-items file://"${item}")
local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-get-item operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:

```

```
#      $1 - The error code returned by the AWS CLI.
#
# Returns:
#      0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchGetItem](#)」の「」を参照してください。

BatchWriteItem

次のコード例は、BatchWriteItem を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function dynamodb_batch_write_item
#
# This function writes a batch of items into a DynamoDB table.
#
# Parameters:
#     -i item -- Path to json file containing the items to write.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_batch_write_item() {
    local item response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_batch_write_item"
        echo "Write a batch of items into a DynamoDB table."
        echo " -i item -- Path to json file containing the items to write."
        echo ""
    }
    while getopt "i:h" option; do
        case "${option}" in
            i) item="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
```

```

        usage
        return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho "    item:  $item"
iecho ""

response=$(aws dynamodb batch-write-item \
    --request-items file://"${item}")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-write-item operation failed.$response"
    return 1
fi

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then

```



```
    echo "$@"
  fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
  elif [ "$err_code" == 255 ]; then
```

```
errecho " 255 is a catch-all error."
fi

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[BatchWriteItem](#)」の「」を参照してください。

CreateTable

次の例は、CreateTable を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください [GitHub](#)。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function dynamodb_create_table
#
# This function creates an Amazon DynamoDB table.
#
# Parameters:
#   -n table_name -- The name of the table to create.
#   -a attribute_definitions -- JSON file path of a list of attributes and their
#   types.
#   -k key_schema -- JSON file path of a list of attributes and their key types.
#   -p provisioned_throughput -- Provisioned throughput settings for the table.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_create_table() {
    local table_name attribute_definitions key_schema provisioned_throughput response
    local option OPTARG # Required to use getopt command in a function.
```

```
#####  
# Function usage explanation  
#####  
function usage() {  
    echo "function dynamodb_create_table"  
    echo "Creates an Amazon DynamoDB table."  
    echo " -n table_name -- The name of the table to create."  
    echo " -a attribute_definitions -- JSON file path of a list of attributes and  
their types."  
    echo " -k key_schema -- JSON file path of a list of attributes and their key  
types."  
    echo " -p provisioned_throughput -- Provisioned throughput settings for the  
table."  
    echo ""  
}  
  
# Retrieve the calling parameters.  
while getopts "n:a:k:p:h" option; do  
    case "${option}" in  
        n) table_name="${OPTARG}" ;;  
        a) attribute_definitions="${OPTARG}" ;;  
        k) key_schema="${OPTARG}" ;;  
        p) provisioned_throughput="${OPTARG}" ;;  
        h)  
            usage  
            return 0  
            ;;  
        \?)  
            echo "Invalid parameter"  
            usage  
            return 1  
            ;;  
    esac  
done  
export OPTIND=1  
  
if [[ -z "$table_name" ]]; then  
    errecho "ERROR: You must provide a table name with the -n parameter."  
    usage  
    return 1  
fi  
  
if [[ -z "$attribute_definitions" ]]; then
```

```
errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
usage
return 1
fi

if [[ -z "$key_schema" ]]; then
errecho "ERROR: You must provide a key schema json file path the -k parameter."
usage
return 1
fi

if [[ -z "$provisioned_throughput" ]]; then
errecho "ERROR: You must provide a provisioned throughput json file path the -p
parameter."
usage
return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho "  attribute_definitions:  $attribute_definitions"
iecho "  key_schema:  $key_schema"
iecho "  provisioned_throughput:  $provisioned_throughput"
iecho ""

response=$(aws dynamodb create-table \
  --table-name "$table_name" \
  --attribute-definitions file://"${attribute_definitions}" \
  --key-schema file://"${key_schema}" \
  --provisioned-throughput "$provisioned_throughput")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
aws_cli_error_log $error_code
errecho "ERROR: AWS reports create-table operation failed.$response"
return 1
fi

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
```

```

    errecho " Command line failed to parse."
elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateTable](#)」の「」を参照してください。

DeleteItem

次のコード例は、DeleteItem を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function dynamodb_delete_item
#
# This function deletes an item from a DynamoDB table.
#
# Parameters:
#     -n table_name  -- The name of the table.
#     -k keys       -- Path to json file containing the keys that identify the item to
delete.

```

```

#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_delete_item() {
    local table_name keys response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_delete_item"
        echo "Delete an item from a DynamoDB table."
        echo " -n table_name -- The name of the table."
        echo " -k keys -- Path to json file containing the keys that identify the item
to delete."
        echo ""
    }
    while getopt "n:k:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            k) keys="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$table_name" ]]; then
        errecho "ERROR: You must provide a table name with the -n parameter."
        usage
        return 1
    fi

    if [[ -z "$keys" ]]; then

```

```

    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho "    keys:  $keys"
iecho ""

response=$(aws dynamodb delete-item \
    --table-name "$table_name" \
    --key file://"${keys}")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-item operation failed.$response"
    return 1
fi

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho

```



```
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteItem](#)」の「」を参照してください。

DeleteTable

次のコード例は、DeleteTable を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function dynamodb_delete_table
#
# This function deletes a DynamoDB table.
#
# Parameters:
#     -n table_name  -- The name of the table to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_delete_table() {
    local table_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function dynamodb_delete_table"
        echo "Deletes an Amazon DynamoDB table."
        echo " -n table_name  -- The name of the table to delete."
        echo ""
    }
}
```

```
# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) table_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
  errecho "ERROR: You must provide a table name with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "  table_name:  $table_name"
iecho ""

response=$(aws dynamodb delete-table \
  --table-name "$table_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-table operation failed.$response"
  return 1
fi

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    fi
}
```

```

elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteTable](#)」の「」を参照してください。

DescribeTable

次のコード例は、DescribeTable を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function dynamodb_describe_table
#
# This function returns the status of a DynamoDB table.
#
# Parameters:
#     -n table_name  -- The name of the table.
#
# Response:
#     - TableStatus:

```

```

# And:
# 0 - Table is active.
# 1 - If it fails.
#####
function dynamodb_describe_table {
    local table_name
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_describe_table"
        echo "Describe the status of a DynamoDB table."
        echo " -n table_name -- The name of the table."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$table_name" ]]; then
        errecho "ERROR: You must provide a table name with the -n parameter."
        usage
        return 1
    fi

    local table_status
    table_status=$(
        aws dynamodb describe-table \

```

```

        --table-name "$table_name" \
        --output text \
        --query 'Table.TableStatus'
    )

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
        aws_cli_error_log "$error_code"
        errecho "ERROR: AWS reports describe-table operation failed.$table_status"
        return 1
    fi

    echo "$table_status"

    return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.

```

```
#
#####
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
  fi

  return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeTable](#)」の「」を参照してください。

GetItem

次のコード例は、GetItem を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
```



```

# function dynamodb_get_item
#
# This function gets an item from a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -k keys -- Path to json file containing the keys that identify the item to
get.
#     [-q query] -- Optional JMESPath query expression.
#
# Returns:
#     The item as text output.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_get_item() {
    local table_name keys query response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_get_item"
        echo "Get an item from a DynamoDB table."
        echo " -n table_name -- The name of the table."
        echo " -k keys -- Path to json file containing the keys that identify the item
to get."
        echo " [-q query] -- Optional JMESPath query expression."
        echo ""
    }
    query=""
    while getopt "n:k:q:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            k) keys="${OPTARG}" ;;
            q) query="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"

```

```
        usage
        return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

if [[ -n "$query" ]]; then
    response=$(aws dynamodb get-item \
        --table-name "$table_name" \
        --key file://"${keys}" \
        --output text \
        --query "$query")
else
    response=$(
        aws dynamodb get-item \
            --table-name "$table_name" \
            --key file://"${keys}" \
            --output text
    )
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports get-item operation failed.$response"
    return 1
fi

if [[ -n "$query" ]]; then
```

```

    echo "$response" | sed "/^\t/s/\t//1" # Remove initial tab that the JMSEPath
query inserts on some strings.
    else
        echo "$response"
    fi

    return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    fi
}

```

```

elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetItem](#)」の「」を参照してください。

ListTables

次の例は、ListTables を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function dynamodb_list_tables
#
# This function lists all the tables in a DynamoDB.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_list_tables() {

```

```

response=$(aws dynamodb list-tables \
  --output text \
  --query "TableNames")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports batch-write-item operation failed.$response"
  return 1
fi

echo "$response" | tr -s "[:space:]" "\n"

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#

```

```
#####  
function aws_cli_error_log() {  
    local err_code=$1  
    errecho "Error code : $err_code"  
    if [ "$err_code" == 1 ]; then  
        errecho " One or more S3 transfers failed."  
    elif [ "$err_code" == 2 ]; then  
        errecho " Command line failed to parse."  
    elif [ "$err_code" == 130 ]; then  
        errecho " Process received SIGINT."  
    elif [ "$err_code" == 252 ]; then  
        errecho " Command syntax invalid."  
    elif [ "$err_code" == 253 ]; then  
        errecho " The system environment or configuration was invalid."  
    elif [ "$err_code" == 254 ]; then  
        errecho " The service returned an error."  
    elif [ "$err_code" == 255 ]; then  
        errecho " 255 is a catch-all error."  
    fi  
  
    return 0  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListTables](#)」の「」を参照してください。

PutItem

次のコード例は、PutItem を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####  
# function dynamodb_put_item
```

```

#
# This function puts an item into a DynamoDB table.
#
# Parameters:
#   -n table_name  -- The name of the table.
#   -i item        -- Path to json file containing the item values.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function dynamodb_put_item() {
    local table_name item response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_put_item"
        echo "Put an item into a DynamoDB table."
        echo " -n table_name  -- The name of the table."
        echo " -i item        -- Path to json file containing the item values."
        echo ""
    }

    while getopt "n:i:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            i) item="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$table_name" ]]; then

```

```

    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho "    item:        $item"
iecho ""
iecho ""

response=$(aws dynamodb put-item \
    --table-name "$table_name" \
    --item file://" $item")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports put-item operation failed.$response"
    return 1
fi

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {

```



```
if [[ $VERBOSE == true ]]; then
    echo "$@"
fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    fi
}
```

```

elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutItem](#)」の「」を参照してください。

Query

次の例は、Query を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function dynamodb_query
#
# This function queries a DynamoDB table.
#
# Parameters:
#   -n table_name -- The name of the table.
#   -k key_condition_expression -- The key condition expression.
#   -a attribute_names -- Path to JSON file containing the attribute names.
#   -v attribute_values -- Path to JSON file containing the attribute values.
#   [-p projection_expression] -- Optional projection expression.
#
# Returns:
#   The items as json output.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#####

```

```
function dynamodb_query() {
  local table_name key_condition_expression attribute_names attribute_values
  projection_expression response
  local option OPTARG # Required to use getopt command in a function.

  # #####
  # Function usage explanation
  # #####
  function usage() {
    echo "function dynamodb_query"
    echo "Query a DynamoDB table."
    echo " -n table_name -- The name of the table."
    echo " -k key_condition_expression -- The key condition expression."
    echo " -a attribute_names -- Path to JSON file containing the attribute names."
    echo " -v attribute_values -- Path to JSON file containing the attribute
values."
    echo " [-p projection_expression] -- Optional projection expression."
    echo ""
  }

  while getopt "n:k:a:v:p:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      k) key_condition_expression="${OPTARG}" ;;
      a) attribute_names="${OPTARG}" ;;
      v) attribute_values="${OPTARG}" ;;
      p) projection_expression="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
  fi
}
```

```
fi

if [[ -z "$key_condition_expression" ]]; then
    errecho "ERROR: You must provide a key condition expression with the -k
parameter."
    usage
    return 1
fi

if [[ -z "$attribute_names" ]]; then
    errecho "ERROR: You must provide a attribute names with the -a parameter."
    usage
    return 1
fi

if [[ -z "$attribute_values" ]]; then
    errecho "ERROR: You must provide a attribute values with the -v parameter."
    usage
    return 1
fi

if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"$attribute_names" \
        --expression-attribute-values file://"$attribute_values")
else
    response=$(aws dynamodb query \
        --table-name "$table_name" \
        --key-condition-expression "$key_condition_expression" \
        --expression-attribute-names file://"$attribute_names" \
        --expression-attribute-values file://"$attribute_values" \
        --projection-expression "$projection_expression")
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports query operation failed.$response"
    return 1
fi
```

```
echo "$response"

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    }
}
```

```

elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、「コマンドリファレンス」の「[クエリ](#)」を参照してください。AWS CLI

Scan

次の例は、Scan を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function dynamodb_scan
#
# This function scans a DynamoDB table.
#
# Parameters:
#     -n table_name -- The name of the table.
#     -f filter_expression -- The filter expression.
#     -a expression_attribute_names -- Path to JSON file containing the expression
#     attribute names.
#     -v expression_attribute_values -- Path to JSON file containing the
#     expression attribute values.
#     [-p projection_expression] -- Optional projection expression.
#

```

```

# Returns:
#     The items as json output.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_scan() {
    local table_name filter_expression expression_attribute_names
    expression_attribute_values projection_expression response
    local option OPTARG # Required to use getopt command in a function.

    # #####
    # Function usage explanation
    # #####
    function usage() {
        echo "function dynamodb_scan"
        echo "Scan a DynamoDB table."
        echo " -n table_name -- The name of the table."
        echo " -f filter_expression -- The filter expression."
        echo " -a expression_attribute_names -- Path to JSON file containing the
expression attribute names."
        echo " -v expression_attribute_values -- Path to JSON file containing the
expression attribute values."
        echo " [-p projection_expression] -- Optional projection expression."
        echo ""
    }

    while getopt "n:f:a:v:p:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            f) filter_expression="${OPTARG}" ;;
            a) expression_attribute_names="${OPTARG}" ;;
            v) expression_attribute_values="${OPTARG}" ;;
            p) projection_expression="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}

```

```
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$filter_expression" ]]; then
    errecho "ERROR: You must provide a filter expression with the -f parameter."
    usage
    return 1
fi

if [[ -z "$expression_attribute_names" ]]; then
    errecho "ERROR: You must provide expression attribute names with the -a
parameter."
    usage
    return 1
fi

if [[ -z "$expression_attribute_values" ]]; then
    errecho "ERROR: You must provide expression attribute values with the -v
parameter."
    usage
    return 1
fi

if [[ -z "$projection_expression" ]]; then
    response=$(aws dynamodb scan \
        --table-name "$table_name" \
        --filter-expression "$filter_expression" \
        --expression-attribute-names file://"${expression_attribute_names}" \
        --expression-attribute-values file://"${expression_attribute_values}")
else
    response=$(aws dynamodb scan \
        --table-name "$table_name" \
        --filter-expression "$filter_expression" \
        --expression-attribute-names file://"${expression_attribute_names}" \
        --expression-attribute-values file://"${expression_attribute_values}" \
        --projection-expression "$projection_expression")
fi
```



```

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports scan operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
}

```

```
if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- API 詳細については、「コマンドリファレンス」の「[スキャン](#)」を参照してください。AWS CLI

UpdateItem

次の例は、UpdateItem を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function dynamodb_update_item
#
# This function updates an item in a DynamoDB table.
#
#
```

```

# Parameters:
#     -n table_name  -- The name of the table.
#     -k keys       -- Path to json file containing the keys that identify the item to
#                   update.
#     -e update expression  -- An expression that defines one or more attributes
#                   to be updated.
#     -v values     -- Path to json file containing the update values.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function dynamodb_update_item() {
    local table_name keys update_expression values response
    local option OPTARG # Required to use getopt command in a function.

    #####
    # Function usage explanation
    #####
    function usage() {
        echo "function dynamodb_update_item"
        echo "Update an item in a DynamoDB table."
        echo " -n table_name  -- The name of the table."
        echo " -k keys       -- Path to json file containing the keys that identify the item
to update."
        echo " -e update expression  -- An expression that defines one or more
attributes to be updated."
        echo " -v values     -- Path to json file containing the update values."
        echo ""
    }

    while getopt "n:k:e:v:h" option; do
        case "${option}" in
            n) table_name="${OPTARG}" ;;
            k) keys="${OPTARG}" ;;
            e) update_expression="${OPTARG}" ;;
            v) values="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage

```

```
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$table_name" ]]; then
    errecho "ERROR: You must provide a table name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$keys" ]]; then
    errecho "ERROR: You must provide a keys json file path the -k parameter."
    usage
    return 1
fi

if [[ -z "$update_expression" ]]; then
    errecho "ERROR: You must provide an update expression with the -e parameter."
    usage
    return 1
fi

if [[ -z "$values" ]]; then
    errecho "ERROR: You must provide a values json file path the -v parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    table_name:  $table_name"
iecho "    keys:        $keys"
iecho "    update_expression:  $update_expression"
iecho "    values:      $values"

response=$(aws dynamodb update-item \
    --table-name "$table_name" \
    --key file://" $keys" \
    --update-expression "$update_expression" \
    --expression-attribute-values file://" $values")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
```

```

    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports update-item operation failed.$response"
    return 1
fi

return 0

}

```

この例で使用されているユーティリティ関数。

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-return-codes.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.

```

```
#
# Returns:
#      0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [UpdateItem](#)」の「」を参照してください。

Bash スクリプト AWS CLI でを使用する Amazon EC2 の例

次のコード例は、Amazon で Bash スクリプト AWS Command Line Interface を使用してアクションを実行し、一般的なシナリオを実装する方法を示しています EC2。

「基本」は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [基本](#)
- [アクション](#)

基本

基本を学ぶ

次のコードサンプルは、以下の操作方法を示しています。

- キーペアとセキュリティグループを作成します。
- Amazon マシンイメージ (AMI) と互換性のあるインスタンスタイプを選択し、インスタンスを作成します。
- インスタンスを停止し、再起動します。
- Elastic IP アドレスをインスタンスに関連付ける。
- を使用してインスタンスに接続しSSH、リソースをクリーンアップします。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

コマンドプロンプトからインタラクティブのシナリオを実行します。

```
#####  
# function get_started_with_ec2_instances  
#  
# Runs an interactive scenario that shows how to get started using EC2 instances.  
#  
#     "EC2 access" permissions are needed to run this code.  
#  
# Returns:
```

```

#      0 - If successful.
#      1 - If an error occurred.
#####
function get_started_with_ec2_instances() {
  # Requires version 4 for mapfile.
  local required_version=4.0

  # Get the current Bash version
  # Check if BASH_VERSION is set
  local current_version
  if [[ -n "$BASH_VERSION" ]]; then
    # Convert BASH_VERSION to a number for comparison
    current_version=$BASH_VERSION
  else
    # Get the current Bash version using the bash command
    current_version=$(bash --version | head -n 1 | awk '{ print $4 }')
  fi

  # Convert version strings to numbers for comparison
  local required_version_num current_version_num
  required_version_num=$(echo "$required_version" | awk -F. '{ print ($1 * 10000) + ($2 * 100) + $3 }')
  current_version_num=$(echo "$current_version" | awk -F. '{ print ($1 * 10000) + ($2 * 100) + $3 }')

  # Compare versions
  if ((current_version_num < required_version_num)); then
    echo "Error: This script requires Bash version $required_version or higher."
    echo "Your current Bash version is number is $current_version."
    exit 1
  fi

  {
    if [ "$EC2_OPERATIONS_SOURCED" != "True" ]; then

      source ./ec2_operations.sh
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the Amazon Elastic Compute Cloud (Amazon EC2) get started with
instances demo."
  echo_repeat "*" 88
  echo

```



```
echo "Let's create an RSA key pair that you can be use to securely connect to "  
echo "your EC2 instance."  
  
echo -n "Enter a unique name for your key: "  
get_input  
local key_name  
key_name=${get_input_result}  
  
local temp_dir  
temp_dir=$(mktemp -d)  
local key_file_name="$temp_dir/${key_name}.pem"  
  
if ec2_create_keypair -n "${key_name}" -f "${key_file_name}"; then  
    echo "Created a key pair $key_name and saved the private key to $key_file_name"  
    echo  
else  
    errecho "The key pair failed to create. This demo will exit."  
    return 1  
fi  
  
chmod 400 "${key_file_name}"  
  
if yes_no_input "Do you want to list some of your key pairs? (y/n) "; then  
    local keys_and_fingerprints  
    keys_and_fingerprints="$(ec2_describe_key_pairs)" && {  
        local image_name_and_id  
        while IFS=$'\n' read -r image_name_and_id; do  
            local entries  
            IFS=$'\t' read -ra entries <<<"$image_name_and_id"  
            echo "Found rsa key ${entries[0]} with fingerprint:"  
            echo "    ${entries[1]}"  
        done <<<"$keys_and_fingerprints"  
    }  
fi  
  
echo_repeat "*" 88  
echo_repeat "*" 88  
  
echo "Let's create a security group to manage access to your instance."  
echo -n "Enter a unique name for your security group: "  
get_input  
local security_group_name
```

```
security_group_name=$(get_input_result)
local security_group_id
security_group_id=$(ec2_create_security_group -n "$security_group_name" \
-d "Security group for EC2 instance") || {
    errecho "The security failed to create. This demo will exit."
    clean_up "$key_name" "$key_file_name"
    return 1
}

echo "Security group created with ID $security_group_id"
echo

local public_ip
public_ip=$(curl -s http://checkip.amazonaws.com)

echo "Let's add a rule to allow SSH only from your current IP address."
echo "Your public IP address is $public_ip"
echo -n "press return to add this rule to your security group."
get_input

if ! ec2_authorize_security_group_ingress -g "$security_group_id" -i "$public_ip"
-p tcp -f 22 -t 22; then
    errecho "The security group rules failed to update. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
fi

echo "Security group rules updated"

local security_group_description
security_group_description="$(ec2_describe_security_groups -g
"${security_group_id}")" || {
    errecho "Failed to describe security groups. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

mapfile -t parameters <<<"$security_group_description"
IFS=$'\t' read -ra entries <<<"${parameters[0]}"
echo "Security group: ${entries[0]}"
echo "    ID: ${entries[1]}"
echo "    VPC: ${entries[2]}"
echo "Inbound permissions:"
IFS=$'\t' read -ra entries <<<"${parameters[1]}"
```

```

echo "    IpProtocol: ${entries[0]}"
echo "    FromPort: ${entries[1]}"
echo "    ToPort: ${entries[2]}"
echo "    CidrIp: ${parameters[2]}"

local parameters
parameters="$(ssm_get_parameters_by_path -p "/aws/service/ami-amazon-linux-
latest")" || {
    errecho "Failed to get parameters. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

local image_ids=""
mapfile -t parameters <<<"$parameters"
for image_name_and_id in "${parameters[@]"; do
    IFS=$'\t' read -ra values <<<"$image_name_and_id"
    if [[ "${values[0]}" == *"amzn2"* ]]; then
        image_ids+="${values[1]} "
    fi
done

local images
images="$(ec2_describe_images -i "$image_ids")" || {
    errecho "Failed to describe images. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

new_line_and_tab_to_list "$images"
local images=("${list_result[@]}")

# Get the size of the array
local images_count=${#images[@]}

if ((images_count == 0)); then
    errecho "No images found. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
fi

echo_repeat "*" 88

```

```
echo_repeat "*" 88

echo "Let's create an instance from an Amazon Linux 2 AMI. Here are some options:"
for ((i = 0; i < images_count; i += 3)); do
    echo "$(((i / 3) + 1)) - ${images[$i]}"
done

integer_input "Please enter the number of the AMI you want to use: " 1
"$((images_count / 3))"
local choice=$get_input_result
choice=$((choice - 1) * 3)

echo "Great choice."
echo

local architecture=${images[$((choice + 1))]}
local image_id=${images[$((choice + 2))]}
echo "Here are some instance types that support the ${architecture} architecture
of the image:"
response="$(ec2_describe_instance_types -a "${architecture}" -t
"*.micro,*.small")" || {
    errecho "Failed to describe instance types. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

local instance_types
mapfile -t instance_types <<<"$response"

# Get the size of the array
local instance_types_count=${#instance_types[@]}

echo "Here are some options:"
for ((i = 0; i < instance_types_count; i++)); do
    echo "$((i + 1)) - ${instance_types[$i]}"
done

integer_input "Which one do you want to use? " 1 "${#instance_types[@]}"
"
choice=$get_input_result
local instance_type=${instance_types[$((choice - 1))]}
echo "Another great choice."
echo
```

```
echo "Creating your instance and waiting for it to start..."
local instance_id
instance_id=$(ec2_run_instances -i "$image_id" -t "$instance_type" -k "$key_name"
-s "$security_group_id") || {
    errecho "Failed to run instance. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

ec2_wait_for_instance_running -i "$instance_id"
echo "Your instance is ready:"
echo

local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

echo
print_instance_details "${instance_details}"

local public_ip
public_ip=$(echo "${instance_details}" | awk '{print $6}')
echo
echo "You can use SSH to connect to your instance"
echo "If the connection attempt times out, you might have to manually update the
SSH ingress rule"
echo "for your IP address in the AWS Management Console."
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's stop and start your instance to see what changes."
echo "Stopping your instance and waiting until it's stopped..."
ec2_stop_instances -i "$instance_id"
ec2_wait_for_instance_stopped -i "$instance_id"

echo "Your instance is stopped. Restarting..."

ec2_start_instances -i "$instance_id"
ec2_wait_for_instance_running -i "$instance_id"
```

```
echo "Your instance is running again."
local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

print_instance_details "${instance_details}"

public_ip=$(echo "${instance_details}" | awk '{print $6}')

echo "Every time your instance is restarted, its public IP address changes"
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "You can allocate an Elastic IP address and associate it with your instance"
echo "to keep a consistent IP address even when your instance restarts."

local result
result=$(ec2_allocate_address -d vpc) || {
    errecho "Failed to allocate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    return 1
}

local elastic_ip allocation_id
elastic_ip=$(echo "$result" | awk '{print $1}')
allocation_id=$(echo "$result" | awk '{print $2}')

echo "Allocated static Elastic IP address: $elastic_ip"

local association_id
association_id=$(ec2_associate_address -i "$instance_id" -a "$allocation_id") || {
    errecho "Failed to associate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
"$allocation_id"
    return 1
}

echo "Associated your Elastic IP with your instance."
echo "You can now use SSH to connect to your instance by using the Elastic IP."
connect_to_instance "$key_file_name" "$elastic_ip"
```

```
echo -n "Press Enter when you're ready to continue the demo: "  
get_input  
  
echo_repeat "*" 88  
echo_repeat "*" 88  
  
echo "Let's stop and start your instance to see what changes."  
echo "Stopping your instance and waiting until it's stopped..."  
ec2_stop_instances -i "$instance_id"  
ec2_wait_for_instance_stopped -i "$instance_id"  
  
echo "Your instance is stopped. Restarting..."  
  
ec2_start_instances -i "$instance_id"  
ec2_wait_for_instance_running -i "$instance_id"  
  
echo "Your instance is running again."  
local instance_details  
instance_details="$(ec2_describe_instances -i "${instance_id}")"  
  
print_instance_details "${instance_details}"  
  
echo "Because you have associated an Elastic IP with your instance, you can"  
echo "connect by using a consistent IP address after the instance restarts."  
connect_to_instance "$key_file_name" "$elastic_ip"  
  
echo -n "Press Enter when you're ready to continue the demo: "  
get_input  
  
echo_repeat "*" 88  
echo_repeat "*" 88  
  
if yes_no_input "Do you want to delete the resources created in this demo: (y/n)"  
"; then  
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id" \  
        "$allocation_id" "$association_id"  
else  
    echo "The following resources were not deleted."  
    echo "Key pair: $key_name"  
    echo "Key file: $key_file_name"  
    echo "Security group: $security_group_id"  
    echo "Instance: $instance_id"  
    echo "Elastic IP address: $elastic_ip"
```

```
fi
}

#####
# function clean_up
#
# This function cleans up the created resources.
# $1 - The name of the ec2 key pair to delete.
# $2 - The name of the key file to delete.
# $3 - The ID of the security group to delete.
# $4 - The ID of the instance to terminate.
# $5 - The ID of the elastic IP address to release.
# $6 - The ID of the elastic IP address to disassociate.
#
# Returns:
# 0 - If successful.
# 1 - If an error occurred.
#####
function clean_up() {
    local result=0
    local key_pair_name=$1
    local key_file_name=$2
    local security_group_id=$3
    local instance_id=$4
    local allocation_id=$5
    local association_id=$6

    if [ -n "$association_id" ]; then
        # bashsupport disable=BP2002
        if (ec2_disassociate_address -a "$association_id"); then
            echo "Disassociated elastic IP address with ID $association_id"
        else
            errecho "The elastic IP address disassociation failed."
            result=1
        fi
    fi

    if [ -n "$allocation_id" ]; then
        # bashsupport disable=BP2002
        if (ec2_release_address -a "$allocation_id"); then
            echo "Released elastic IP address with ID $allocation_id"
        else
            errecho "The elastic IP address release failed."
            result=1
        fi
    fi
}
```



```
    fi
  fi

  if [ -n "$instance_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_terminate_instances -i "$instance_id"); then
      echo "Started terminating instance with ID $instance_id"

      ec2_wait_for_instance_terminated -i "$instance_id"
    else
      errecho "The instance terminate failed."
      result=1
    fi
  fi

  if [ -n "$security_group_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_security_group -i "$security_group_id"); then
      echo "Deleted security group with ID $security_group_id"
    else
      errecho "The security group delete failed."
      result=1
    fi
  fi

  if [ -n "$key_pair_name" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_keypair -n "$key_pair_name"); then
      echo "Deleted key pair named $key_pair_name"
    else
      errecho "The key pair delete failed."
      result=1
    fi
  fi

  if [ -n "$key_file_name" ]; then
    rm -f "$key_file_name"
  fi

  return $result
}

#####
# function ssm_get_parameters_by_path
```

```
#
# This function retrieves one or more parameters from the AWS Systems Manager
# Parameter Store
# by specifying a parameter path.
#
# Parameters:
#     -p parameter_path - The path of the parameter(s) to retrieve.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ssm_get_parameters_by_path() {
    local parameter_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ssm_get_parameters_by_path"
        echo "Retrieves one or more parameters from the AWS Systems Manager Parameter
Store by specifying a parameter path."
        echo "  -p parameter_path - The path of the parameter(s) to retrieve."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "p:h" option; do
        case "${option}" in
            p) parameter_path="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$parameter_path" ]]; then
        errecho "ERROR: You must provide a parameter path with the -p parameter."
    fi
}
```

```

    usage
    return 1
fi

response=$(aws ssm get-parameters-by-path \
  --path "$parameter_path" \
  --query "Parameters[*].[Name, Value]" \
  --output text) || {
  aws_cli_error_log $?
  errecho "ERROR: AWS reports get-parameters-by-path operation failed.$response"
  return 1
}

echo "$response"

return 0
}

#####
# function print_instance_details
#
# This function prints the details of an Amazon Elastic Compute Cloud (Amazon EC2)
# instance.
#
# Parameters:
#   instance_details - The instance details in the format "InstanceId ImageId
#   InstanceType KeyName VpcId PublicIpAddress State.Name".
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function print_instance_details() {
  local instance_details="$1"

  if [[ -z "${instance_details}" ]]; then
    echo "Error: Missing required instance details argument."
    return 1
  fi

  local instance_id image_id instance_type key_name vpc_id public_ip state
  instance_id=$(echo "${instance_details}" | awk '{print $1}')
  image_id=$(echo "${instance_details}" | awk '{print $2}')
  instance_type=$(echo "${instance_details}" | awk '{print $3}')

```

```

key_name=$(echo "${instance_details}" | awk '{print $4}')
vpc_id=$(echo "${instance_details}" | awk '{print $5}')
public_ip=$(echo "${instance_details}" | awk '{print $6}')
state=$(echo "${instance_details}" | awk '{print $7}')

echo "    ID: ${instance_id}"
echo "    Image ID: ${image_id}"
echo "    Instance type: ${instance_type}"
echo "    Key name: ${key_name}"
echo "    VPC ID: ${vpc_id}"
echo "    Public IP: ${public_ip}"
echo "    State: ${state}"

return 0
}

#####
# function connect_to_instance
#
# This function displays the public IP address of an Amazon Elastic Compute Cloud
# (Amazon EC2) instance and prompts the user to connect to the instance via SSH.
#
# Parameters:
#     $1 - The name of the key file used to connect to the instance.
#     $2 - The public IP address of the instance.
#
# Returns:
#     None
#####
function connect_to_instance() {
    local key_file_name="$1"
    local public_ip="$2"

    # Validate the input parameters
    if [[ -z "$key_file_name" ]]; then
        echo "ERROR: You must provide a key file name as the first argument." >&2
        return 1
    fi

    if [[ -z "$public_ip" ]]; then
        echo "ERROR: You must provide a public IP address as the second argument." >&2
        return 1
    fi
}

```

```
# Display the public IP address and connection command
echo "To connect, run the following command:"
echo "    ssh -i ${key_file_name} ec2-user@${public_ip}"

# Prompt the user to connect to the instance
if yes_no_input "Do you want to connect now? (y/n) "; then
    echo "After you have connected, you can return to this example by typing 'exit'"
    ssh -i "${key_file_name}" ec2-user@"${public_ip}"
fi
}

#####
# function get_input
#
# This function gets user input from the command line.
#
# Outputs:
#   User input to stdout.
#
# Returns:
#   0
#####
function get_input() {

    if [ -z "${mock_input+x}" ]; then
        read -r get_input_result
    else

        if [ "$mock_input_array_index" -lt ${#mock_input_array[@]} ]; then
            get_input_result="${mock_input_array[$mock_input_array_index]}"
            # bashsupport disable=BP2001
            # shellcheck disable=SC2206
            ((mock_input_array_index++))
            echo -n "$get_input_result"
        else
            echo "MOCK_INPUT_ARRAY has no more elements" 1>&2
            return 1
        fi
    fi

    return 0
}

#####
```

```
# function yes_no_input
#
# This function requests a yes/no answer from the user, following to a prompt.
#
# Parameters:
#     $1 - The prompt.
#
# Returns:
#     0 - If yes.
#     1 - If no.
#####
function yes_no_input() {
    if [ -z "$1" ]; then
        echo "Internal error yes_no_input"
        return 1
    fi

    local index=0
    local response="N"
    while [[ $index -lt 10 ]]; do
        index=$((index + 1))
        echo -n "$1"
        if ! get_input; then
            return 1
        fi
        response=$(echo "$get_input_result" | tr '[:upper:]' '[:lower:]')
        if [ "$response" = "y" ] || [ "$response" = "n" ]; then
            break
        else
            echo -e "\nPlease enter or 'y' or 'n'."
        fi
    done

    echo

    if [ "$response" = "y" ]; then
        return 0
    else
        return 1
    fi
}
#####
# function integer_input
```

```
#
# This function prompts the user to enter an integer within a specified range
# and validates the input.
#
# Parameters:
#     $1 - The prompt message to display to the user.
#     $2 - The minimum value of the accepted range.
#     $3 - The maximum value of the accepted range.
#
# Returns:
#     The valid integer input from the user.
#     If the input is invalid or out of range, the function will continue
#     prompting the user until a valid input is provided.
#####
function integer_input() {
    local prompt="$1"
    local min_value="$2"
    local max_value="$3"
    local input=""

    while true; do
        # Display the prompt message and wait for user input
        echo -n "$prompt"

        if ! get_input; then
            return 1
        fi

        input="$get_input_result"

        # Check if the input is a valid integer
        if [[ "$input" =~ ^-?[0-9]+$ ]]; then
            # Check if the input is within the specified range
            if ((input >= min_value && input <= max_value)); then
                return 0
            else
                echo "Error: Input, $input, must be between $min_value and $max_value."
            fi
        else
            echo "Error: Invalid input- $input. Please enter an integer."
        fi
    done
}
#####
```

```
# function new_line_and_tab_to_list
#
# This function takes a string input containing newlines and tabs, and
# converts it into a list (array) of elements.
#
# Parameters:
#     $1 - The input string containing newlines and tabs.
#
# Returns:
#     The resulting list (array) is stored in the global variable
#     'list_result'.
#####
function new_line_and_tab_to_list() {
    local input=$1
    export list_result

    list_result=()
    mapfile -t lines <<<"$input"
    local line
    for line in "${lines[@]"; do
        IFS=$'\t' read -ra parameters <<<"$line"
        list_result+=("${parameters[@]}")
    done
}

#####
# function echo_repeat
#
# This function prints a string 'n' times to stdout.
#
# Parameters:
#     $1 - The string.
#     $2 - Number of times to print the string.
#
# Outputs:
#     String 'n' times to stdout.
#
# Returns:
#     0
#####
function echo_repeat() {
    local end=$2
    for ((i = 0; i < end; i++)); do
        echo -n "$1"
    done
}
```



```
done
echo
}
```

このシナリオで使用される DynamoDB 関数。

```
#####
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
# 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#     -f file_path - File to store the key pair.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_create_keypair() {
    local key_pair_name file_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_create_keypair"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
RSA key pair"
        echo " and writes it to a file."
        echo "  -n key_pair_name - A key pair name."
        echo "  -f file_path - File to store the key pair."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:f:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;
            f) file_path="${OPTARG}" ;;
            h)

```

```
        usage
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
fi

response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
}

if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
fi

return 0
}

#####
# function ec2_describe_key_pairs
#
```

```

# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
pairs.
#
# Parameters:
#     -h - Display help.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_key_pairs() {
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_key_pairs"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
pairs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "h" option; do
        case "${option}" in
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local response

    response=$(aws ec2 describe-key-pairs \
        --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
        --output text) || {
        aws_cli_error_log ${?}
    }
}

```

```

    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
}

echo "$response"

return 0
}

#####
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#     -n security_group_name - The name of the security group.
#     -d security_group_description - The description of the security group.
#
# Returns:
#     The ID of the created security group, or an error message if the operation
#     fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_create_security_group() {
    local security_group_name security_group_description response

    # Function to display usage information
    function usage() {
        echo "function ec2_create_security_group"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -n security_group_name - The name of the security group."
        echo "  -d security_group_description - The description of the security group."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "n:d:h" option; do
        case "${option}" in
            n) security_group_name="${OPTARG}" ;;
            d) security_group_description="${OPTARG}" ;;
            h)

```

```
        usage
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$security_group_name" ]]; then
    errecho "ERROR: You must provide a security group name with the -n parameter."
    return 1
fi

if [[ -z "$security_group_description" ]]; then
    errecho "ERROR: You must provide a security group description with the -d
parameter."
    return 1
fi

# Create the security group
response=$(aws ec2 create-security-group \
    --group-name "$security_group_name" \
    --description "$security_group_description" \
    --query "GroupId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-security-group operation failed."
    errecho "$response"
    return 1
}

echo "$response"
return 0
}

#####
# function ec2_describe_security_groups
#
```

```

# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
security groups.
#
# Parameters:
#     -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_security_groups() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_security_groups"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
groups."
        echo "  -g security_group_id - The ID of the security group to describe
(optional)."

```

```

    if [[ -n "$security_group_id" ]]; then
        response=$(aws ec2 describe-security-groups --group-ids "$security_group_id" --
query "${query}" --output text)
    else
        response=$(aws ec2 describe-security-groups --query "${query}" --output text)
    fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports describe-security-groups operation failed.$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
(Amazon EC2) security group.
#
# Parameters:
#     -g security_group_id - The ID of the security group.
#     -i ip_address - The IP address or CIDR block to authorize.
#     -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#     -f from_port - The start of the port range to authorize.
#     -t to_port - The end of the port range to authorize.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_authorize_security_group_ingress() {
    local security_group_id ip_address protocol from_port to_port response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function ec2_authorize_security_group_ingress"

```

```
    echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
EC2) security group."
    echo "  -g security_group_id - The ID of the security group."
    echo "  -i ip_address - The IP address or CIDR block to authorize."
    echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
    echo "  -f from_port - The start of the port range to authorize."
    echo "  -t to_port - The end of the port range to authorize."
    echo ""
}

# Retrieve the calling parameters.
while getopts "g:i:p:f:t:h" option; do
    case "${option}" in
        g) security_group_id="${OPTARG}" ;;
        i) ip_address="${OPTARG}" ;;
        p) protocol="${OPTARG}" ;;
        f) from_port="${OPTARG}" ;;
        t) to_port="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -g parameter."
    usage
    return 1
fi

if [[ -z "$ip_address" ]]; then
    errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
    usage
    return 1
fi
```



```
if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
fi

if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
fi

if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
fi

response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
    return 1
}

return 0
}

#####
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# images.
#
# Parameters:
#     -i image_ids - A space-separated list of image IDs (optional).
#     -h - Display help.
#
# And:
```

```

#      0 - If successful.
#      1 - If it fails.
#####
function ec2_describe_images() {
    local image_ids response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopt "i:h" option; do
    case "${option}" in
        i) image_ids="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local aws_cli_args=()

if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
fi

response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {

```

```

aws_cli_error_log ${?}
errecho "ERROR: AWS reports describe-images operation failed.$response"
return 1
}

echo "$response"

return 0
}

#####
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE      Comma-separated list of instance types (e.g.,
# t2.micro)
# -h, --help                    Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
#####
function ec2_describe_instance_types() {
    local architecture=""
    local instance_types=""

    # bashsupport disable=BP5008
    function usage() {
        echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
        echo "  -a, --architecture ARCHITECTURE Specify the processor architecture
(e.g., x86_64)"
        echo "  -t, --type INSTANCE_TYPE      Comma-separated list of instance types
(e.g., t2.micro)"
        echo "  -h, --help                    Show this help message"
    }

    while [[ $# -gt 0 ]]; do
        case "$1" in
            -a | --architecture)
                architecture="$2"
                shift 2

```

```
    ;;
    -t | --type)
        instance_types="$2"
        shift 2
        ;;
    -h | --help)
        usage
        return 0
        ;;
    *)
        echo "Unknown argument: $1"
        return 1
        ;;
esac
done

if [[ -z "$architecture" ]]; then
    errecho "Error: Architecture not specified."
    usage
    return 1
fi

if [[ -z "$instance_types" ]]; then
    errecho "Error: Instance type not specified."
    usage
    return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '['
{
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '"${items[$i]}"' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ',' >>"$tmp_json_file"
    fi
done
```

```

echo -n ']],
{
  "Name": "instance-type",
  "Values": [' >>"$tmp_json_file"
IFS=', ' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '""${items[$i]}""' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ', ' >>"$tmp_json_file"
  fi
done

echo -n ']]]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://"${tmp_json_file}" \
  --query 'InstanceTypes[*].[InstanceType]' --output text)

local error_code=$?

rm "$tmp_json_file"

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  echo "ERROR: AWS reports describe-instance-types operation failed."
  return 1
fi

echo "$response"
return 0
}

#####
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#   -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#   -t instance_type - The instance type to use (e.g., t2.micro).
#   -k key_pair_name - The name of the key pair to use.

```

```

# -s security_group_id - The ID of the security group to use.
# -c count - The number of instances to launch (default: 1).
# -h - Display help.
#
# Returns:
# 0 - If successful.
# 1 - If it fails.
#####
function ec2_run_instances() {
    local image_id instance_type key_pair_name security_group_id count response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_run_instances"
        echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo " -i image_id - The ID of the Amazon Machine Image (AMI) to use."
        echo " -t instance_type - The instance type to use (e.g., t2.micro)."
        echo " -k key_pair_name - The name of the key pair to use."
        echo " -s security_group_id - The ID of the security group to use."
        echo " -c count - The number of instances to launch (default: 1)."
        echo " -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:t:k:s:c:h" option; do
        case "${option}" in
            i) image_id="${OPTARG}" ;;
            t) instance_type="${OPTARG}" ;;
            k) key_pair_name="${OPTARG}" ;;
            s) security_group_id="${OPTARG}" ;;
            c) count="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}

```

```
export OPTIND=1

if [[ -z "$image_id" ]]; then
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
    usage
    return 1
fi

if [[ -z "$instance_type" ]]; then
    errecho "ERROR: You must provide an instance type with the -t parameter."
    usage
    return 1
fi

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -k parameter."
    usage
    return 1
fi

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
fi

if [[ -z "$count" ]]; then
    count=1
fi

response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
}
```

```
    echo "$response"

    return 0
}

#####
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances.
#
# Parameters:
#     -i instance_id - The ID of the instance to describe (optional).
#     -q query - The query to filter the response (optional).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_instances() {
    local instance_id query response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_instances"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID of the instance to describe (optional).\"
        echo "  -q query - The query to filter the response (optional).\"
        echo "  -h - Display help.\"
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:q:h" option; do
        case "${option}" in
            i) instance_id="${OPTARG}" ;;
            q) query="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
        esac
    done
}
```



```

    \?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
  esac
done
export OPTIND=1

local aws_cli_args=()

if [[ -n "$instance_id" ]]; then
  # shellcheck disable=SC2206
  aws_cli_args+=("--instance-ids" $instance_id)
fi

local query_arg=""
if [[ -n "$query" ]]; then
  query_arg="--query '$query'"
else
  query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
fi

# shellcheck disable=SC2086
response=$(aws ec2 describe-instances \
  "${aws_cli_args[@]}" \
  $query_arg \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports describe-instances operation failed.$response"
  return 1
}

echo "$response"

return 0
}

#####
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances.

```

```
#
# Parameters:
#   -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_stop_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_stop_instances"
        echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$instance_ids" ]]; then
        errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
        usage
    fi
}
```

```

    return 1
fi

response=$(aws ec2 stop-instances \
  --instance-ids "${instance_ids}") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports stop-instances operation failed with $response."
  return 1
}

return 0
}

#####
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#   -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_start_instances() {
  local instance_ids
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_start_instances"
    echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopt "i:h" option; do
    case "${option}" in

```

```

    i) instance_ids="${OPTARG}" ;;
    h)
        usage
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
    usage
    return 1
fi

response=$(aws ec2 start-instances \
--instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
}

return 0
}

#####
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
# Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#     -d domain - The domain for the Elastic IP address (either 'vpc' or
# 'standard').
#
# Returns:
#     The allocated Elastic IP address, or an error message if the operation
# fails.

```

```

# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_allocate_address() {
    local domain response

    # Function to display usage information
    function usage() {
        echo "function ec2_allocate_address"
        echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
(Amazon EC2) instances in a specific AWS Region."
        echo " -d domain - The domain for the Elastic IP address (either 'vpc' or
'standard')."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "d:h" option; do
        case "${option}" in
            d) domain="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$domain" ]]; then
        errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
'standard')."
        return 1
    fi

    if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
        errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    fi
}

```

```

    return 1
fi

# Allocate the Elastic IP address
response=$(aws ec2 allocate-address \
  --domain "$domain" \
  --query "[PublicIp,AllocationId]" \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports allocate-address operation failed."
  errecho "$response"
  return 1
}

echo "$response"
return 0
}

#####
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
# Cloud (Amazon EC2) instance.
#
# Parameters:
#   -a allocation_id - The allocation ID of the Elastic IP address to associate.
#   -i instance_id - The ID of the EC2 instance to associate the Elastic IP
# address with.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_associate_address() {
  local allocation_id instance_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_associate_address"
    echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
associate."
  }

```

```
    echo " -i instance_id - The ID of the EC2 instance to associate the Elastic IP
address with."
    echo ""
}

# Parse the command-line arguments
while getopts "a:i:h" option; do
    case "${option}" in
        a) allocation_id="${OPTARG}" ;;
        i) instance_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
fi

if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
fi

# Associate the Elastic IP address
response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
}
```

```
}

echo "$response"
return 0
}

#####
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
  Cloud (Amazon EC2) instance.
#
# Parameters:
#   -a association_id - The association ID that represents the association of
  the Elastic IP address with an instance.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_disassociate_address() {
  local association_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_disassociate_address"
    echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
  (Amazon EC2) instance."
    echo "  -a association_id - The association ID that represents the association
  of the Elastic IP address with an instance."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) association_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"

```



```

        usage
        return 1
        ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$association_id" ]]; then
    errecho "ERROR: You must provide an association ID with the -a parameter."
    return 1
fi

response=$(aws ec2 disassociate-address \
    --association-id "$association_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports disassociate-address operation failed."
    errecho "$response"
    return 1
}

return 0
}

#####
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
# (Amazon EC2) instance.
#
# Parameters:
#     -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_release_address() {
    local allocation_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_release_address"
    }
}

```

```
    echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
release."
    echo ""
}

# Parse the command-line arguments
while getopts "a:h" option; do
    case "${option}" in
        a) allocation_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
fi

response=$(aws ec2 release-address \
--allocation-id "$allocation_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports release-address operation failed."
    errecho "$response"
    return 1
}

return 0
}

#####
# function ec2_terminate_instances
#
```

```
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#     -i instance_ids - A space-separated list of instance IDs.
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_terminate_instances() {
    local instance_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_terminate_instances"
        echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_ids - A space-separated list of instance IDs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Check if instance ID is provided
    if [[ -z "${instance_ids}" ]]; then
```

```

    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
fi

# shellcheck disable=SC2086
response=$(aws ec2 terminate-instances \
  "--instance-ids" $instance_ids \
  --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports terminate-instances operation failed.$response"
  return 1
}

return 0
}

#####
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#   -i security_group_id - The ID of the security group to delete.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_delete_security_group() {
  local security_group_id response
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_delete_security_group"
    echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -i security_group_id - The ID of the security group to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopt "i:h" option; do

```

```

    case "${option}" in
        i) security_group_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
}

return 0
}

#####
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_keypair() {

```

```
local key_pair_name response

local option OPTARG # Required to use getopt command in a function.
# bashsupport disable=BP5008
function usage() {
    echo "function ec2_delete_keypair"
    echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
    echo "  -n key_pair_name - A key pair name."
    echo ""
}

# Retrieve the calling parameters.
while getopt "n:h" option; do
    case "${option}" in
        n) key_pair_name="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -n parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-key-pair \
    --key-name "$key_pair_name") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
    return 1
}

return 0
}
```

このシナリオで使用されるユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi
}
```

```
    return 0  
}
```

- API 詳細については、AWS CLI コマンドリファレンスの以下のトピックを参照してください。
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)
 - [DeleteKeyPair](#)
 - [DeleteSecurityGroup](#)
 - [DescribeImages](#)
 - [DescribeInstanceTypes](#)
 - [DescribeInstances](#)
 - [DescribeKeyPairs](#)
 - [DescribeSecurityGroups](#)
 - [DisassociateAddress](#)
 - [ReleaseAddress](#)
 - [RunInstances](#)
 - [StartInstances](#)
 - [StopInstances](#)
 - [TerminateInstances](#)
 - [UnmonitorInstances](#)

アクション

AllocateAddress

次の例は、AllocateAddress を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
# Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#     -d domain - The domain for the Elastic IP address (either 'vpc' or
# 'standard').
#
# Returns:
#     The allocated Elastic IP address, or an error message if the operation
# fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_allocate_address() {
    local domain response

    # Function to display usage information
    function usage() {
        echo "function ec2_allocate_address"
        echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
        (Amazon EC2) instances in a specific AWS Region."
        echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
        'standard')."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "d:h" option; do
        case "${option}" in
```

```
d) domain="${OPTARG}" ;;
h)
    usage
    return 0
    ;;
\?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
'standard')."
    return 1
fi

if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
fi

# Allocate the Elastic IP address
response=$(aws ec2 allocate-address \
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
}

echo "$response"
return 0
}
```

この例で使用されているユーティリティ関数。

```
#####  
# function errecho  
#  
# This function outputs everything sent to it to STDERR (standard error output).  
#####  
function errecho() {  
    printf "%s\n" "$*" 1>&2  
}  
  
#####  
# function aws_cli_error_log()  
#  
# This function is used to log the error messages from the AWS CLI.  
#  
# The function expects the following argument:  
#     $1 - The error code returned by the AWS CLI.  
#  
# Returns:  
#     0: - Success.  
#  
#####  
function aws_cli_error_log() {  
    local err_code=$1  
    errecho "Error code : $err_code"  
    if [ "$err_code" == 1 ]; then  
        errecho " One or more S3 transfers failed."  
    elif [ "$err_code" == 2 ]; then  
        errecho " Command line failed to parse."  
    elif [ "$err_code" == 130 ]; then  
        errecho " Process received SIGINT."  
    elif [ "$err_code" == 252 ]; then  
        errecho " Command syntax invalid."  
    elif [ "$err_code" == 253 ]; then  
        errecho " The system environment or configuration was invalid."  
    elif [ "$err_code" == 254 ]; then  
        errecho " The service returned an error."  
    elif [ "$err_code" == 255 ]; then  
        errecho " 255 is a catch-all error."  
    fi  
  
    return 0  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AllocateAddress](#)」の「」を参照してください。

AssociateAddress

次のコード例は、AssociateAddress を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
# Cloud (Amazon EC2) instance.
#
# Parameters:
#   -a allocation_id - The allocation ID of the Elastic IP address to associate.
#   -i instance_id - The ID of the EC2 instance to associate the Elastic IP
# address with.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_associate_address() {
    local allocation_id instance_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_associate_address"
        echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo "  -a allocation_id - The allocation ID of the Elastic IP address to
associate."
    }
}
```

```
    echo " -i instance_id - The ID of the EC2 instance to associate the Elastic IP
address with."
    echo ""
}

# Parse the command-line arguments
while getopts "a:i:h" option; do
    case "${option}" in
        a) allocation_id="${OPTARG}" ;;
        i) instance_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
fi

if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
fi

# Associate the Elastic IP address
response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
}
```

```
}

echo "$response"
return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    }
}
```

```
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssociateAddress](#)」の「」を参照してください。

AuthorizeSecurityGroupIngress

次のコード例は、AuthorizeSecurityGroupIngress を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
# (Amazon EC2) security group.
#
# Parameters:
#   -g security_group_id - The ID of the security group.
#   -i ip_address - The IP address or CIDR block to authorize.
#   -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#   -f from_port - The start of the port range to authorize.
#   -t to_port - The end of the port range to authorize.
#
# And:
#   0 - If successful.
#   1 - If it fails.
```

```
#####  
function ec2_authorize_security_group_ingress() {  
    local security_group_id ip_address protocol from_port to_port response  
    local option OPTARG # Required to use getopt command in a function.  
  
    # bashsupport disable=BP5008  
    function usage() {  
        echo "function ec2_authorize_security_group_ingress"  
        echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon  
EC2) security group."  
        echo " -g security_group_id - The ID of the security group."  
        echo " -i ip_address - The IP address or CIDR block to authorize."  
        echo " -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."  
        echo " -f from_port - The start of the port range to authorize."  
        echo " -t to_port - The end of the port range to authorize."  
        echo ""  
    }  
  
    # Retrieve the calling parameters.  
    while getopt "g:i:p:f:t:h" option; do  
        case "${option}" in  
            g) security_group_id="${OPTARG}" ;;  
            i) ip_address="${OPTARG}" ;;  
            p) protocol="${OPTARG}" ;;  
            f) from_port="${OPTARG}" ;;  
            t) to_port="${OPTARG}" ;;  
            h)  
                usage  
                return 0  
                ;;  
            \?)  
                echo "Invalid parameter"  
                usage  
                return 1  
                ;;  
        esac  
    done  
    export OPTIND=1  
  
    if [[ -z "$security_group_id" ]]; then  
        errecho "ERROR: You must provide a security group ID with the -g parameter."  
        usage  
        return 1  
    fi
```



```
if [[ -z "$ip_address" ]]; then
    errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
    usage
    return 1
fi

if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
fi

if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
fi

if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
fi

response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation failed.
$response"
    return 1
}

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AuthorizeSecurityGroupIngress](#)」の「」を参照してください。

CreateKeyPair

次のコード例は、CreateKeyPair を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
# 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#     -f file_path - File to store the key pair.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_create_keypair() {
    local key_pair_name file_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_create_keypair"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit
        RSA key pair"
        echo " and writes it to a file."
        echo "  -n key_pair_name - A key pair name."
        echo "  -f file_path - File to store the key pair."
    }
}
```

```
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:f:h" option; do
    case "${option}" in
        n) key_pair_name="${OPTARG}" ;;
        f) file_path="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
fi

response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
}

if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
```

```
fi

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
```

```

    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateKeyPair](#)」の「」を参照してください。

CreateSecurityGroup

次の例は、CreateSecurityGroup を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#     -n security_group_name - The name of the security group.
#     -d security_group_description - The description of the security group.
#
# Returns:
#     The ID of the created security group, or an error message if the operation
#     fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####

```

```
function ec2_create_security_group() {
    local security_group_name security_group_description response

    # Function to display usage information
    function usage() {
        echo "function ec2_create_security_group"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -n security_group_name - The name of the security group."
        echo "  -d security_group_description - The description of the security group."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "n:d:h" option; do
        case "${option}" in
            n) security_group_name="${OPTARG}" ;;
            d) security_group_description="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$security_group_name" ]]; then
        errecho "ERROR: You must provide a security group name with the -n parameter."
        return 1
    fi

    if [[ -z "$security_group_description" ]]; then
        errecho "ERROR: You must provide a security group description with the -d
parameter."
        return 1
    fi

    # Create the security group
    response=$(aws ec2 create-security-group \
```

```

--group-name "$security_group_name" \
--description "$security_group_description" \
--query "GroupId" \
--output text) || {
aws_cli_error_log ${?}
errecho "ERROR: AWS reports create-security-group operation failed."
errecho "$response"
return 1
}

echo "$response"
return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    fi
}

```



```

elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateSecurityGroup](#)」の「」を参照してください。

DeleteKeyPair

次のコード例は、DeleteKeyPair を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#

```

```
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_keypair() {
    local key_pair_name response

    local option OPTARG # Required to use getopt command in a function.
    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_delete_keypair"
        echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
        echo "  -n key_pair_name - A key pair name."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$key_pair_name" ]]; then
        errecho "ERROR: You must provide a key pair name with the -n parameter."
        usage
        return 1
    fi

    response=$(aws ec2 delete-key-pair \
        --key-name "$key_pair_name") || {
        aws_cli_error_log ${?}
        errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
        return 1
    }
}
```

```

}

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then

```

```

    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteKeyPair](#)」の「」を参照してください。

DeleteSecurityGroup

次のコード例は、DeleteSecurityGroup を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Parameters:
#     -i security_group_id - The ID of the security group to delete.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_security_group() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008

```

```
function usage() {
    echo "function ec2_delete_security_group"
    echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
    echo "  -i security_group_id - The ID of the security group to delete."
    echo ""
}

# Retrieve the calling parameters.
while getopts "i:h" option; do
    case "${option}" in
        i) security_group_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
}

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteSecurityGroup](#)」の「」を参照してください。

DescribeImages

次のコード例は、DescribeImages を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# images.
#
# Parameters:
#   -i image_ids - A space-separated list of image IDs (optional).
#   -h - Display help.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_describe_images() {
    local image_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_images"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
        echo "  -i image_ids - A space-separated list of image IDs (optional)."
        echo "  -h - Display help."
        echo ""
    }
}
```

```

# Retrieve the calling parameters.
while getopts "i:h" option; do
  case "${option}" in
    i) image_ids="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

local aws_cli_args=()

if [[ -n "$image_ids" ]]; then
  # shellcheck disable=SC2206
  aws_cli_args+=("--image-ids" $image_ids)
fi

response=$(aws ec2 describe-images \
  "${aws_cli_args[@]}" \
  --query 'Images[*].[Description,Architecture,ImageId]' \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports describe-images operation failed.$response"
  return 1
}

echo "$response"

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho

```



```
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeImages](#)」の「」を参照してください。

DescribeInstanceTypes

次の例は、DescribeInstanceTypes を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE       Comma-separated list of instance types (e.g.,
#                                 t2.micro)
# -h, --help                       Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
#####
function ec2_describe_instance_types() {
    local architecture=""
    local instance_types=""

    # bashsupport disable=BP5008
    function usage() {
        echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
        echo "  -a, --architecture ARCHITECTURE Specify the processor architecture
(e.g., x86_64)"
        echo "  -t, --type INSTANCE_TYPE       Comma-separated list of instance types
(e.g., t2.micro)"
        echo "  -h, --help                       Show this help message"
    }
}
```

```
}

while [[ $# -gt 0 ]]; do
  case "$1" in
    -a | --architecture)
      architecture="$2"
      shift 2
      ;;
    -t | --type)
      instance_types="$2"
      shift 2
      ;;
    -h | --help)
      usage
      return 0
      ;;
    *)
      echo "Unknown argument: $1"
      return 1
      ;;
  esac
done

if [[ -z "$architecture" ]]; then
  errecho "Error: Architecture not specified."
  usage
  return 1
fi

if [[ -z "$instance_types" ]]; then
  errecho "Error: Instance type not specified."
  usage
  return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '[
  {
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
```

```
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '""${items[$i]}""' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ',' >>"$tmp_json_file"
    fi
done
echo -n ']],
{
    "Name": "instance-type",
    "Values": [' >>"$tmp_json_file"
IFS=', ' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '""${items[$i]}""' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ',' >>"$tmp_json_file"
    fi
done

echo -n ']]]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://" $tmp_json_file" \
    --query 'InstanceTypes[*].[InstanceType]' --output text)

local error_code=$?

rm "$tmp_json_file"

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    echo "ERROR: AWS reports describe-instance-types operation failed."
    return 1
fi

echo "$response"
return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstanceTypes](#)」の「」を参照してください。

DescribeInstances

次のコード例は、DescribeInstances を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#   -i instance_id - The ID of the instance to describe (optional).
#   -q query - The query to filter the response (optional).
#   -h - Display help.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_describe_instances() {
    local instance_id query response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_instances"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID of the instance to describe (optional).\"
        echo "  -q query - The query to filter the response (optional).\"
        echo "  -h - Display help.\"
    }
}
```

```
    echo ""
}

# Retrieve the calling parameters.
while getopts "i:q:h" option; do
    case "${option}" in
        i) instance_id="${OPTARG}" ;;
        q) query="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local aws_cli_args=()

if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
fi

local query_arg=""
if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
fi

# shellcheck disable=SC2086
response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
}
```

```
}

echo "$response"

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
```



```

    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeInstances](#)」の「」を参照してください。

DescribeKeyPairs

次のコード例は、DescribeKeyPairs を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
# pairs.
#
# Parameters:
#     -h - Display help.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_key_pairs() {
    local option OPTARG # Required to use getopt command in a function.

```

```
# bashsupport disable=BP5008
function usage() {
    echo "function ec2_describe_key_pairs"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
pairs."
    echo " -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopts "h" option; do
    case "${option}" in
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local response

response=$(aws ec2 describe-key-pairs \
    --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
    return 1
}

echo "$response"

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DescribeKeyPairs](#)」の「」を参照してください。

DescribeSecurityGroups

次の例は、DescribeSecurityGroups を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_describe_security_groups
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# security groups.
#
# Parameters:
#     -g security_group_id - The ID of the security group to describe (optional).
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_security_groups() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_security_groups"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) security
groups."
        echo "  -g security_group_id - The ID of the security group to describe
(optional)."
        echo ""
    }
}
```

```
# Retrieve the calling parameters.
while getopts "g:h" option; do
  case "${option}" in
    g) security_group_id="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"

if [[ -n "$security_group_id" ]]; then
  response=$(aws ec2 describe-security-groups --group-ids "$security_group_id" --
query "${query}" --output text)
else
  response=$(aws ec2 describe-security-groups --query "${query}" --output text)
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports describe-security-groups operation failed.$response"
  return 1
fi

echo "$response"

return 0
}
```

この例で使用されているユーティリティ関数。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DescribeSecurityGroups](#)」の「」を参照してください。

DisassociateAddress

次の例は、DisassociateAddress を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
# Cloud (Amazon EC2) instance.
#
# Parameters:
#     -a association_id - The association ID that represents the association of
#     the Elastic IP address with an instance.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_disassociate_address() {
    local association_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_disassociate_address"
        echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
        (Amazon EC2) instance."
        echo "  -a association_id - The association ID that represents the association
        of the Elastic IP address with an instance."
        echo ""
    }
}
```

```

# Parse the command-line arguments
while getopts "a:h" option; do
  case "${option}" in
    a) association_id="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$association_id" ]]; then
  errecho "ERROR: You must provide an association ID with the -a parameter."
  return 1
fi

response=$(aws ec2 disassociate-address \
  --association-id "$association_id") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports disassociate-address operation failed."
  errecho "$response"
  return 1
}

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####

```



```
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DisassociateAddress](#)」の「」を参照してください。

ReleaseAddress

次の例は、ReleaseAddress を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
  (Amazon EC2) instance.
#
# Parameters:
#   -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_release_address() {
    local allocation_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_release_address"
        echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
  (Amazon EC2) instance."
        echo "  -a allocation_id - The allocation ID of the Elastic IP address to
  release."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:h" option; do
        case "${option}" in
            a) allocation_id="${OPTARG}" ;;

```

```

    h)
    usage
    return 0
    ;;
    \(?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
  esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
  errecho "ERROR: You must provide an allocation ID with the -a parameter."
  return 1
fi

response=$(aws ec2 release-address \
  --allocation-id "$allocation_id") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports release-address operation failed."
  errecho "$response"
  return 1
}

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####

```

```
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi


    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ReleaseAddress](#)」の「」を参照してください。

RunInstances

次の例は、RunInstances を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#     -t instance_type - The instance type to use (e.g., t2.micro).
#     -k key_pair_name - The name of the key pair to use.
#     -s security_group_id - The ID of the security group to use.
#     -c count - The number of instances to launch (default: 1).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_run_instances() {
    local image_id instance_type key_pair_name security_group_id count response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_run_instances"
        echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
        echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
        echo "  -k key_pair_name - The name of the key pair to use."
        echo "  -s security_group_id - The ID of the security group to use."
        echo "  -c count - The number of instances to launch (default: 1)."
        echo "  -h - Display help."
        echo ""
    }
}
```

```
# Retrieve the calling parameters.
while getopts "i:t:k:s:c:h" option; do
  case "${option}" in
    i) image_id="${OPTARG}" ;;
    t) instance_type="${OPTARG}" ;;
    k) key_pair_name="${OPTARG}" ;;
    s) security_group_id="${OPTARG}" ;;
    c) count="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$image_id" ]]; then
  errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
  usage
  return 1
fi

if [[ -z "$instance_type" ]]; then
  errecho "ERROR: You must provide an instance type with the -t parameter."
  usage
  return 1
fi

if [[ -z "$key_pair_name" ]]; then
  errecho "ERROR: You must provide a key pair name with the -k parameter."
  usage
  return 1
fi

if [[ -z "$security_group_id" ]]; then
  errecho "ERROR: You must provide a security group ID with the -s parameter."
  usage
```

```

    return 1
fi

if [[ -z "$count" ]]; then
    count=1
fi

response=$(aws ec2 run-instances \
  --image-id "$image_id" \
  --instance-type "$instance_type" \
  --key-name "$key_pair_name" \
  --security-group-ids "$security_group_id" \
  --count "$count" \
  --query 'Instances[*].[InstanceId]' \
  --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
}

echo "$response"

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:

```

```
#      $1 - The error code returned by the AWS CLI.
#
# Returns:
#      0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[RunInstances](#)」の「」を参照してください。

StartInstances

次のコード例は、StartInstances を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances.
#
# Parameters:
#     -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_start_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_start_instances"
        echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
        esac
    done
}
```

```

        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
    usage
    return 1
fi

response=$(aws ec2 start-instances \
--instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
}

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#

```

```
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StartInstances](#)」の「」を参照してください。

StopInstances

次のコード例は、StopInstances を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_stop_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_stop_instances"
        echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
        esac
    done
}
```

```

        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
    usage
    return 1
fi

response=$(aws ec2 stop-instances \
--instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports stop-instances operation failed with $response."
    return 1
}

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#

```

```
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi


    return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[StopInstances](#)」の「」を参照してください。

TerminateInstances

次のコード例は、TerminateInstances を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#     -i instance_ids - A space-separated list of instance IDs.
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_terminate_instances() {
    local instance_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_terminate_instances"
        echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_ids - A space-separated list of instance IDs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage

```

```

        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

# Check if instance ID is provided
if [[ -z "${instance_ids}" ]]; then
    echo "Error: Missing required instance IDs parameter."
    usage
    return 1
fi

# shellcheck disable=SC2086
response=$(aws ec2 terminate-instances \
    "--instance-ids" $instance_ids \
    --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports terminate-instances operation failed.$response"
    return 1
}

return 0
}

```

この例で使用されているユーティリティ関数。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```



```
#####  
# function aws_cli_error_log()  
#  
# This function is used to log the error messages from the AWS CLI.  
#  
# The function expects the following argument:  
#     $1 - The error code returned by the AWS CLI.  
#  
# Returns:  
#     0: - Success.  
#  
#####  
function aws_cli_error_log() {  
    local err_code=$1  
    errecho "Error code : $err_code"  
    if [ "$err_code" == 1 ]; then  
        errecho " One or more S3 transfers failed."  
    elif [ "$err_code" == 2 ]; then  
        errecho " Command line failed to parse."  
    elif [ "$err_code" == 130 ]; then  
        errecho " Process received SIGINT."  
    elif [ "$err_code" == 252 ]; then  
        errecho " Command syntax invalid."  
    elif [ "$err_code" == 253 ]; then  
        errecho " The system environment or configuration was invalid."  
    elif [ "$err_code" == 254 ]; then  
        errecho " The service returned an error."  
    elif [ "$err_code" == 255 ]; then  
        errecho " 255 is a catch-all error."  
    fi  
  
    return 0  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [TerminateInstances](#)」の「」を参照してください。

HealthImaging Bash スクリプト AWS CLI で使用する例

次のコード例は、AWS Command Line Interface で Bash スクリプトを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています HealthImaging。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

CreateDatastore

次の例は、CreateDatastore を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function imaging_create_datastore
#
# This function creates an AWS HealthImaging data store for importing DICOM P10
# files.
#
# Parameters:
#     -n data_store_name - The name of the data store.
#
# Returns:
#     The datastore ID.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
```

```
function imaging_create_datastore() {
    local datastore_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function imaging_create_datastore"
        echo "Creates an AWS HealthImaging data store for importing DICOM P10 files."
        echo "  -n data_store_name - The name of the data store."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) datastore_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$datastore_name" ]]; then
        errecho "ERROR: You must provide a data store name with the -n parameter."
        usage
        return 1
    fi

    response=$(aws medical-imaging create-datastore \
        --datastore-name "$datastore_name" \
        --output text \
        --query 'datastoreId')

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
        aws_cli_error_log $error_code
    fi
}
```

```

    errecho "ERROR: AWS reports medical-imaging create-datastore operation failed.
$response"
    return 1
fi

echo "$response"

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateDatastore](#)」の「」を参照してください。

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

DeleteDatastore

次の例は、DeleteDatastore を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function imaging_delete_datastore
#
# This function deletes an AWS HealthImaging data store.
#
# Parameters:
#     -i datastore_id - The ID of the data store.

```

```

#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function imaging_delete_datastore() {
    local datastore_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function imaging_delete_datastore"
        echo "Deletes an AWS HealthImaging data store."
        echo "  -i datastore_id - The ID of the data store."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) datastore_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$datastore_id" ]]; then
        errecho "ERROR: You must provide a data store ID with the -i parameter."
        usage
        return 1
    fi

    response=$(aws medical-imaging delete-datastore \
        --datastore-id "$datastore_id")

    local error_code=${?}

```

```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports medical-imaging delete-datastore operation failed.
$response"
    return 1
fi

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteDatastore](#)」の「」を参照してください。

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

GetDatastore

次の例は、GetDatastore を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function imaging_get_datastore
#
# Get a data store's properties.
#
# Parameters:

```

```

#     -i data_store_id - The ID of the data store.
#
# Returns:
#     [datastore_name, datastore_id, datastore_status, datastore_arn,  created_at,
updated_at]
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function imaging_get_datastore() {
    local datastore_id option OPTARG # Required to use getopt command in a function.
    local error_code
    # bashsupport disable=BP5008
    function usage() {
        echo "function imaging_get_datastore"
        echo "Gets a data store's properties."
        echo "  -i datastore_id - The ID of the data store."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) datastore_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$datastore_id" ]]; then
        errecho "ERROR: You must provide a data store ID with the -i parameter."
        usage
        return 1
    fi

    local response

```

```

response=$(
  aws medical-imaging get-datastore \
    --datastore-id "$datastore_id" \
    --output text \
    --query "[ datastoreProperties.datastoreName,
datastoreProperties.datastoreId, datastoreProperties.datastoreStatus,
datastoreProperties.datastoreArn,  datastoreProperties.createdAt,
datastoreProperties.updatedAt]"
)
error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports list-datastores operation failed.$response"
  return 1
fi

echo "$response"

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [GetDatastore](#)」の「」を参照してください。

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

ListDatastores

次の例は、ListDatastores を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

```

#####
# function errecho
#

```



```

# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function imaging_list_datastores
#
# List the HealthImaging data stores in the account.
#
# Returns:
#     [[datastore_name, datastore_id, datastore_status]]
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function imaging_list_datastores() {
    local option OPTARG # Required to use getopt command in a function.
    local error_code
    # bashsupport disable=BP5008
    function usage() {
        echo "function imaging_list_datastores"
        echo "Lists the AWS HealthImaging data stores in the account."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "h" option; do
        case "${option}" in
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local response

```

```
response=$(aws medical-imaging list-datastores \
  --output text \
  --query "datastoreSummaries[*][datastoreName, datastoreId, datastoreStatus]")
error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports list-datastores operation failed.$response"
  return 1
fi

echo "$response"

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[ListDatastores](#)」の「」を参照してください。

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

IAM Bash スクリプト AWS CLI で を使用する例

次のコード例は、AWS Command Line Interface で Bash スクリプトを使用してアクションを実行し、一般的なシナリオを実装する方法を示していますIAM。

「基本」は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [基本](#)
- [アクション](#)

基本

基本を学ぶ

次のコードサンプルは、ユーザーを作成してロールを割り当てる方法を示しています。

Warning

セキュリティ上のリスクを回避するため、専用ソフトウェアの開発時や実際のデータの使用時に、認証にIAMユーザーを使用しないでください。代わりに、[AWS IAM Identity Center](#)などの ID プロバイダーとのフェデレーションを使用してください。

- 権限のないユーザーを作成します。
- 指定したアカウントに Amazon S3 バケットへのアクセス権限を付与するロールを作成します。
- ユーザーにロールを引き受けさせるポリシーを追加します。
- ロールを引き受け、一時的な認証情報を使用して S3 バケットを一覧表示しリソースをクリーンアップします。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####  
# function iam_create_user_assume_role  
#  
# Scenario to create an IAM user, create an IAM role, and apply the role to the  
# user.  
#  
# "IAM access" permissions are needed to run this code.
```

```
# "STS assume role" permissions are needed to run this code. (Note: It might be
# necessary to
# create a custom policy).
#
# Returns:
# 0 - If successful.
# 1 - If an error occurred.
#####
function iam_create_user_assume_role() {
{
if [ "$IAM_OPERATIONS_SOURCED" != "True" ]; then

source ./iam_operations.sh
fi
}

echo_repeat "*" 88
echo "Welcome to the IAM create user and assume role demo."
echo
echo "This demo will create an IAM user, create an IAM role, and apply the role to
the user."
echo_repeat "*" 88
echo

echo -n "Enter a name for a new IAM user: "
get_input
user_name=$get_input_result

local user_arn
user_arn=$(iam_create_user -u "$user_name")

# shellcheck disable=SC2181
if [[ ${?} == 0 ]]; then
echo "Created demo IAM user named $user_name"
else
errecho "$user_arn"
errecho "The user failed to create. This demo will exit."
return 1
fi

local access_key_response
access_key_response=$(iam_create_user_access_key -u "$user_name")
# shellcheck disable=SC2181
if [[ ${?} != 0 ]]; then
```

```
errecho "The access key failed to create. This demo will exit."
clean_up "$user_name"
return 1
fi

IFS=$'\t ' read -r -a access_key_values <<<"$access_key_response"
local key_name=${access_key_values[0]}
local key_secret=${access_key_values[1]}

echo "Created access key named $key_name"

echo "Wait 10 seconds for the user to be ready."
sleep 10
echo_repeat "*" 88
echo

local iam_role_name
iam_role_name=$(generate_random_name "test-role")
echo "Creating a role named $iam_role_name with user $user_name as the principal."

local assume_role_policy_document="{
  \"Version\": \"2012-10-17\",
  \"Statement\": [{
    \"Effect\": \"Allow\",
    \"Principal\": {\"AWS\": \"$user_arn\"},
    \"Action\": \"sts:AssumeRole\"
  }]
}"

local role_arn
role_arn=$(iam_create_role -n "$iam_role_name" -p "$assume_role_policy_document")

# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
  echo "Created IAM role named $iam_role_name"
else
  errecho "The role failed to create. This demo will exit."
  clean_up "$user_name" "$key_name"
  return 1
fi

local policy_name
policy_name=$(generate_random_name "test-policy")
local policy_document="{
```

```

        \ "Version\": \ "2012-10-17\",
        \ "Statement\": [{
            \ "Effect\": \ "Allow\",
            \ "Action\": \ "s3:ListAllMyBuckets\",
            \ "Resource\": \ "arn:aws:s3:::*\"}]}"

local policy_arn
policy_arn=$(iam_create_policy -n "$policy_name" -p "$policy_document")
# shellcheck disable=SC2181
if [[ ${?} == 0 ]]; then
    echo "Created IAM policy named $policy_name"
else
    errecho "The policy failed to create."
    clean_up "$user_name" "$key_name" "$iam_role_name"
    return 1
fi

if (iam_attach_role_policy -n "$iam_role_name" -p "$policy_arn"); then
    echo "Attached policy $policy_arn to role $iam_role_name"
else
    errecho "The policy failed to attach."
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn"
    return 1
fi

local assume_role_policy_document="{
    \ "Version\": \ "2012-10-17\",
    \ "Statement\": [{
        \ "Effect\": \ "Allow\",
        \ "Action\": \ "sts:AssumeRole\",
        \ "Resource\": \ "$role_arn\"}]}"

local assume_role_policy_name
assume_role_policy_name=$(generate_random_name "test-assume-role-")

# shellcheck disable=SC2181
local assume_role_policy_arn
assume_role_policy_arn=$(iam_create_policy -n "$assume_role_policy_name" -p
"$assume_role_policy_document")
# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
    echo "Created IAM policy named $assume_role_policy_name for sts assume role"
else
    errecho "The policy failed to create."

```

```
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
    return 1
fi

echo "Wait 10 seconds to give AWS time to propagate these new resources and
connections."
sleep 10
echo_repeat "*" 88
echo

echo "Try to list buckets without the new user assuming the role."
echo_repeat "*" 88
echo

# Set the environment variables for the created user.
# bashsupport disable=BP2001
export AWS_ACCESS_KEY_ID=$key_name
# bashsupport disable=BP2001
export AWS_SECRET_ACCESS_KEY=$key_secret

local buckets
buckets=$(s3_list_buckets)

# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. This should not have
happened."
else
    errecho "Because the role with permissions has not been assumed, listing buckets
failed."
fi

echo
echo_repeat "*" 88
echo "Now assume the role $iam_role_name and list the buckets."
echo_repeat "*" 88
echo

local credentials
credentials=$(sts_assume_role -r "$role_arn" -n "AssumeRoleDemoSession")
# shellcheck disable=SC2181
```

```
if [ ${?} == 0 ]; then
    echo "Assumed role $iam_role_name"
else
    errecho "Failed to assume role."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"
    return 1
fi

IFS=$'\t ' read -r -a credentials <<<"$credentials"

export AWS_ACCESS_KEY_ID=${credentials[0]}
export AWS_SECRET_ACCESS_KEY=${credentials[1]}
# bashsupport disable=BP2001
export AWS_SESSION_TOKEN=${credentials[2]}

buckets=$(s3_list_buckets)

# shellcheck disable=SC2181
if [ ${?} == 0 ]; then
    local bucket_count
    bucket_count=$(echo "$buckets" | wc -w | xargs)
    echo "There are $bucket_count buckets in the account. Listing buckets succeeded
because of "
    echo "the assumed role."
else
    errecho "Failed to list buckets. This should not happen."
    export AWS_ACCESS_KEY_ID=""
    export AWS_SECRET_ACCESS_KEY=""
    export AWS_SESSION_TOKEN=""
    clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"
    return 1
fi

local result=0
export AWS_ACCESS_KEY_ID=""
export AWS_SECRET_ACCESS_KEY=""

echo
echo_repeat "*" 88
echo "The created resources will now be deleted."
```



```

echo_repeat "*" 88
echo

clean_up "$user_name" "$key_name" "$iam_role_name" "$policy_arn" "$policy_arn"
"$assume_role_policy_arn"

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    result=1
fi

return $result
}

```

このシナリオで使用されるIAM関数。

```

#####
# function iam_user_exists
#
# This function checks to see if the specified AWS Identity and Access Management
# (IAM) user already exists.
#
# Parameters:
#     $1 - The name of the IAM user to check.
#
# Returns:
#     0 - If the user already exists.
#     1 - If the user doesn't exist.
#####
function iam_user_exists() {
    local user_name
    user_name=$1

    # Check whether the IAM user already exists.
    # We suppress all output - we're interested only in the return code.

    local errors
    errors=$(aws iam get-user \
        --user-name "$user_name" 2>&1 >/dev/null)

    local error_code=${?}

```

```

if [[ $error_code -eq 0 ]]; then
    return 0 # 0 in Bash script means true.
else
    if [[ $errors != *"error"*(NoSuchEntity)* ]]; then
        aws_cli_error_log $error_code
        errecho "Error calling iam get-user $errors"
    fi

    return 1 # 1 in Bash script means false.
fi
}

#####
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#     -u user_name  -- The name of the user to create.
#
# Returns:
#     The ARN of the user.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_user() {
    local user_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user"
        echo "Creates an WS Identity and Access Management (IAM) user. You must supply a
username:"
        echo "  -u user_name    The name of the user. It must be unique within the
account."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:h" option; do
        case "${option}" in

```

```
u) user_name="${OPTARG}" ;;
h)
  usage
  return 0
  ;;
\?)
  echo "Invalid parameter"
  usage
  return 1
  ;;
esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "  User name:  $user_name"
iecho ""

# If the user already exists, we don't want to try to create it.
if (iam_user_exists "$user_name"); then
  errecho "ERROR: A user with that name already exists in the account."
  return 1
fi

response=$(aws iam create-user --user-name "$user_name" \
  --output text \
  --query 'User.Arn')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-user operation failed.$response"
  return 1
fi

echo "$response"
```

```

    return 0
}

#####
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#     -u user_name -- The name of the IAM user.
#     [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#     [access_key_id access_key_secret]
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_user_access_key() {
    local user_name file_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user_access_key"
        echo "Creates an AWS Identity and Access Management (IAM) key pair."
        echo "  -u user_name   The name of the IAM user."
        echo "  [-f file_name] Optional file name for the access key output."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:f:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            f) file_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
        esac
    done
}

```

```
        ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
fi

if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
fi

local key_id key_secret
# shellcheck disable=SC2086
key_id=$(echo $response | cut -f 2 -d ' ')
# shellcheck disable=SC2086
key_secret=$(echo $response | cut -f 4 -d ' ')

echo "$key_id $key_secret"

return 0
}

#####
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
```

```

#     -n role_name -- The name of the IAM role.
#     -p policy_json -- The assume role policy document.
#
# Returns:
#     The ARN of the role.
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_role() {
    local role_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user_access_key"
        echo "Creates an AWS Identity and Access Management (IAM) role."
        echo "  -n role_name  The name of the IAM role."
        echo "  -p policy_json -- The assume role policy document."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_document="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$role_name" ]]; then
        errecho "ERROR: You must provide a role name with the -n parameter."
        usage
        return 1
    fi
}

```

```

fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#     -n policy_name -- The name of the IAM policy.
#     -p policy_json -- The policy document.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_policy() {
    local policy_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

```

```
# bashsupport disable=BP5008
function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name  The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
    case "${option}" in
        n) policy_name="${OPTARG}" ;;
        p) policy_document="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)
```



```

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
fi

echo "$response"
}

#####
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a role.
#
# Parameters:
#   -n role_name -- The name of the IAM role.
#   -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_attach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_attach_role_policy"
        echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
        echo "  -n role_name    The name of the IAM role."
        echo "  -p policy_ARN -- The IAM policy document ARN."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_arn="${OPTARG}" ;;
        esac
    done
}

```

```
h)
  usage
  return 0
  ;;
\?)
  echo "Invalid parameter"
  usage
  return 1
  ;;
esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam attach-role-policy \
  --role-name "$role_name" \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
  return 1
fi

echo "$response"

return 0
}

#####
# function iam_detach_role_policy
```

```
#
# This function detaches an IAM policy to a role.
#
# Parameters:
#   -n role_name -- The name of the IAM role.
#   -p policy_arn -- The IAM policy document ARN..
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_detach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_detach_role_policy"
        echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
        echo "  -n role_name  The name of the IAM role."
        echo "  -p policy_arn -- The IAM policy document ARN."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_arn="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$role_name" ]]; then
```

```

    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#     -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_policy() {
    local policy_arn response
    local option OPTARG # Required to use getopt command in a function.

```

```
# bashsupport disable=BP5008
function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an WS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
    case "${option}" in
        n) policy_arn="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  Policy arn:  $policy_arn"
iecho ""

response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
fi
```

```

fi

iecho "delete-policy response:$response"
iecho

return 0
}

#####
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_role() {
    local role_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_role"
        echo "Deletes an WS Identity and Access Management (IAM) role"
        echo "  -n role_name -- The name of the IAM role."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
        esac
    done
}

```

```
        ;;
    esac
done
export OPTIND=1

echo "role_name:$role_name"
if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    Role name: $role_name"
iecho ""

response=$(aws iam delete-role \
    --role-name "$role_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
fi

iecho "delete-role response:$response"
iecho

return 0
}

#####
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#     -u user_name -- The name of the user.
#     -k access_key -- The access key to delete.
#
# Returns:
#     0 - If successful.
```

```

# 1 - If it fails.
#####
function iam_delete_access_key() {
    local user_name access_key response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_access_key"
        echo "Deletes an WS Identity and Access Management (IAM) access key for the
specified IAM user"
        echo "  -u user_name    The name of the user."
        echo "  -k access_key    The access key to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:k:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            k) access_key="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$user_name" ]]; then
        errecho "ERROR: You must provide a username with the -u parameter."
        usage
        return 1
    fi

    if [[ -z "$access_key" ]]; then
        errecho "ERROR: You must provide an access key with the -k parameter."
        usage
        return 1
    fi
}

```



```

fi

iecho "Parameters:\n"
iecho "  Username:  $user_name"
iecho "  Access key:  $access_key"
iecho ""

response=$(aws iam delete-access-key \
  --user-name "$user_name" \
  --access-key-id "$access_key")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
  return 1
fi

iecho "delete-access-key response:$response"
iecho

return 0
}

#####
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#   -u user_name  -- The name of the user to create.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_delete_user() {
  local user_name response
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"

```

```
    echo "Deletes an WS Identity and Access Management (IAM) user. You must supply a
username:"
    echo "  -u user_name    The name of the user."
    echo ""
}

# Retrieve the calling parameters.
while getopts "u:h" option; do
    case "${option}" in
        u) user_name="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  User name:  $user_name"
iecho ""

# If the user does not exist, we don't want to try to delete it.
if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
fi

response=$(aws iam delete-user \
    --user-name "$user_name")

local error_code=${?}
```

```
if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
fi

iecho "delete-user response:$response"
iecho

return 0
}
```

- API 詳細については、AWS CLI コマンドリファレンスの以下のトピックを参照してください。
 - [AttachRolePolicy](#)
 - [CreateAccessKey](#)
 - [CreatePolicy](#)
 - [CreateRole](#)
 - [CreateUser](#)
 - [DeleteAccessKey](#)
 - [DeletePolicy](#)
 - [DeleteRole](#)
 - [DeleteUser](#)
 - [DeleteUserPolicy](#)
 - [DetachRolePolicy](#)
 - [PutUserPolicy](#)

アクション

AttachRolePolicy

次のコード例は、AttachRolePolicy を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_attach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function iam_attach_role_policy"
    echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
    echo " -n role_name The name of the IAM role."
    echo " -p policy_ARN -- The IAM policy document ARN."
    echo ""
}
```

```
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam attach-role-policy \
  --role-name "$role_name" \
  --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
  return 1
fi
```

```

    echo "$response"

    return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [AttachRolePolicy](#)」の「」を参照してください。

CreateAccessKey

次の例は、CreateAccessKey を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_create_user_access_key
#
# This function creates an IAM access key for the specified user.
#
# Parameters:
#     -u user_name -- The name of the IAM user.
#     [-f file_name] -- The optional file name for the access key output.
#
# Returns:
#     [access_key_id access_key_secret]

```

```
# And:
# 0 - If successful.
# 1 - If it fails.
#####
function iam_create_user_access_key() {
    local user_name file_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user_access_key"
        echo "Creates an AWS Identity and Access Management (IAM) key pair."
        echo " -u user_name The name of the IAM user."
        echo " [-f file_name] Optional file name for the access key output."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:f:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            f) file_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$user_name" ]]; then
        errecho "ERROR: You must provide a username with the -u parameter."
        usage
        return 1
    fi

    response=$(aws iam create-access-key \
        --user-name "$user_name" \
        --output text)
```

```
local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
fi

if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
fi

local key_id key_secret
# shellcheck disable=SC2086
key_id=$(echo $response | cut -f 2 -d ' ')
# shellcheck disable=SC2086
key_secret=$(echo $response | cut -f 4 -d ' ')

echo "$key_id $key_secret"

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateAccessKey](#)」の「」を参照してください。

CreatePolicy

次のコード例は、CreatePolicy を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
```



```

# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_create_policy
#
# This function creates an IAM policy.
#
# Parameters:
#     -n policy_name -- The name of the IAM policy.
#     -p policy_json -- The policy document.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_policy() {
    local policy_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_policy"
        echo "Creates an AWS Identity and Access Management (IAM) policy."
        echo "  -n policy_name  The name of the IAM policy."
        echo "  -p policy_json -- The policy document."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) policy_name="${OPTARG}" ;;
            p) policy_document="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)

```

```
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
fi

echo "$response"
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreatePolicy](#)」の「」を参照してください。

CreateRole

次の例は、CreateRole を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_create_role
#
# This function creates an IAM role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_json -- The assume role policy document.
#
# Returns:
#     The ARN of the role.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_role() {
    local role_name policy_document response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user_access_key"
```

```
    echo "Creates an AWS Identity and Access Management (IAM) role."
    echo "  -n role_name    The name of the IAM role."
    echo "  -p policy_json -- The assume role policy document."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
    case "${option}" in
        n) role_name="${OPTARG}" ;;
        p) policy_document="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

local error_code=${?}
```

```
if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateRole](#)」の「」を参照してください。

CreateUser

次の例は、CreateUser を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
```

```

#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_create_user
#
# This function creates the specified IAM user, unless
# it already exists.
#
# Parameters:
#     -u user_name  -- The name of the user to create.
#
# Returns:
#     The ARN of the user.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_create_user() {
    local user_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_create_user"
        echo "Creates an WS Identity and Access Management (IAM) user. You must supply a
username:"
        echo "  -u user_name    The name of the user. It must be unique within the
account."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
        esac
    done

```

```
\?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    User name:    $user_name"
iecho ""

# If the user already exists, we don't want to try to create it.
if (iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name already exists in the account."
    return 1
fi

response=$(aws iam create-user --user-name "$user_name" \
    --output text \
    --query 'User.Arn')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-user operation failed.$response"
    return 1
fi

echo "$response"

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateUser](#)」の「」を参照してください。

DeleteAccessKey

次の例は、DeleteAccessKey を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_access_key
#
# This function deletes an IAM access key for the specified IAM user.
#
# Parameters:
#     -u user_name -- The name of the user.
#     -k access_key -- The access key to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_access_key() {
    local user_name access_key response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
```



```
function usage() {
    echo "function iam_delete_access_key"
    echo "Deletes an WS Identity and Access Management (IAM) access key for the
specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key    The access key to delete."
    echo ""
}

# Retrieve the calling parameters.
while getopts "u:k:h" option; do
    case "${option}" in
        u) user_name="${OPTARG}" ;;
        k) access_key="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  Username:  $user_name"
iecho "  Access key: $access_key"
iecho ""
```

```

response=$(aws iam delete-access-key \
  --user-name "$user_name" \
  --access-key-id "$access_key")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
  return 1
fi

iecho "delete-access-key response:$response"
iecho

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteAccessKey](#)」の「」を参照してください。

DeletePolicy

次のコード例は、DeletePolicy を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {

```

```

    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#     -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_policy() {
    local policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_policy"
        echo "Deletes an WS Identity and Access Management (IAM) policy"
        echo "  -n policy_arn -- The name of the IAM policy arn."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) policy_arn="${OPTARG}" ;;
            h)
                usage

```

```
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    Policy arn: $policy_arn"
iecho ""

response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
fi

iecho "delete-policy response:$response"
iecho

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeletePolicy](#)」の「」を参照してください。

DeleteRole

次の例は、DeleteRole を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#
# Returns:
#     0 - If successful.
```

```
# 1 - If it fails.
#####
function iam_delete_role() {
    local role_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_role"
        echo "Deletes an WS Identity and Access Management (IAM) role"
        echo " -n role_name -- The name of the IAM role."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    echo "role_name:$role_name"
    if [[ -z "$role_name" ]]; then
        errecho "ERROR: You must provide a role name with the -n parameter."
        usage
        return 1
    fi

    iecho "Parameters:\n"
    iecho "    Role name: $role_name"
    iecho ""

    response=$(aws iam delete-role \
        --role-name "$role_name")
}
```

```

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
fi

iecho "delete-role response:$response"
iecho

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteRole](#)」の「」を参照してください。

DeleteUser

次の例は、DeleteUser を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

```

```
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_delete_user
#
# This function deletes the specified IAM user.
#
# Parameters:
#     -u user_name -- The name of the user to create.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_user() {
    local user_name response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_user"
        echo "Deletes an WS Identity and Access Management (IAM) user. You must supply a
username:"
        echo "  -u user_name    The name of the user."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "u:h" option; do
        case "${option}" in
            u) user_name="${OPTARG}" ;;
            h)
                usage
                return 0
            ;;
        esac
    done
}
```



```
\?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "    User name:    $user_name"
iecho ""

# If the user does not exist, we don't want to try to delete it.
if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
fi

response=$(aws iam delete-user \
    --user-name "$user_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
fi

iecho "delete-user response:$response"
iecho

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[DeleteUser](#)」の「」を参照してください。

DetachRolePolicy

次の例は、DetachRolePolicy を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_detach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
```

```
function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
    echo "  -n role_name    The name of the IAM role."
    echo "  -p policy_arn -- The IAM policy document ARN."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
    case "${option}" in
        n) role_name="${OPTARG}" ;;
        p) policy_arn="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam detach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}
```

```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DetachRolePolicy](#)」の「」を参照してください。

GetUser

次のコード例は、GetUser を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_user_exists
#

```

```
# This function checks to see if the specified AWS Identity and Access Management
(IAM) user already exists.
#
# Parameters:
#     $1 - The name of the IAM user to check.
#
# Returns:
#     0 - If the user already exists.
#     1 - If the user doesn't exist.
#####
function iam_user_exists() {
    local user_name
    user_name=$1

    # Check whether the IAM user already exists.
    # We suppress all output - we're interested only in the return code.

    local errors
    errors=$(aws iam get-user \
        --user-name "$user_name" 2>&1 >/dev/null)

    local error_code=${?}

    if [[ $error_code -eq 0 ]]; then
        return 0 # 0 in Bash script means true.
    else
        if [[ $errors != *"error"*(NoSuchEntity)* ]]; then
            aws_cli_error_log $error_code
            errecho "Error calling iam get-user $errors"
        fi


        return 1 # 1 in Bash script means false.
    fi
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetUser](#)」の「」を参照してください。

ListAccessKeys

次の例は、ListAccessKeys を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_list_access_keys
#
# This function lists the access keys for the specified user.
#
# Parameters:
#     -u user_name -- The name of the IAM user.
#
# Returns:
#     access_key_ids
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_list_access_keys() {

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_list_access_keys"
        echo "Lists the AWS Identity and Access Management (IAM) access key IDs for the
specified user."
        echo "  -u user_name  The name of the IAM user."
        echo ""
    }
}
```

```
local user_name response
local option OPTARG # Required to use getopt command in a function.
# Retrieve the calling parameters.
while getopt "u:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

response=$(aws iam list-access-keys \
  --user-name "$user_name" \
  --output text \
  --query 'AccessKeyMetadata[].AccessKeyId')

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports list-access-keys operation failed.$response"
  return 1
fi

echo "$response"

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListAccessKeys](#)」の「」を参照してください。

ListUsers

次のコード例は、ListUsers を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function iam_list_users
#
# List the IAM users in the account.
#
# Returns:
#     The list of users names
#     And:
#     0 - If the user already exists.
#     1 - If the user doesn't exist.
#####
function iam_list_users() {
    local option OPTARG # Required to use getopt command in a function.
    local error_code
    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_list_users"
        echo "Lists the AWS Identity and Access Management (IAM) user in the account."
    }
}
```



```
    echo ""
}

# Retrieve the calling parameters.
while getopts "h" option; do
    case "${option}" in
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local response

response=$(aws iam list-users \
    --output text \
    --query "Users[].UserName")
error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports list-users operation failed.$response"
    return 1
fi

echo "$response"

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [ListUsers](#)」の「」を参照してください。

Bash スクリプトを使用した Amazon S3 の例 AWS CLI

次のコード例は、Amazon S3 AWS Command Line Interface で Bash スクリプトを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています。

「基本」は、重要なオペレーションをサービス内で実行する方法を示すコード例です。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [基本](#)
- [アクション](#)

基本

基本を学ぶ

次のコードサンプルは、以下の操作方法を示しています。

- バケットを作成し、そこにファイルをアップロードします。
- バケットからオブジェクトをダウンロードします。
- バケット内のサブフォルダにオブジェクトをコピーします。
- バケット内のオブジェクトを一覧表示します。
- バケットオブジェクトとバケットを削除します。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function s3_getting_started
#
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#     0 - If successful.
#     1 - If an error occurred.
#####
function s3_getting_started() {
    {
        if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
            cd bucket-lifecycle-operations || exit

            source ./bucket_operations.sh
            cd ..
        fi
    }

    echo_repeat "*" 88
    echo "Welcome to the Amazon S3 getting started demo."
    echo_repeat "*" 88
    echo "A unique bucket will be created by appending a Universally Unique
Identifier to a bucket name prefix."
    echo -n "Enter a prefix for the S3 bucket that will be used in this demo: "
    get_input
    bucket_name_prefix=$get_input_result
    local bucket_name
    bucket_name=$(generate_random_name "$bucket_name_prefix")

    local region_code
    region_code=$(aws configure get region)

    if create_bucket -b "$bucket_name" -r "$region_code"; then
        echo "Created demo bucket named $bucket_name"
    else
        errecho "The bucket failed to create. This demo will exit."
        return 1
    fi

    local file_name
    while [ -z "$file_name" ]; do
        echo -n "Enter a file you want to upload to your bucket: "
```

```
get_input
file_name=$get_input_result

if [ ! -f "$file_name" ]; then
    echo "Could not find file $file_name. Are you sure it exists?"
    file_name=""
fi
done

local key
key="$(basename "$file_name")"

local result=0
if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
    echo "Uploaded file $file_name into bucket $bucket_name with key $key."
else
    result=1
fi

local destination_file
destination_file="$file_name.download"
if yes_no_input "Would you like to download $key to the file $destination_file?
(y/n) "; then
    if download_object_from_bucket "$bucket_name" "$destination_file" "$key"; then
        echo "Downloaded $key in the bucket $bucket_name to the file
$destination_file."
    else
        result=1
    fi
fi

if yes_no_input "Would you like to copy $key a new object key in your bucket? (y/
n) "; then
    local to_key
    to_key="demo/$key"
    if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
        echo "Copied $key in the bucket $bucket_name to the $to_key."
    else
        result=1
    fi
fi

local bucket_items
bucket_items=$(list_items_in_bucket "$bucket_name")
```

```

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    result=1
fi

echo "Your bucket contains the following items."
echo -e "Name\t\tSize"
echo "$bucket_items"

if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
(y/n) "; then
    bucket_items=$(echo "$bucket_items" | cut -f 1)

    if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
        echo "The following items were deleted from the bucket $bucket_name"
        echo "$bucket_items"
    else
        result=1
    fi

    if delete_bucket "$bucket_name"; then
        echo "Deleted the bucket $bucket_name"
    else
        result=1
    fi
fi

return $result
}

```

このシナリオで使用される Amazon S3 関数。

```

#####
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#     -b bucket_name -- The name of the bucket to create.
#     -r region_code -- The code for an AWS Region in which to

```

```
#           create the bucket.
#
# Returns:
#   The URL of the bucket that was created.
#   And:
#   0 - If successful.
#   1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function create_bucket"
        echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
        echo "  -b bucket_name    The name of the bucket. It must be globally unique."
        echo "  [-r region_code]   The code for an AWS Region in which the bucket is
created."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "b:r:h" option; do
        case "${option}" in
            b) bucket_name="${OPTARG}" ;;
            r) region_code="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done

    if [[ -z "$bucket_name" ]]; then
        errecho "ERROR: You must provide a bucket name with the -b parameter."
        usage
        return 1
    fi
}
```

```

local bucket_config_arg
# A location constraint for "us-east-1" returns an error.
if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
iecho "    Bucket name:    $bucket_name"
iecho "    Region code:    $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
fi

# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
}

#####
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file to.
#     $2 - The path and file name of the local file to copy to the bucket.
#     $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.

```

```
#####
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3

  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

#####
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#   $1 - The name of the bucket to download the object from.
#   $2 - The path and file name to store the downloaded bucket.
#   $3 - The key (name) of the object in the bucket.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function download_object_from_bucket() {
  local bucket_name=$1
  local destination_file_name=$2
  local object_name=$3
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
    --key "$object_name" \
    "$destination_file_name")

```



```

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
fi
}

#####
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response

    response=$(aws s3api copy-object \
        --bucket "$bucket_name" \
        --copy-source "$bucket_name/$source_key" \
        --key "$destination_key")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
        return 1
    fi
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and

```

```

# size fields from the Contents collection.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.
#     And:
#     0 - If successful.
#     1 - If it fails.
#####
function list_items_in_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api list-objects \
        --bucket "$bucket_name" \
        --output text \
        --query 'Contents[].{Key: Key, Size: Size}')

    # shellcheck disable=SC2181
    if [[ ${?} -eq 0 ]]; then
        echo "$response"
    else
        errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
        return 1
    fi
}

#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - A list of keys in the bucket to delete.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_items_in_bucket() {
    local bucket_name=$1

```

```

local keys=$2
local response

# Create the JSON for the items to delete.
local delete_items
delete_items="{\"Objects\":["
for key in $keys; do
    delete_items="{\"delete_items\":{\"Key\": \"$key\"},"
done
delete_items=${delete_items%?} # Remove the final comma.
delete_items="{\"delete_items\"]}"

response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

# shellcheck disable=SC2181
if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-object operation failed.\n$response"
    return 1
fi
}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api delete-bucket \
        --bucket "$bucket_name")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then

```

```

    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}

```

- API 詳細については、AWS CLI コマンドリファレンスの以下のトピックを参照してください。
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

アクション

CopyObject

次の例は、CopyObject を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

```

```
#####
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file from and to.
#     $2 - The key of the source file to copy.
#     $3 - The key of the destination file.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_item_in_bucket() {
    local bucket_name=$1
    local source_key=$2
    local destination_key=$3
    local response

    response=$(aws s3api copy-object \
        --bucket "$bucket_name" \
        --copy-source "$bucket_name/$source_key" \
        --key "$destination_key")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api copy-object operation failed.\n$response"
        return 1
    fi
}


```

- API 詳細については、AWS CLI 「コマンドリファレンス [CopyObject](#)」の「」を参照してください。

CreateBucket

次のコード例は、CreateBucket を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
#####
function iecho() {
    if [[ $VERBOSE == true ]]; then
        echo "$@"
    fi
}

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#     -b bucket_name -- The name of the bucket to create.
#     -r region_code -- The code for an AWS Region in which to
#                       create the bucket.
#
# Returns:
#     The URL of the bucket that was created.
```

```

# And:
# 0 - If successful.
# 1 - If it fails.
#####
function create_bucket() {
    local bucket_name region_code response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function create_bucket"
        echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
        echo " -b bucket_name    The name of the bucket. It must be globally unique."
        echo " [-r region_code]    The code for an AWS Region in which the bucket is
created."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "b:r:h" option; do
        case "${option}" in
            b) bucket_name="${OPTARG}" ;;
            r) region_code="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done

    if [[ -z "$bucket_name" ]]; then
        errecho "ERROR: You must provide a bucket name with the -b parameter."
        usage
        return 1
    fi

    local bucket_config_arg
    # A location constraint for "us-east-1" returns an error.
    if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then

```

```
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
fi

iecho "Parameters:\n"
iecho "    Bucket name:    $bucket_name"
iecho "    Region code:    $region_code"
iecho ""

# If the bucket already exists, we don't want to try to create it.
if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
fi

# shellcheck disable=SC2086
response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

# shellcheck disable=SC2181
if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
fi
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [CreateBucket](#)」の「」を参照してください。

DeleteBucket

次の例は、DeleteBucket を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。


```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_bucket() {
    local bucket_name=$1
    local response

    response=$(aws s3api delete-bucket \
        --bucket "$bucket_name")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
        return 1
    fi
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteBucket](#)」の「」を参照してください。

DeleteObject

次の例は、DeleteObject を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

 Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_item_in_bucket
#
# This function deletes the specified file from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - The key (file name) in the bucket to delete.

# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function delete_item_in_bucket() {
    local bucket_name=$1
    local key=$2
    local response

    response=$(aws s3api delete-object \
        --bucket "$bucket_name" \
        --key "$key")

    # shellcheck disable=SC2181
    if [[ $? -ne 0 ]]; then
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n$response"
    fi
}
```

```

    return 1
  fi
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteObject](#)」の「」を参照してください。

DeleteObjects

次の例は、DeleteObjects を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket.
#     $2 - A list of keys in the bucket to delete.

# Returns:
#     0 - If successful.
#     1 - If it fails.

```

```
#####  
function delete_items_in_bucket() {  
    local bucket_name=$1  
    local keys=$2  
    local response  
  
    # Create the JSON for the items to delete.  
    local delete_items  
    delete_items="{\"Objects\":["  
    for key in $keys; do  
        delete_items="$delete_items{\"Key\": \"$key\"},"  
    done  
    delete_items=${delete_items%?} # Remove the final comma.  
    delete_items="$delete_items]"  
  
    response=$(aws s3api delete-objects \  
        --bucket "$bucket_name" \  
        --delete "$delete_items")  
  
    # shellcheck disable=SC2181  
    if [[ $? -ne 0 ]]; then  
        errecho "ERROR: AWS reports s3api delete-object operation failed.\n$response"  
        return 1  
    fi  
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [DeleteObjects](#)」の「」を参照してください。

GetObject

次のコード例は、GetObject を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#     $1 - The name of the bucket to download the object from.
#     $2 - The path and file name to store the downloaded bucket.
#     $3 - The key (name) of the object in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function download_object_from_bucket() {
    local bucket_name=$1
    local destination_file_name=$2
    local object_name=$3
    local response

    response=$(aws s3api get-object \
        --bucket "$bucket_name" \
        --key "$object_name" \
        "$destination_file_name")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
        errecho "ERROR: AWS reports put-object operation failed.\n$response"
        return 1
    fi
}
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[GetObject](#)」の「」を参照してください。

HeadBucket

次のコード例は、HeadBucket を使用する方法を示しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function bucket_exists
#
# This function checks to see if the specified bucket already exists.
#
# Parameters:
#     $1 - The name of the bucket to check.
#
# Returns:
#     0 - If the bucket already exists.
#     1 - If the bucket doesn't exist.
#####
function bucket_exists() {
    local bucket_name
    bucket_name=$1

    # Check whether the bucket already exists.
    # We suppress all output - we're interested only in the return code.

    if aws s3api head-bucket \
        --bucket "$bucket_name" \
        >/dev/null 2>&1; then
        return 0 # 0 in Bash script means true.
    else
        return 1 # 1 in Bash script means false.
    fi
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス[HeadBucket](#)」の「」を参照してください。

ListObjectsV2

次の例は、ListObjectsV2 を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#     $1 - The name of the bucket.
#
# Returns:
#     The list of files in text format.
#
# And:
#     0 - If successful.
#     1 - If it fails.
```

```
#####
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}
#####
```

- API 詳細については、「コマンドリファレンス」の[ListObjectsV2](#)を参照してください。
AWS CLI

PutObject

次の例は、PutObject を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
```



```

    printf "%s\n" "$*" 1>&2
}

#####
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#     $1 - The name of the bucket to copy the file to.
#     $2 - The path and file name of the local file to copy to the bucket.
#     $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function copy_file_to_bucket() {
    local response bucket_name source_file destination_file_name
    bucket_name=$1
    source_file=$2
    destination_file_name=$3

    response=$(aws s3api put-object \
        --bucket "$bucket_name" \
        --body "$source_file" \
        --key "$destination_file_name")

    # shellcheck disable=SC2181
    if [[ ${?} -ne 0 ]]; then
        errecho "ERROR: AWS reports put-object operation failed.\n$response"
        return 1
    fi
}

```

- API 詳細については、AWS CLI 「コマンドリファレンス [PutObject](#)」の「」を参照してください。

AWS STS Bash スクリプト AWS CLI で を使用する例

次のコード例は、AWS Command Line Interface で Bash スクリプトを使用してアクションを実行し、一般的なシナリオを実装する方法を示しています AWS STS。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

各例には、完全なソースコードへのリンクが含まれています。ここでは、コンテキストでコードを設定および実行する方法の手順を確認できます。

トピック

- [アクション](#)

アクション

AssumeRole

次の例は、AssumeRole を使用する方法を説明しています。

AWS CLI Bash スクリプトを使用する

Note

の詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
#####  
# function iecho  
#  
# This function enables the script to display the specified text only if  
# the global variable $VERBOSE is set to true.  
#####  
function iecho() {  
    if [[ $VERBOSE == true ]]; then  
        echo "$@"  
    fi  
}  
}
```

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function sts_assume_role
#
# This function assumes a role in the AWS account and returns the temporary
# credentials.
#
# Parameters:
#     -n role_session_name -- The name of the session.
#     -r role_arn -- The ARN of the role to assume.
#
# Returns:
#     [access_key_id, secret_access_key, session_token]
# And:
#     0 - If successful.
#     1 - If an error occurred.
#####
function sts_assume_role() {
    local role_session_name role_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function sts_assume_role"
        echo "Assumes a role in the AWS account and returns the temporary credentials:"
        echo "  -n role_session_name -- The name of the session."
        echo "  -r role_arn -- The ARN of the role to assume."
        echo ""
    }

    while getopt n:r:h option; do
        case "${option}" in
            n) role_session_name=${OPTARG} ;;
            r) role_arn=${OPTARG} ;;
            h)
                usage
            ;;
        esac
    done
```

```
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done

response=$(aws sts assume-role \
  --role-session-name "$role_session_name" \
  --role-arn "$role_arn" \
  --output text \
  --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}
```

- API 詳細については、AWS CLI 「コマンドリファレンス [AssumeRole](#)」の「」を参照してください。

のセキュリティ AWS CLI

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS、最もセキュリティに敏感な組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャの恩恵を受けることができます。

セキュリティは、AWS とユーザーの間で責任を共有します。[責任共有モデル](#)、は、これをクラウドのセキュリティ、およびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ – クラウドで AWS AWS サービスを実行するインフラストラクチャを保護する AWS 責任があります。は、安全に使用できるサービス AWS も提供します。サードパーティーの監査者は、[AWS コンプライアンスプログラム](#)コンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。に適用されるコンプライアンスプログラムの詳細については AWS Command Line Interface、[AWS 「コンプライアンスプログラムによる対象範囲内のサービスコンプライアンス」](#)を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、AWS Command Line Interface () を使用する際に責任共有モデルを適用する方法を理解するのに役立ちますAWS CLI。次のトピックでは、セキュリティとコンプライアンスの目的 AWS CLI に合わせて を設定する方法を示します。また、 を使用して AWS リソースのモニタリングと保護 AWS CLI を行う方法についても説明します。

トピック

- [でのデータ保護 AWS CLI](#)
- [Identity and Access Management](#)
- [この AWS 製品またはサービスのコンプライアンス検証](#)
- [この AWS 製品またはサービスの耐障害性](#)
- [この AWS 製品またはサービスのインフラストラクチャセキュリティ](#)
- [TLS の最小バージョンの の強制 AWS CLI](#)

でのデータ保護 AWS CLI

責任 AWS [共有モデル](#)、のデータ保護に適用されます AWS Command Line Interface。このモデルで説明されているように、AWS は、すべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「[データプライバシーFAQ](#)」を参照してください。欧州でのデータ保護の詳細については、AWS 「セキュリティブログ」の [AWS 「責任共有モデル」とGDPR 「ブログ記事」](#) を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント、AWS IAM Identity Center または AWS Identity and Access Management () を使用して個々のユーザーを設定することをお勧めしますIAM。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。1.2 が必要でTLS、1.3 TLS をお勧めします。
- で APIとユーザーアクティビティのログ記録を設定します AWS CloudTrail。証 CloudTrail 跡を使用して AWS アクティビティをキャプチャする方法については、AWS CloudTrail 「ユーザーガイド」の [CloudTrail 「証跡の操作」](#) を参照してください。
- AWS 暗号化ソリューションと、内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは AWS を介して にアクセスするときに FIPS 140-3 検証済みの暗号化モジュールが必要な場合はAPI、FIPSエンドポイントを使用します。利用可能なFIPS エンドポイントの詳細については、「[連邦情報処理標準 \(FIPS\) 140-3](#)」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これには、コンソール、AWS CLI または を使用して AWS のサービス API AWS CLI または他の を操作する場合も含まれます AWS SDKs。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。URL を外部サーバーに提供する場合は、そのサーバーへのリクエストを検証URLするために認証情報を に含めないことを強くお勧めします。

データ暗号化

セキュリティで保護されたサービスの重要な特徴として、情報はアクティブに使用されていないときに暗号化されます。

保管中の暗号化

自体 AWS CLI は、ユーザーに代わって AWS サービスとやり取りするために必要な認証情報以外の顧客データを保存しません。

を使用して、顧客データをローカルコンピュータに送信して保存するための AWS サービスを AWS CLI 呼び出す場合は、そのサービスのユーザーガイドの「セキュリティとコンプライアンス」の章を参照して、そのデータの保存、保護、暗号化の方法を確認してください。

転送中の暗号化

デフォルトでは、AWS CLI および AWS サービスエンドポイントを実行しているクライアントコンピュータから送信されるすべてのデータは、HTTPS/TLS 接続を介してすべてを送信することで暗号化されます。

HTTPS/ の使用を有効にするために何もする必要はありません TLS。 --no-verify-ssl コマンドラインオプションを使用して個々のコマンドに対して明示的に無効にしない限り、常に有効になります。

Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つです。IAM 管理者は、誰を認証 (サインイン) し、誰に AWS リソースの使用を許可する (アクセス許可を付与する) かを制御します。IAM は追加料金なしで AWS のサービス 使用できる です。

トピック

- [対象者](#)
- [アイデンティティを使用した認証](#)
- [ポリシーを使用したアクセスの管理](#)
- [での の AWS のサービス 仕組み IAM](#)
- [AWS ID とアクセスのトラブルシューティング](#)

対象者

AWS Identity and Access Management (IAM) の使用方法は、で行う作業によって異なります AWS。

サービスユーザー – AWS のサービス を使用してジョブを実行する場合、管理者は必要な認証情報とアクセス許可を提供します。より多くの AWS 機能を使用して作業を行う際には、追加のアクセス許可が必要になる場合があります。アクセスの管理方法を理解しておく、管理者に適切な許可をリクエストするうえで役立ちます。の機能にアクセスできない場合は AWS、[AWS ID とアクセスのトラブルシューティング](#) または AWS のサービス 使用している のユーザーガイドを参照してください。

サービス管理者 – 社内の AWS リソースを担当している場合は、へのフルアクセスがある可能性があります AWS。サービスユーザーがどの AWS 機能やリソースにアクセスできるかを判断するのは、お客様の仕事です。その後、IAM管理者にリクエストを送信して、サービスユーザーのアクセス許可を変更する必要があります。このページの情報を確認して、の基本概念を理解しますIAM。会社IAMで を使用する方法の詳細については AWS、使用している AWS のサービスのユーザーガイドを参照してください。

IAM 管理者 – IAM管理者の場合は、へのアクセスを管理するポリシーの作成方法の詳細を知りたい場合があります AWS。で使用できる AWS ID ベースのポリシーの例を表示するにはIAM、AWS のサービス 使用している のユーザーガイドを参照してください。

アイデンティティを使用した認証

認証は、アイデンティティ認証情報 AWS を使用して にサインインする方法です。として、IAMユーザーとして AWS アカウントのルートユーザー、またはIAMロールを引き受けることで、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーテッド ID AWS として にサインインできます。AWS IAM Identity Center (IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーテッド ID の例です。フェデレーテッド ID としてサインインすると、管理者は以前に IAMロールを使用して ID フェデレーションをセットアップしていました。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることとなります。

ユーザーの種類に応じて、AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、AWS サインイン ユーザーガイドの「[へのサインイン方法 AWS アカウント](#)」を参照してください。

AWS プログラムでにアクセスする場合、はソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号化して署名します。AWS ツールを使用しない場合は、自分でリクエストに署名する必要があります。推奨される方法を使用してリクエストに署名する方法の詳細については、IAM ユーザーガイドの[AWS API 「リクエストの署名バージョン 4」](#)を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用する AWS ことをお勧めします。詳細については、AWS IAM Identity Center 「ユーザーガイド」の[「多要素認証」](#)と[AWS 「ユーザーガイド」の「多要素認証IAM」](#)を参照してください。IAM

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての および リソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインイン ID から始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインしてアクセスします。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、IAM 「ユーザーガイド」の[「ルートユーザー認証情報を必要とするタスク」](#)を参照してください。

フェデレーテッドアイデンティティ

ベストプラクティスとして、では、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用してアクセスするために ID プロバイダーとのフェデレーション AWS のサービスの使用を要求します。

フェデレーテッド ID は、エンタープライズユーザーディレクトリ、ウェブアイデンティティプロバイダー、AWS Directory Service、アイデンティティセンターディレクトリ、または ID ソースを通じて提供された認証情報 AWS のサービス を使用してアクセスするすべてのユーザーのユーザーです。フェデレーテッド ID がにアクセスすると AWS アカウント、それらはロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成したり、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用できます。IAM Identity Center の詳細については、AWS IAM Identity Center 「ユーザーガイド」の[IAM 「Identity Center とは」](#)を参照してください。

IAM ユーザーとグループ

[IAM ユーザー](#)とは、1人のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウントを持つ内の ID です。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を持つIAMユーザーを作成する代わりに、一時的な認証情報に依存することをお勧めします。ただし、IAMユーザーとの長期的な認証情報を必要とする特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM「ユーザーガイド」の「[長期的な認証情報を必要とするユースケースのアクセスキーを定期的にローテーションする](#)」を参照してください。

[IAM グループ](#)は、IAMユーザーのコレクションを指定する ID です。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、という名前のグループがありIAMAdmins、そのグループにIAMリソースを管理するアクセス許可を付与できます。

ユーザーは、ロールとは異なります。ユーザーは1人の人または1つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時認証情報が提供されます。詳細については、「[ユーザーガイド](#)」のIAM「[ユーザーのユースケース](#)」を参照してください。IAM

IAM ロール

[IAM ロール](#)は、特定のアクセス許可 AWS アカウントを持つ内の ID です。ユーザーと似ていますがIAM、特定の人物には関連付けられていません。でIAMロールを一時的に引き受けるには AWS Management Console、[ユーザーからIAMロール \(コンソール\) に切り替える](#)ことができます。または AWS API オペレーションを AWS CLI 呼び出すか、カスタム を使用してロールを引き受けることができますURL。ロールを使用する方法の詳細については、IAM ユーザーガイドの「[ロールを引き受ける方法](#)」を参照してください。

IAM 一時的な認証情報を持つ ロールは、次の状況で役立ちます。

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールの詳細については、IAM ユーザーガイドの「[サードパーティー ID プロバイダーのロールの作成](#)」を参照してください。IAM Identity Center を使用する場合は、アクセス許可セットを設定します。ID が認証された後にアクセスできる内容を制御するために、IAM Identity Center はアクセス

許可セットをのロールに関連付けますIAM。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「[アクセス許可セット](#)」を参照してください。

- 一時的なIAMユーザーアクセス許可 – IAM ユーザーまたはロールは、特定のタスクに対して異なるアクセス許可を一時的に引き受けるIAMロールを引き受けることができます。
- クロスアカウントアクセス – IAMロールを使用して、別のアカウントの誰か (信頼できるプリンシパル) が自分のアカウントのリソースにアクセスすることを許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、一部の AWS のサービス、(プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、IAM「ユーザーガイド」の「[のクロスアカウントリソースアクセスIAM](#)」を参照してください。
- クロスサービスアクセス – 他の の機能 AWS のサービス を使用するものもあります AWS のサービス。例えば、サービスで呼び出しを行う場合、そのサービスが Amazon でアプリケーションを実行EC2したりAmazon S3にオブジェクトを保存したりするのが一般的です。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス または リソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「[アクセスセッションの転送](#)」を参照してください。
- サービスロール – サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受けるIAMロールです。IAM 管理者は、内からサービスロールを作成、変更、削除できますIAM。詳細については、IAM「ユーザーガイド」の「[にアクセス許可を委任するロールの作成 AWS のサービス](#)」を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント、 サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。
- Amazon で実行されているアプリケーション EC2 – IAMロールを使用して、EC2インスタンスで実行され、AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管

理できます。これは、EC2インスタンス内にアクセスキーを保存するよりも望ましいです。AWS ロールをEC2インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルには、ロールが含まれており、EC2インスタンスで実行されているプログラムが一時的な認証情報を取得できるようにします。詳細については、IAM「[ユーザーガイド](#)」のIAM「[ロールを使用して Amazon EC2インスタンスで実行されているアプリケーションにアクセス許可を付与する](#)」を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御するには、ポリシー AWS を作成し、AWS アイデンティティまたはリソースにアタッチします。ポリシーは AWS、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義するオブジェクトです。は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーはJSONドキュメント AWS として保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「[ユーザーガイド](#)」のJSON「[ポリシーの概要](#)」を参照してください。IAM

管理者はポリシーを使用して AWS JSON、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するには、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行に使用する方法に関係なく、アクションのアクセス許可を定義します。例えば、iam:GetRoleアクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、AWS Management Console、AWS CLIまたは AWS からロール情報を取得できますAPI。

アイデンティティベースのポリシー

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできるJSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。ID ベースのポリシーを作成する方法については、IAM「[ユーザーガイド](#)」の「[カスタマーマネージドポリシーによるカスタムIAMアクセス許可の定義](#)」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。マネージドポリシーは、内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには AWS、管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーを選択する方法については、IAM ユーザーガイドの「[マネージドポリシーとインラインポリシーの選択](#)」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースにアタッチするJSONポリシードキュメントです。リソースベースのポリシーの例としては、IAMロール信頼ポリシーと Amazon S3 バケットポリシーがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#) 必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、またはを含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーIAMでは、から AWS 管理ポリシーを使用することはできません。

アクセスコントロールリスト (ACLs)

アクセスコントロールリスト (ACLs) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするアクセス許可を持っているかを制御します。ACLs はリソースベースのポリシーに似ていますが、JSONポリシードキュメント形式は使用しません。

Amazon S3、および Amazon VPCは AWS WAF、をサポートするサービスの例ですACLs。の詳細についてはACLs、「Amazon Simple Storage Service デベロッパガイド」の「[アクセスコントロールリスト \(ACL\) 概要](#)」を参照してください。

その他のポリシータイプ

AWS は、追加であまり一般的ではないポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- **アクセス許可の境界** – アクセス許可の境界は、アイデンティティベースのポリシーがIAMエンティティ (IAMユーザーまたはロール) に付与できる最大アクセス許可を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。Principal

フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、IAM「ユーザーガイド」の[IAM「エンティティのアクセス許可の境界」](#)を参照してください。

- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) の最大アクセス許可を指定する JSON ポリシーです AWS Organizations。AWS Organizations は、ビジネスが所有する複数の をグループ化して一元管理するためのサービス AWS アカウントです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCPs) をアカウントの一部またはすべてに適用できます。は、各 を含むメンバーアカウントのエンティティのアクセス許可 SCP を制限します AWS アカウントのルートユーザー。Organizations との詳細については SCPs、AWS Organizations 「ユーザーガイド」の[「サービスコントロールポリシー」](#)を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「ユーザーガイド」の[「セッションポリシー」](#)を参照してください。IAM

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうかが AWS を決定する方法については、ユーザーガイドの[「ポリシー評価ロジック」](#)を参照してください。IAM

での の AWS のサービス 仕組み IAM

ほとんどの IAM 機能で AWS のサービス がどのように機能するかの概要を確認するには、IAM ユーザーガイドの[AWS「で機能する のサービス IAM」](#)を参照してください。

AWS のサービス で特定の を使用する方法については IAM、関連するサービスのユーザーガイドのセキュリティセクションを参照してください。

AWS ID とアクセスのトラブルシューティング

AWS 以下の情報は、 および の使用時に発生する可能性のある一般的な問題を診断して修正するのに役立ちます IAM。

トピック

- [でアクションを実行する権限がありません AWS](#)
- [iam を実行する権限がありません。PassRole](#)
- [自分の外部のユーザーに自分の AWS リソース AWS アカウント へのアクセスを許可したい](#)

でアクションを実行する権限がありません AWS

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojacksonIAMユーザーがコンソールを使用して架空の`my-example-widget`リソースの詳細を表示しようとしたときに、架空の`aws:GetWidget`アクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

この場合、`aws:GetWidget` アクションを使用して `my-example-widget` リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

iam を実行する権限がありません。PassRole

`iam:PassRole` アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して AWS にロールを渡すことができるようにする必要があります。

一部の AWS のサービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

次のエラー例は、という名前の IAM ユーザーがコンソールを使用して `marymajor` でアクションを実行しようとするると発生します AWS。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

自分の外部のユーザーに自分の AWS リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACLs) をサポートするサービスでは、これらのポリシーを使用して、リソースへのアクセスをユーザーに許可できます。

詳細については、以下を参照してください。

- がこれらの機能 AWS をサポートしているかどうかについては、「」を参照してください [での AWS のサービス 仕組み IAM](#)。
- 所有 AWS アカウント している リソースへのアクセスを提供する方法については、IAM 「ユーザーガイド」の [「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」](#) を参照してください。
- サードパーティー にリソースへのアクセスを提供する方法については AWS アカウント、IAM ユーザーガイドの [「サードパーティー AWS アカウント が所有する へのアクセスを提供する」](#) を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの [「外部認証されたユーザーへのアクセスを提供する \(ID フェデレーション\)」](#) を参照してください。
- クロスアカウントアクセスにロールとリソースベースのポリシーを使用する違いについては、IAM ユーザーガイドの [「の クロスアカウントリソースアクセスIAM」](#) を参照してください。


この AWS 製品またはサービスのコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、[AWS のサービス 「コンプライアンスプログラムによるスコープ」](#) の「」の「」を参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、[AWS 「コンプライアンスプログラム」](#) を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードすることができます AWS Artifact。詳細については、「」の [AWS Artifact](#)」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービスは、データの機密性、会社のコンプライアンス目的、および適用される法律と規制によって決まります。は、コンプライアンスに役立つ以下のリソース AWS を提供します。

- [セキュリティとコンプライアンスのクイックスタートガイド](#) – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- [Amazon Web Services HIPAA のセキュリティとコンプライアンスのためのアーキテクチャ](#) – このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

 Note

すべての AWS のサービスが HIPAA 対象とは限りません。詳細については、[HIPAA 「対象サービスリファレンス」](#)を参照してください。

- [AWS コンプライアンスリソース](#) – このワークブックとガイドのコレクションは、お客様の業界とロケーションに適用される場合があります。
- [AWS カスタマーコンプライアンスガイド](#) – コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council ()、国際標準化機構 (ISO) など PCI) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- AWS Config デベロッパガイドの[ルールによるリソースの評価](#) – この AWS Config サービスは、リソース設定が内部プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。
- [AWS Security Hub](#) – これにより AWS のサービス、内のセキュリティ状態を包括的に確認できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、[Security Hub のコントロールリファレンス](#)を参照してください。
- [Amazon GuardDuty](#) – これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境をモニタリングすることで、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出できます。GuardDuty は、特定のコンプライアンスフレームワークで義務付けられている侵入検出要件を満たすことで DSS、PCI などのさまざまなコンプライアンス要件に対応するのに役立ちます。

- [AWS Audit Manager](#) – これにより AWS のサービス、AWS 使用状況を継続的に監査し、リスクと規制や業界標準へのコンプライアンスの管理を簡素化できます。

この AWS 製品またはサービスは、サポートする特定の Amazon Web Services (AWS) サービスを通じて[責任共有モデル](#)に従います。AWS サービスセキュリティ情報については、[AWS 「サービスセキュリティドキュメント」](#) ページと[AWS、AWS コンプライアンスプログラムによるコンプライアンスの取り組みの対象となるサービス](#)を参照してください。

この AWS 製品またはサービスの耐障害性

AWS グローバルインフラストラクチャは、AWS リージョン とアベイラビリティゾーンを中心に構築されています。

AWS リージョン は、低レイテンシー、高スループット、および冗長性の高いネットワークに接続されている、物理的に分離および分離された複数のアベイラビリティゾーンを提供します。

アベイラビリティゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

AWS リージョンとアベイラビリティゾーンの詳細については、[AWS 「グローバルインフラストラクチャ」](#)を参照してください。

この AWS 製品またはサービスは、サポートする特定の Amazon Web Services (AWS) サービスを通じて[責任共有モデル](#)に従います。AWS サービスセキュリティ情報については、[AWS 「サービスセキュリティドキュメント」](#) ページと[AWS、AWS コンプライアンスプログラムによるコンプライアンスの取り組みの対象となるサービス](#)を参照してください。

この AWS 製品またはサービスのインフラストラクチャセキュリティ

この AWS 製品またはサービスはマネージドサービスを使用しているため、グローバルネットワークセキュリティによって AWS 保護されています。AWS セキュリティサービスと [ガインフラストラクチャ AWS を保護する方法](#)については、[AWS 「クラウドセキュリティ」](#)を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するに

は、「Security Pillar AWS Well-Architected Framework」の「[Infrastructure Protection](#)」を参照してください。

AWS 公開されたAPI呼び出しを使用して、ネットワークを介してこの AWS 製品またはサービスにアクセスします。クライアントは以下をサポートする必要があります:

- Transport Layer Security (TLS)。1.2 が必要でTLS、1.3 TLS をお勧めします。
- (DHEエフェメラルディフィ-ヘルマンPFS) や (エリプティックカーブエフェメラルディフィ-ヘルマン) など、完全なフォワードシークレット ECDHE () を持つ暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

さらに、リクエストは、アクセスキー ID とプリンIAMシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service](#) (AWS STS) を使用して、一時セキュリティ認証情報を生成し、リクエストに署名することもできます。

この AWS 製品またはサービスは、サポートする特定の Amazon Web Services (AWS) サービスを通じて[責任共有モデル](#)に従います。AWS サービスセキュリティ情報については、[AWS 「サービスセキュリティドキュメント」](#) ページと[AWS 、 AWS コンプライアンスプログラム によるコンプライアンスの取り組みの対象となるサービス](#)を参照してください。

TLS の最小バージョンの の強制 AWS CLI

AWS Command Line Interface () を使用する場合AWS CLI、Transport Layer Security (TLS) プロトコルは、AWS CLI と 間の通信を保護する上で重要な役割を果たします AWS のサービス。AWS サービスと通信するときにセキュリティを強化するには、1.2 TLS 以降を使用する必要があります。

AWS CLI バージョン 2 では、 がサポートするサービスで、少なくとも TLS 1.2 を使用するようにコンパイルされた内部 Python スクリプトを使用します。のバージョン 2 を使用する限り AWS CLI、この最小値を適用するためにそれ以上のステップは必要ありません。セキュリティを強化するために、必ず の最新バージョンに更新してください AWS CLI。

AWS CLI および は、暗号化、認証、データ整合性を提供するTLSプロトコルを使用して、データを安全に交換 AWS のサービス できます。TLS プロトコルを活用することで、AWS CLI とのインタラクション AWS のサービス が不正アクセスやデータ侵害から保護され、AWS エコシステムのセキュリティ全体が強化されます。

責任 AWS [共有モデル](#)、 のデータ保護に適用されます AWS Command Line Interface。このモデルで説明されているように、AWS はすべての を実行するグローバルインフラストラクチャを保護す

る責任があります AWS のサービス。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、AWS のサービス 使用する のセキュリティ設定および管理タスクについても責任を負います。データ保護の詳細については、「」を参照してください [the section called “データ保護”](#)。

のエラーのトラブルシューティング AWS CLI

このセクションでは、問題を解決するために従うべき一般的なエラーとトラブルシューティング手順について説明します。最初に、[一般的なトラブルシューティング](#)に従うことをお勧めします。

目次

- [最初に試す一般的なトラブルシューティング](#)
 - [AWS CLI コマンドのフォーマットを確認する](#)
 - [AWS CLI コマンドが を使用している AWS リージョン ことを確認します。](#)
 - [最新バージョンの AWS CLI を実行していることを確認する](#)
 - [--debug オプションを使用する](#)
 - [AWS CLI コマンド履歴ログを有効にして確認する](#)
 - [AWS CLI が設定されていることを確認します。](#)
- [コマンドが見つからないエラー](#)
- [「aws --version」コマンドが、インストールしたのとは異なるバージョンを返す](#)
- [aws --version 「」コマンドは、AWS CLI](#)
- [は、パラメータ名が不完全なコマンド AWS CLI を処理しました](#)
- [アクセス拒否エラー](#)
- [無効な認証情報とキーエラー](#)
- [署名がエラーと一致しない](#)
- [SSL 証明書エラー](#)
- [無効なJSONエラー](#)
- [追加リソース](#)

最初に試す一般的なトラブルシューティング

でエラーが発生したり、問題が発生した場合は AWS CLI、トラブルシューティングに役立つ以下の一般的なヒントをお勧めします。

[先頭に戻る](#)

AWS CLI コマンドのフォーマットを確認する

コマンドが存在しないことを示すエラー、またはドキュメントに記述されているパラメータ (Parameter validation failed) が認識されないことを示すエラーが表示された場合は、コマンドの形式が正しくない可能性があります。以下について確認することをお勧めします。

- コマンドでスペルミスや形式エラーがないか確認します。
- コマンドで使用している [すべての引用符とエスケープが端末に合っている](#)ことを確認します。
- [AWS CLI スケルトン](#) を生成して、コマンド構造を確認します。
- についてはJSON、[JSON値に関する追加のトラブルシューティング](#)を参照してください。ターミナルの処理JSONフォーマットに問題がある場合は、[Blobs を使用してJSONデータをに直接渡す AWS CLI](#)ことで、ターミナルのクォーティングルールをスキップすることをお勧めします。

特定のコマンドの構造化方法の詳細については、[AWS CLI バージョン 2 リファレンスガイド](#) を参照してください。

[先頭に戻る](#)

AWS CLI コマンドが `aws` を使用している AWS リージョン ことを確認します。

Note

AWS リージョン を使用する場合 AWS CLI、明示的に、またはデフォルトのリージョンを設定して、`aws` を指定する必要があります。指定できるすべての AWS リージョンについては、[AWS 「」の「リージョンとエンドポイント」](#)を参照してくださいAmazon Web Services 全般のリファレンス。で使用される AWS リージョン 識別子 AWS CLI は、 および サービスエンドポイントに表示される AWS Management Console URLsものと同じ名前です。

`aws` が指定された で使用 AWS のサービス できない場合、またはリソースが別の リージョンにある場合、エラー AWS リージョン や予期しない結果が発生する可能性があります AWS リージョン。優先順位に従って、AWS リージョン は次の方法で設定されます。

- `--region` コマンドラインオプション。
- SDK 互換性のある [AWS_REGION](#) 環境変数。

- [AWS_DEFAULT_REGION](#) 環境変数。
- [region](#) プロファイル設定。

リソース AWS リージョン に正しい を使用していることを確認します。

[先頭に戻る](#)

最新バージョンの AWS CLI を実行していることを確認する

コマンドが存在しないことを示すエラー、または[AWS CLI バージョン 2 リファレンスガイド](#)で使用可能なパラメータを認識しないことを示すエラーが表示された場合は、まずコマンドが正しくフォーマットされていることを確認します。形式が正しい場合は、最新バージョンの AWS CLI にアップグレードすることをお勧めします。の更新バージョン AWS CLI は、ほぼ営業日にリリースされます。のこれらの新しいバージョンでは、新しい AWS サービス、機能、パラメータが導入されています AWS CLI。これらの新しいサービスや機能、パラメータにアクセスする唯一の方法は、その要素が最初に導入された後にリリースされたバージョンにアップグレードすることです。

のバージョンを更新する方法は、「」で説明されているように、最初にインストールした方法 AWS CLI によって異なります [the section called “インストール/更新”](#)。

バンドルされたインストーラのいずれかを使用した場合は、オペレーティングシステム用の最新バージョンをダウンロードしてインストールする前に、既存のインストールの削除が必要になる場合があります。

[先頭に戻る](#)

--debug オプションを使用する

がすぐに理解できないエラー AWS CLI を報告した場合、または予期されない結果を生成する場合は、--debug オプションを使用してコマンドを再度実行することで、エラーの詳細を確認できます。このオプションを使用すると、AWS CLI は、コマンドを処理するために必要な各ステップの詳細を出力します。出力される詳細は、いつエラーが発生し、そのエラーがどこで開始されたかを特定するために役立つヒントを提供します。

この出力は、後で確認できるようテキストファイルに送信するか、求められた場合に AWS Support に送信できます。

--debug オプションを含める場合、詳細の一部には以下が含まれます。

- 認証情報の検索

- 指定されたパラメータの解析
- AWS サーバーに送信されたリクエストの作成
- に送信されたリクエストの内容 AWS
- raw レスポンスの内容
- フォーマットされた出力

--debug オプションを使用した場合と使用しない場合のコマンドの例を次に示します。

```
$ aws iam list-groups --profile MyTestProfile
{
  "Groups": [
    {
      "Path": "/",
      "GroupName": "MyTestGroup",
      "GroupId": "AGPA0123456789EXAMPLE",
      "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
      "CreateDate": "2019-08-12T19:34:04Z"
    }
  ]
}
```

```
$ aws iam list-groups --profile MyTestProfile --debug
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.16.215 Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205
2019-08-12 12:36:18,305 - MainThread - awscli.clidriver - DEBUG - Arguments entered to
CLI: ['iam', 'list-groups', '--debug']
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function add_scalar_parsers at 0x7fdf173161e0>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function register_uri_param_handler at 0x7fdf17dec400>
2019-08-12 12:36:18,305 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function inject_assume_role_provider_cache at
0x7fdf17da9378>
2019-08-12 12:36:18,307 - MainThread - botocore.credentials - DEBUG - Skipping
environment variable credential check because profile name was explicitly set.
2019-08-12 12:36:18,307 - MainThread - botocore.hooks - DEBUG - Event session-
initialized: calling handler <function attach_history_handler at 0x7fdf173ed9d8>
2019-08-12 12:36:18,308 - MainThread - botocore.loaders - DEBUG - Loading JSON
file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
service-2.json
```



```
2019-08-12 12:36:18,317 - MainThread - botocore.hooks - DEBUG - Event building-command-
table.iam: calling handler <function add_waiters at 0x7fdf1731a840>
2019-08-12 12:36:18,320 - MainThread - botocore.loaders - DEBUG - Loading JSON
file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
waiters-2.json
2019-08-12 12:36:18,321 - MainThread - awscli.clidriver - DEBUG - OrderedDict([('path-
prefix', <awscli.arguments.CLIArument object at 0x7fdf171ac780>), ('marker',
<awscli.arguments.CLIArument object at 0x7fdf171b09e8>), ('max-items',
<awscli.arguments.CLIArument object at 0x7fdf171b09b0>)])
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_streaming_output_arg at
0x7fdf17316510>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_cli_input_json at
0x7fdf17da9d90>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function unify_paging_params at
0x7fdf17328048>
2019-08-12 12:36:18,326 - MainThread - botocore.loaders - DEBUG - Loading JSON
file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
paginator-1.json
2019-08-12 12:36:18,326 - MainThread - awscli.customizations.paginate - DEBUG -
Modifying paging parameters for operation: ListGroups
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_generate_skeleton at
0x7fdf1737eae8>
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event
before-building-argument-table-parser.iam.list-groups: calling handler
<bound method OverrideRequiredArgsArgument.override_required_args of
<awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event
before-building-argument-table-parser.iam.list-groups: calling handler
<bound method GenerateCliSkeletonArgument.override_required_args of
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
0x7fdf171c5978>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event operation-
args-parsed.iam.list-groups: calling handler functools.partial(<function
check_should_enable_pagination at 0x7fdf17328158>, ['marker', 'max-items'], {'max-
items': <awscli.arguments.CLIArument object at 0x7fdf171b09b0>}, OrderedDict([('path-
prefix', <awscli.arguments.CLIArument object at 0x7fdf171ac780>), ('marker',
<awscli.arguments.CLIArument object at 0x7fdf171b09e8>), ('max-items',
<awscli.customizations.paginate.PageArgument object at 0x7fdf171c58d0>), ('cli-
input-json', <awscli.customizations.cliinputjson.CliInputJSONArgument object at
0x7fdf171b0a58>), ('starting-token', <awscli.customizations.paginate.PageArgument
```

```
object at 0x7fdf171b0a20>), ('page-size', <awscli.customizations.paginate.PageArgument
object at 0x7fdf171c5828>), ('generate-cli-skeleton',
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
0x7fdf171c5978>]]))
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.path-prefix: calling handler <awscli.paramfile.URIArgumentHandler
object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.marker: calling handler <awscli.paramfile.URIArgumentHandler object
at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.max-items: calling handler <awscli.paramfile.URIArgumentHandler
object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
Event load-cli-arg.iam.list-groups.cli-input-json: calling handler
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
Event load-cli-arg.iam.list-groups.starting-token: calling handler
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.page-size: calling handler <awscli.paramfile.URIArgumentHandler
object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event
load-cli-arg.iam.list-groups.generate-cli-skeleton: calling handler
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG
- Event calling-command.iam.list-groups: calling handler
<bound method CliInputJSONArgument.add_to_call_parameters of
<awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG -
Event calling-command.iam.list-groups: calling handler <bound
method GenerateCliSkeletonArgument.generate_json_skeleton of
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
0x7fdf171c5978>>
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
credentials via: assume-role
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
credentials via: assume-role-with-web-identity
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
credentials via: shared-credentials-file
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - INFO - Found credentials
in shared credentials file: ~/.aws/credentials
2019-08-12 12:36:18,330 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/endpoints.json
```

```
2019-08-12 12:36:18,334 - MainThread - botocore.hooks - DEBUG - Event choose-service-
name: calling handler <function handle_service_name_alias at 0x7fdf1898eb70>
2019-08-12 12:36:18,337 - MainThread - botocore.hooks - DEBUG - Event creating-client-
class.iam: calling handler <function add_generate_presigned_url at 0x7fdf18a028c8>
2019-08-12 12:36:18,337 - MainThread - botocore.regions - DEBUG - Using partition
endpoint for iam, us-west-2: aws-global
2019-08-12 12:36:18,337 - MainThread - botocore.args - DEBUG - The s3 config key is not
a dictionary type, ignoring its value of: None
2019-08-12 12:36:18,340 - MainThread - botocore.endpoint - DEBUG - Setting iam timeout
as (60, 60)
2019-08-12 12:36:18,341 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/_retry.json
2019-08-12 12:36:18,341 - MainThread - botocore.client - DEBUG - Registering retry
handlers for service: iam
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
parameter-build.iam.ListGroups: calling handler <function generate_idempotent_uuid at
0x7fdf189b10d0>
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
call.iam.ListGroups: calling handler <function inject_api_version_header_if_needed at
0x7fdf189b2a60>
2019-08-12 12:36:18,343 - MainThread - botocore.endpoint - DEBUG - Making
request for OperationModel(name=ListGroups) with params: {'url_path': '/',
'query_string': '', 'method': 'POST', 'headers': {'Content-Type': 'application/x-
www-form-urlencoded; charset=utf-8', 'User-Agent': 'aws-cli/1.16.215 Python/3.7.3
Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205'}, 'body': {'Action':
'ListGroups', 'Version': '2010-05-08'}, 'url': 'https://iam.amazonaws.com/',
'context': {'client_region': 'aws-global', 'client_config': <botoconfig.config.Config
object at 0x7fdf16e9a4a8>, 'has_streaming_input': False, 'auth_type': None}}
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event request-
created.iam.ListGroups: calling handler <bound method RequestSigner.handler of
<botoconfig.signers.RequestSigner object at 0x7fdf16e9a470>>
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event choose-
signer.iam.ListGroups: calling handler <function set_operation_specific_signer at
0x7fdf18996f28>
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - Calculating signature
using v4 auth.
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - CanonicalRequest:
POST
/

content-type:application/x-www-form-urlencoded; charset=utf-8
host:iam.amazonaws.com
x-amz-date:20190812T193618Z
```

```

content-type;host;x-amz-date
5f776d91EXAMPLE9b8cb5eb5d6d4a787a33ae41c8cd6eEXAMPLEEca69080e1e1f
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - StringToSign:
AWS4-HMAC-SHA256
20190812T193618Z
20190812/us-east-1/iam/aws4_request
ab7e367eEXAMPLE2769f178ea509978cf8bfa054874b3EXAMPLE8d043fab6cc9
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - Signature:
d85a0EXAMPLEeb40164f2f539cdc76d4f294fe822EXAMPLE18ad1ddf58a1a3ce7
2019-08-12 12:36:18,344 - MainThread - botocore.endpoint - DEBUG - Sending
http request: <AWSPreparedRequest stream_output=False, method=POST,
url=https://iam.amazonaws.com/, headers={'Content-Type': b'application/
x-www-form-urlencoded; charset=utf-8', 'User-Agent': b'aws-cli/1.16.215
Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205',
'X-Amz-Date': b'20190812T193618Z', 'Authorization': b'AWS4-HMAC-SHA256
Credential=AKIA01234567890EXAMPLE-east-1/iam/aws4_request, SignedHeaders=content-
type;host;x-amz-date, Signature=d85a07692aceb401EXAMPLEa1b18ad1ddf58a1a3ce7EXAMPLE',
'Content-Length': '36'}>
2019-08-12 12:36:18,344 - MainThread - urllib3.util.retry - DEBUG - Converted retries
value: False -> Retry(total=False, connect=None, read=None, redirect=0, status=None)
2019-08-12 12:36:18,344 - MainThread - urllib3.connectionpool - DEBUG - Starting new
HTTPS connection (1): iam.amazonaws.com:443
2019-08-12 12:36:18,664 - MainThread - urllib3.connectionpool - DEBUG - https://
iam.amazonaws.com:443 "POST / HTTP/1.1" 200 570
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response headers:
{'x-amzn-RequestId': '74c11606-bd38-11e9-9c82-559da0adb349', 'Content-Type': 'text/
xml', 'Content-Length': '570', 'Date': 'Mon, 12 Aug 2019 19:36:18 GMT'}
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response body:
b'<ListGroupResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">\n
<ListGroupResult>\n  <IsTruncated>>false</IsTruncated>\n  <Groups>\n
    <member>\n      <Path>/</Path>\n      <GroupName>MyTestGroup</GroupName>
\n      <Arn>arn:aws:iam::123456789012:group/MyTestGroup</Arn>\n
    <GroupId>AGPA1234567890EXAMPLE</GroupId>\n      <CreateDate>2019-08-12T19:34:04Z</
CreateDate>\n    </member>\n  </Groups>\n </ListGroupResult>\n
  <ResponseMetadata>\n    <RequestId>74c11606-bd38-11e9-9c82-559da0adb349</RequestId>\n
  </ResponseMetadata>\n</ListGroupResponse>\n'
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event needs-
retry.iam.ListGroups: calling handler <botocore.retryhandler.RetryHandler object at
0x7fdf16e9a780>
2019-08-12 12:36:18,665 - MainThread - botocore.retryhandler - DEBUG - No retry needed.
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event after-
call.iam.ListGroups: calling handler <function json_decode_policies at 0x7fdf189b1d90>
{
  "Groups": [

```

```
{
  "Path": "/",
  "GroupName": "MyTestGroup",
  "GroupId": "AGPA123456789012EXAMPLE",
  "Arn": "arn:aws:iam::123456789012:group/MyTestGroup",
  "CreateDate": "2019-08-12T19:34:04Z"
}
]
```

[先頭に戻る](#)

AWS CLI コマンド履歴ログを有効にして確認する

[cli_history](#) ファイル設定を使用して、AWS CLI コマンド履歴ログを有効にできます。この設定を有効にすると、aws コマンドの履歴 AWS CLI を記録します。

aws history list コマンドを使用すると履歴を一覧表示でき、そこから得られた command_id を aws history show コマンドに使用すると詳細を確認できます。詳細については、AWS CLI リファレンスガイドの「[aws history](#)」を参照してください。

--debug オプションを含める場合、詳細の一部には以下が含まれます。

- API botocore に対して行われた呼び出し
- ステータスコード
- HTTP レスポンス
- ヘッダー
- リターンコード

この情報を使用して、パラメータデータとAPI呼び出しが期待どおりに動作していることを確認してから、コマンドが失敗しているプロセスのどのステップで推測できます。

[先頭に戻る](#)

AWS CLI が設定されていることを確認します。

config および credentials ファイル、IAMユーザー、またはロールが正しく設定されていない場合、さまざまなエラーが発生する可能性があります。config および credentials ファイル、IAMユーザー、またはロールによるエラーの解決の詳細については、[the section called “アクセス拒否エラー”](#) および [the section called “無効な認証情報とキーエラー”](#)。

[先頭に戻る](#)

コマンドが見つからないエラー

このエラーは、オペレーティングシステムが AWS CLI コマンドを見つけられないことを意味します。インストールが不完全であるか、更新が必要な可能性があります。

考えられる原因: インストールしたバージョンよりも新しい AWS CLI 機能を使用しようとしているか、フォーマットが正しくない

エラーの例:

```
$ aws s3 copy
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

    aws help
    aws <command> help
    aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                | website
cp                | mv
.....
```

コマンドの形式が正しくない場合や、機能がリリースされる前のバージョンを使用している場合は、さまざまなエラーが発生する可能性があります。これら 2 つの問題のエラーの解決方法の詳細については、「[the section called “AWS CLI コマンドのフォーマットを確認する”](#)」および「[the section called “最新バージョンの AWS CLI を実行していることを確認する”](#)」を参照してください。

[先頭に戻る](#)

考えられる原因: インストール後に端末を再起動する必要がある

エラーの例:

```
$ aws --version
command not found: aws
```

を最初にインストールまたは更新した後に aws コマンドが見つからない場合は AWS CLI、PATH 更新を認識するためにターミナルを再起動する必要がある場合があります。

[先頭に戻る](#)

考えられる原因: が完全にインストール AWS CLI されませんでした

エラーの例:

```
$ aws --version
command not found: aws
```

を最初にインストールまたは更新した後にawsコマンドが見つからない場合は AWS CLI、完全にインストールされていない可能性があります。「[the section called “インストール/更新”](#)」に記載された、プラットフォーム用のステップに従って、再インストールを試してください。

[先頭に戻る](#)

考えられる原因: にはアクセス許可 AWS CLI がありません (Linux)

Linux AWS CLI で を最初にインストールまたは更新した後にawsコマンドが見つからない場合、インストールしたフォルダに対するexecuteアクセス許可がない可能性があります。AWS CLI のインストールPATHに を使用して次のコマンドを実行し、へのアクセス[chmod](#)許可を付与します AWS CLI。

```
$ sudo chmod -R 755 /usr/local/aws-cli/
```

[先頭に戻る](#)

考えられる原因: インストール中にオペレーティングシステムの PATH が更新されなかった。

エラーの例:

```
$ aws --version
command not found: aws
```

オペレーティングシステムの aws 環境変数への PATH 実行ファイルの追加が必要になる場合があります。AWS CLI を に追加するにはPATH、オペレーティングシステムで以下の手順に従ってください。

Linux and macOS

1. ユーザーディレクトリでシェルのプロファイルスクリプトを見つけます。現在使用しているシェルが不明な場合は、echo \$SHELL を実行します。

```
$ ls -a ~
```

```
. .. .bash_logout .bash_profile .bashrc Desktop Documents Downloads
```

- Bash - .bash_profile、.profile、.bash_login
 - Zsh - .zshrc
 - Tcsh - .tcshrc、.cshrc、.login
2. プロファイルスクリプトにエクスポートコマンドを追加します。次のコマンドは、local の bin を現在の PATH 変数に追加します。

```
export PATH=/usr/local/bin:$PATH
```

3. 現在のセッションに更新されたプロファイルをリロードします。

```
$ source ~/.bash_profile
```

Windows

1. Windows コマンドプロンプトで、/R *path* パラメータ で where コマンドを使用して検索する aws ファイルの場所を検索します。結果には、aws を含むすべてのフォルダを返します。

```
C:\> where /R c:\ aws
c:\Program Files\Amazon\AWSCLIV2\aws.exe
...
```

デフォルトでは、AWS CLI バージョン 2 は次の場所にあります。

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
```

2. Windows キーを押し、「**environment variables**」と入力します。
3. 候補のリストから、[Edit environment variables for your account] (アカウントの環境変数を編集する) を選択します。
4. を選択しPATH、編集 を選択します。
5. 最初のステップで見つかったパスを [Variable value] (変数値) フィールドに追加します (例: **C:\Program Files\Amazon\AWSCLIV2\aws.exe**)。
6. [OK] を 2 回選択して、新しい設定を適用します。
7. 実行中のコマンドプロンプトを閉じ、コマンドプロンプトウィンドウを再度開きます。

[先頭に戻る](#)

「aws --version」コマンドが、インストールしたのとは異なるバージョンを返す

ターミナルが PATH に対して想定 AWS CLI とは異なる を返している可能性があります。

考えられる原因: インストール後に端末を再起動する必要がある

aws コマンドで間違ったバージョンが表示される場合は、PATH の更新を認識させるため、端末の再起動が必要である場合があります。アクティブなターミナルだけでなく、開いているすべてのターミナルを閉じる必要があります。

[先頭に戻る](#)

考えられる原因: インストール後にシステムを再起動する必要がある

aws コマンドで間違ったバージョンが表示される場合は、PATH の更新を認識させるため、ターミナルの再起動が必要である場合があります。

[先頭に戻る](#)

考えられる原因: の複数のバージョンがある AWS CLI

を更新 AWS CLI し、既存のインストールとは異なるインストール方法を使用した場合、複数のバージョンがインストールされる可能性があります。例えば、Linux や macOS の場合、現在のインストールでは pip を使用したが、.pkg インストールファイルを使用して更新を試みた場合 (特に PATH が古いバージョンを指している場合)、これにより何らかの競合が発生する可能性があります。

この問題を解決するには、[AWS CLI のすべてのバージョンをアンインストール](#)して、クリーンインストールを実行します。

すべてのバージョンをアンインストールした後、オペレーティングシステム用の手順に従って、[AWS CLI バージョン 1](#) または [AWS CLI バージョン 2](#) の目的のバージョンをインストールします。

i Note

AWS CLI バージョン 1 の既存のインストールで AWS CLI バージョン 2 をインストールした後にこの問題が発生した場合は、[the section called “移行手順”](#)。

[先頭に戻る](#)

aws --version 「」 コマンドは、AWS CLI

これは、システムにまだ AWS CLI がインストールされている場合によく発生します。

考えられる原因: アンインストール後に端末を再起動する必要がある

aws --version コマンドがまだ機能する場合は、端末の更新を認識させるため、端末の再起動が必要な場合があります。

[先頭に戻る](#)

考えられる原因: AWS CLI システムに のバージョンが複数あるか、最初に をインストールしたのと同じアンインストール方法を使用していなかった AWS CLI

のインストールに使用した方法 AWS CLI とは異なる方法で をアンインストールした場合、または複数のバージョンをインストールした場合、 が正しくアンインストールされない AWS CLI 可能性があります。例えば、現在のインストールに pip を使用した場合は、pip を使用してアンインストールする必要があります。これを解決するには、インストールに使用したの AWS CLI と同じ方法でアンインストールします。

1. オペレーティングシステム用の手順、および [AWS CLI バージョン 1](#) と [AWS CLI バージョン 2](#) をアンインストールするための元のインストール方法に従います。
2. 開いているターミナルをすべて閉じます。
3. 目的の端末を開き、次のコマンドを入力して、バージョンが返されないことを確認します。

```
$ aws --version
command not found: aws
```

出力にまだバージョンがリストされている場合、 は別の方法でインストール AWS CLI されているか、複数のバージョンがある可能性が高くなります。 をインストールした方法がわからない場合は AWS CLI、バージョン出力が受信されるまで、オペレーティングシステムに適した [AWS CLI バージョン 1](#) と [AWS CLI バージョン 2](#) の各アンインストール方法の手順に従います。

Note

パッケージマネージャーを使用して AWS CLI をインストールした場合 (pip、apt、brew など) は、同じパッケージマネージャーを使用してアンインストールする必要があります。パッケージのすべてのバージョンをアンインストールする方法については、パッケージマネージャーの手順に従ってください。

[先頭に戻る](#)

は、パラメータ名が不完全なコマンド AWS CLI を処理しました

考えられる原因: AWS CLI パラメータの認識されている省略形を使用しました

AWS CLI は Python を使用して構築されているため、は [allow_abbrev](#) 引数を含む Python argparse ライブラリ AWS CLI を使用します。パラメータの略語は によって認識 AWS CLI され、処理されます。

次の [create-change-set](#) コマンド例では、CloudFormation スタック名を変更します。パラメータ `--change-set-n` は の略語として認識され `--change-set-name`、コマンド AWS CLI を処理します。

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-n my-change-set
```

省略形が複数のコマンドに該当する場合、パラメータは省略形として認識されません。

次の [create-change-set](#) コマンド例では、CloudFormation スタック名を変更します。パラメータ `--change-set-` は、省略形として認識されません。これは、`--change-set-name` や `--change-set-type` など、複数のパラメータに該当する場合があります。したがって、AWS CLI は コマンドを処理しません。

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set- my-change-set
```

⚠ Warning

パラメータの省略形を意図的に使用しないでください。これらは信頼性が低く、下位互換性也没有ありません。省略形の混乱を生じるような新しいパラメータをコマンドに追加すると、コマンドが壊れてしまいます。

さらに、パラメータが単一値の引数である場合、コマンドで予期しない動作が発生する可能性があります。単一値の引数のインスタスを複数渡すと、最後のインスタスのみが実行されます。次の例で、パラメータ `--filters` は単一値の引数です。パラメータ `--filters` とパラメータ `--filter` が指定されています。`--filter` パラメータは `--filters` の省略形です。これにより、`--filters` のインスタスが 2 つ適用されることになるため、最後の `--filter` 引数のみが実行されます。

```
$ aws ec2 describe-vpc-peering-connections \  
  --filters Name=tag:TagName,Values=VpcPeeringConnection \  
  --filter Name=status-code,Values=active
```

予期しない動作を防ぐために、コマンドを実行する前に有効なパラメータを使用していることを確認してください。

[先頭に戻る](#)

アクセス拒否エラー

考えられる原因: AWS CLI プログラムファイルに「実行」アクセス許可がありません

Linux または macOS で、aws プログラムに呼び出し元ユーザーの実行許可があることを確認します。通常、この権限は 755 に設定されます。

ユーザーに実行アクセス許可を追加するには、次のコマンドを実行し、`aws` を置き換えます。`~/.local/bin/aws` コンピュータ上のプログラムへのパス。

```
$ chmod +x ~/.local/bin/aws
```

[先頭に戻る](#)

考えられる原因: ID に オペレーションを実行するアクセス許可IAMがない

エラーの例:

```
$ aws s3 ls
```

```
An error occurred (AccessDenied) when calling the ListBuckets operation: Access denied.
```

AWS CLI コマンドを実行すると、ユーザーをIAMアカウントまたはロールに関連付ける認証情報を使用して、ユーザーに代わってAWS操作が実行されます。アタッチされたポリシーは、で実行するコマンドに対応するAPIアクションを呼び出すアクセス許可を付与する必要がありますAWS CLI。

ほとんどのコマンドは、コマンド名と一致する名前を指定して1つのアクションを呼び出します。ただし、などのカスタムコマンドは複数のをaws s3 sync呼び出しますAPIs。--debugオプションを使用して、どのコマンドAPIsが呼び出すかを確認できます。

ユーザーまたはロールにポリシーによって割り当てられた適切なアクセス許可があることを確認した場合は、AWS CLI コマンドが想定している認証情報を使用していることを確認してください。[認証情報に関する次のセクション](#)を参照して、AWS CLI が使用している認証情報が想定どおりであることを確認します。

アクセスIAM許可の割り当ての詳細については、IAM ユーザーガイドの[アクセス管理の概要: アクセス許可とポリシー](#)を参照してください。

[先頭に戻る](#)

無効な認証情報とキーエラー

エラーの例:

```
$ aws s3 ls
```

```
An error occurred (InvalidAccessKeyId) when calling the ListBuckets operation: The AWS Access Key Id you provided does not exist in our records.
```

```
$ aws s3 ls
```

```
An error occurred (InvalidClientTokenId) when calling the ListBuckets operation: The security token included in the request is invalid.
```

考えられる原因: AWS CLI が誤った認証情報を読み取っているか、予期しない場所から読み取っている

が想定とは異なる場所から認証情報を読み取っているか、キーペア情報が正しくない AWS CLI 可能性があります。aws configure list を実行して、使用される認証情報を確認することができます。

次の例は、デフォルトのプロファイルに使用される認証情報をチェックする方法を示しています。

```
$ aws configure list
  Name                               Value                               Type    Location
  ----                               -
  profile                             <not set>                           None    None
  access_key                          *****XYVA                          shared-credentials-file
  secret_key                          *****ZAGY                          shared-credentials-file
  region                               us-west-2                             config-file  ~/.aws/config
```

次の例は、名前付きプロファイルの認証情報をチェックする方法を示しています。

```
$ aws configure list --profile saanvi
  Name                               Value                               Type    Location
  ----                               -
  profile                             saanvi                             manual  --profile
  access_key                          *****                          shared-credentials-file
  secret_key                          *****                          shared-credentials-file
  region                               us-west-2                             config-file  ~/.aws/config
```

キーペアの詳細を確認するには、config および credentials ファイルを確認します。config ファイルと credentials ファイルの詳細については、「[the section called “の設定と認証情報ファイルの設定 AWS CLI”](#)」を参照してください。認証情報の優先順位を含む、認証情報と認証の詳細については、「[認証とアクセス認証情報](#)」を参照してください。

[先頭に戻る](#)

考えられる原因: コンピュータのクロックが同期していない

有効な認証情報を使用している場合は、クロックが同期していない可能性があります。Linux または macOS では、date を実行して時刻を確認します。

```
$ date
```

システムクロックのずれが数分以内の場合は、`ntpd` を使用して同期します。

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Windows では、コントロールパネルで日付と時刻オプションを使用してシステムクロックを設定します。

[先頭に戻る](#)

署名がエラーと一致しない

エラーの例:

```
$ aws s3 ls
An error occurred (SignatureDoesNotMatch) when calling the ListBuckets operation: The
request signature we
calculated does not match the signature you provided. Check your key and signing
method.
```

がコマンド AWS CLI を実行すると、暗号化されたリクエストを AWS サーバーに送信して、適切な AWS サービスオペレーションを実行します。認証情報 (アクセスキーとシークレットキー) は暗号化に参与し、 がリクエストを行った人物を認証 AWS できるようにします。このプロセスが正しく行われない原因には、次のようにいくつかあります。

考えられる原因: クロックが AWS サーバーと同期していない

[リプレイ攻撃](#)への保護対策として、現在の時刻を暗号化/復号プロセス中に使用できます。クライアントとサーバーの時間に許容範囲外の差がある場合、プロセスが失敗し、リクエストが拒否される可能性があります。このエラーは、クロックがホストマシンのクロックと同期していない仮想マシンでコマンドを実行した場合にも発生します。考えられる原因の 1 つは、仮想マシンが休止した後、ウェイクアップしてからホストマシンとクロックを同期するまで時間がかかる場合です。

Linux または macOS では、`date` を実行して時刻を確認します。

```
$ date
```

システムクロックのずれが数分以内の場合は、`ntpd` を使用して同期します。

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Windows では、コントロールパネルで日付と時刻オプションを使用してシステムクロックを設定します。

[先頭に戻る](#)

考えられる原因: オペレーティングシステムが特定の特殊文字を含む AWS キーを誤って処理している

AWS キーに `-`、`,`、`'` などの特定の特殊文字が含まれている場合%、一部のオペレーティングシステムバリエーションは文字列を不適切に処理し+/`'`、キー文字列が正しく解釈されない原因となります。

作成の一環として認証情報ファイルを新しいインスタンスに構築するツールなど、他のツールやスクリプトを使用してキーを処理する場合、これらのツールやスクリプトは特殊文字を独自に処理して、認識 AWS されなくなったものに変換される可能性があります。

シークレットキーを再生成して、問題の発生原因となっている特殊文字を含まないキーを取得することをお勧めします。

[先頭に戻る](#)

SSL 証明書エラー

考えられる原因: AWS CLI はプロキシの証明書を信頼していません

エラーの例:

```
$ aws s3 ls
[SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed
```

AWS CLI コマンドを使用すると、`[SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed` エラーメッセージが表示されます。これは、プロキシの証明書が自己署名され、会社が認証局 (CA) に設定されているなどの要因により、プロキシの証明書が信頼 AWS CLI

されないことが原因です。これにより、ローカル CA レジストリで会社の CA ルート証明書を検索 AWS CLI できなくなります。

これを修正するには、[ca_bundle](#) 設定 .pem ファイル 設定、[--ca-bundle](#) コマンドライン オプション、または [AWS_CA_BUNDLE](#) 環境変数を使用して、会社ファイル AWS CLI の場所を に指示します。

[先頭に戻る](#)

考えられる原因: 設定が正しい CA ルート証明書の場所を指していない

エラーの例:

```
$ aws s3 ls
SSL validation failed for regionname [Errno 2] No such file or directory
```

これは、認証局 (CA) バンドルファイルの場所が、AWS CLI で正しく設定されていないことによって発生します。この問題を解決するには、企業の .pem ファイルの場所を確認し、[ca_bundle](#) 設定ファイルの設定、[--ca-bundle](#) コマンドライン オプション、または [AWS_CA_BUNDLE](#) 環境変数を使用して、AWS CLI の設定を更新します。

[先頭に戻る](#)

考えられる原因: 設定で正しい が使用されていない AWS リージョン

エラーの例:

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

が指定された で使用 AWS のサービス できない場合、またはリソースが別の がある場合、エラー AWS リージョン や予期しない結果が発生する可能性があります AWS リージョン。トラブルシューティングステップについては、「[the section called “AWS CLI コマンドが 使用している AWS リージョン ことを確認します。”](#)」を参照してください。

[先頭に戻る](#)

考えられる原因: TLSバージョンを更新する必要があります

エラーの例:

```
$ aws s3 ls
```

```
[SSL: UNSAFE_LEGACY_RENEGOTIATION_DISABLED] unsafe legacy renegotiation disabled
```

AWS のサービスは、デバイスのバージョンと互換性 TLS のない TLS のバージョンを使用しています。この問題を解決するには、サポートされている TLS バージョンに更新します。詳細については、「[the section called “最小 TLS バージョンの強制”](#)」を参照してください。

[先頭に戻る](#)

無効な JSON エラー

エラーの例:

```
$ aws dynamodb update-table \  
  --provisioned-throughput '{"ReadCapacityUnits':15,WriteCapacityUnits':10}' \  
  --table-name MyDDBTable  
Error parsing parameter '--provisioned-throughput': Invalid JSON: Expecting property  
  name enclosed in  
double quotes: line 1 column 25 (char 24)  
JSON received: {"ReadCapacityUnits":15,WriteCapacityUnits":10}
```

AWS CLI コマンドを使用すると、Invalid JSON 「」エラーメッセージが表示されます。これは通常、想定された JSON 形式でコマンドを入力するとエラーが発生し、JSON が正しく読み取れ AWS CLI ません。

考えられる原因: AWS CLI が使用する JSON 有効な が入力されませんでした

コマンドに有効な JSON が入力されていることを確認します。フォーマットに問題がある JSON 場合は、JSON バリデーターを使用することをお勧めします。

コマンドラインでより高度な JSON 使用法を得るには、などのコマンドライン JSON プロセッサを使用して JSON 文字列を作成する jq ことを検討してください。の詳細については jq、「」の「[jq リポジトリ](#)」を参照してください GitHub。

[先頭に戻る](#)

考えられる原因: ターミナルのクォーティングルールにより、への有効な送信が妨げ JSON られている AWS CLI

がコマンドから何か AWS CLI を受信する前に、ターミナルは独自のクォーティングルールとエスケープルールを使用してコマンドを処理します。ターミナルのフォーマットルールにより、コ

マンドが に渡される前に一部のJSONコンテンツが削除される場合があります AWS CLI。コマンドを作成するときは、必ず [ターミナルの引用ルール](#) を使用してください。

トラブルシューティングを行うには、echo コマンドを使用して、シェルがパラメータをどのように処理しているかを確認します。

```
$ echo {"ReadCapacityUnits":15,"WriteCapacityUnits":10}
ReadCapacityUnits:15 WriteCapacityUnits:10
```

```
$ echo '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}'
{"ReadCapacityUnits":15,"WriteCapacityUnits":10}
```

有効な が返されるまで、コマンドを変更しますJSON。

詳細なトラブルシューティングについては、--debug パラメータを使用して、AWS CLIに渡された内容を正確に表示するデバッグログを表示します。

```
$ aws dynamodb update-table \
  --provisioned-throughput '{"ReadCapacityUnits":15,WriteCapacityUnits":10}' \
  --table-name MyDDBTable \
  --debug
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - CLI version: aws-
cli/1.18.147
Python/2.7.18 Linux/5.4.196-119.356.amzn2int.x86_64 botocore/1.18.6
2022-07-19 22:25:07,741 - MainThread - awscli.clidriver - DEBUG - Arguments entered
to CLI:
['dynamodb', 'update-table', '--provisioned-throughput',
 '{"ReadCapacityUnits":15,WriteCapacityUnits":10}',
 '--table-name', 'MyDDBTable', '--debug']
```

ターミナルのクォーティングルールを使用して、への送信時にJSON入力が発生する問題を修正します AWS CLI。引用ルールの詳細については、「[the section called “文字列を含む引用符”](#)」を参照してください。

Note

JSON で を有効にできない場合は AWS CLI、Blobs を使用してJSONデータをJSONに直接渡すことで、ターミナルのデータ入力の引用ルールをバイパスすることをお勧めします AWS CLI。BLOB の詳細については、「[blob](#)」を参照してください。

[先頭に戻る](#)

追加リソース

AWS CLI 問題に関するその他のヘルプについては、GitHubまたは [AWS re:PostAWS CLI コミュニティ](#) を参照してください。

[先頭に戻る](#)

AWS CLI バージョン 1 から AWS CLI バージョン 2 への移行

このセクションでは、AWS CLI バージョン 1 を AWS CLI バージョン 2 に更新する手順について説明します。AWS CLI バージョン 2 は AWS CLI バージョン 1 に基づいて構築され、コミュニティのフィードバックに基づく機能と機能強化が含まれています。AWS CLI バージョン 2 はの最新バージョン AWS CLI であり、すべての最新機能をサポートしています。バージョン 2 で導入された一部の機能は、バージョン 1 との下位互換性がないため、これらの機能にアクセスするには、アップグレードする必要があります。

予期しない問題を回避するには、バージョン 2 に移行する前に、[バージョンの違いについて説明します](#)。AWS CLI バージョン 2 には、下位互換性のためにスクリプトまたはコマンドを更新する必要がある可能性のある新機能と変更が含まれています。

AWS CLI バージョン 1 と 2 は同じ aws コマンド名を使用します。両方のバージョンがインストールされている場合、コンピュータによって検索パスで最初に見つかったバージョンが使用されます。これにより、新しいバージョンがインストールされている場合でも、aws コマンド名が古い AWS CLI バージョンを呼び出す可能性があります。

AWS CLI バージョン 1 を以前にインストールした場合は、「」の手順に従います [the section called “移行手順”](#)。

AWS CLI バージョン 1 を以前にインストールしていない場合は、「」の手順に従います [使用を開始する](#)。

トピック

- [AWS CLI バージョン 2 の新機能と変更点](#)
- [AWS CLI バージョン 1 から AWS CLI バージョン 2 をインストールする](#)

AWS CLI バージョン 2 の新機能と変更点

このトピックでは、AWS CLI バージョン 1 と AWS CLI バージョン 2 の動作に関する新機能と変更について説明します。これらの変更では、バージョン 1 と同じ動作をバージョン 2 で実行するために、スクリプトまたはコマンドを更新する必要がある場合があります。

トピック

- [AWS CLI バージョン 2 の新機能](#)
- [AWS CLI バージョン 1 と AWS CLI バージョン 2 の変更のブレイク](#)

AWS CLI バージョン 2 の新機能

AWS CLI バージョン 2 はの最新バージョン AWS CLI であり、すべての最新機能をサポートしています。バージョン 2 で導入された一部の機能は、バージョン 1 との下位互換性がないため、これらの機能にアクセスするには、アップグレードする必要があります。主な機能は以下のとおりです。

Python インタプリタは不要です

AWS CLI バージョン 2 では、Python を別途インストールする必要はありません。組み込みバージョンが含まれています。

[ウィザード](#)

AWS CLI ウィザードはバージョン 2 で使用できます。ウィザードの指示に従って、特定のコマンドを作成できます。

[IAM Identity Center 認証](#)

組織が AWS IAM Identity Center (IAM Identity Center) を使用している場合、ユーザーは Active Directory、組み込みの IAM Identity Center ディレクトリ、または [IAM Identity Center に接続された別の IdP](#) にサインインできます。次に、AWS CLI コマンドの実行を許可する (IAM) AWS Identity and Access Management ロールにマッピングされます。

[自動プロンプト](#)

有効にすると、AWS CLI バージョン 2 では、コマンドの実行時に aws コマンド、パラメータ、リソースの入力を求めるプロンプトが表示されます。

[の公式 Amazon ECR Public または Docker イメージの実行 AWS CLI](#)

の公式 Docker イメージ AWS CLI は、AWS 直接サポートおよび維持する分離、移植性、セキュリティを提供します。これにより、自分でインストールを管理することなく、コンテナベースの環境で AWS CLI バージョン 2 を使用できます。

[クライアント側のページャー](#)

AWS CLI バージョン 2 では、出力にクライアント側のポケットベルプログラムを使用できます。デフォルトでは、この機能がオンになり、オペレーティングシステムのデフォルトのページャープログラムを介してすべての出力を返します。

[aws configure import](#)

AWS Management Consoleから生成された重要な .csv 認証情報をインポートします。 .csv ファイルは、IAMユーザー名と一致するプロファイル名でインポートされます。

[aws configure list-profiles](#)

設定したすべてのプロファイルの名前を一覧表示します。

[the section called “YAML ストリーム出力形式”](#)

yaml および yaml-stream形式は、 [YAML](#)形式を利用すると同時に、データをストリーミングすることで、大きなデータセットをより迅速に表示できます。クエリ全体がダウンロードされる前に、YAMLデータの表示と使用を開始できます。

[DynamoDB 用の新しい高レベルの ddb コマンド](#)

AWS CLI バージョン 2 には、高レベルの Amazon DynamoDB コマンド [ddb put](#)と [ddb select](#)があります。これらのコマンドは、DynamoDB テーブルに項目を配置し、DynamoDB テーブルまたはインデックスを検索するためのシンプルなインターフェイスを提供します。

[aws logs tail](#)

AWS CLI バージョン 2 には、Amazon CloudWatch Logs グループのログを格納するカスタム `aws logs tail` コマンドがあります。デフォルトでは、コマンドは過去 10 分間に、関連付けられたすべての CloudWatch Logs ストリームからログを返します。

[高レベルの s3 コマンドに対するメタデータのサポートを追加](#)

AWS CLI バージョン 2 では、高レベル `s3` コマンドに `--copy-props` パラメータが追加されます。このパラメータを使用すると、Amazon Simple Storage Service (Amazon S3) の追加のメタデータとタグを設定できます。

[AWS_REGION](#)

AWS CLI バージョン 2 には、という AWS SDK互換の環境変数があります `AWS_REGION`。この変数は `AWS リージョン`、リクエストを送信する を指定します。これは、 `AWS_DEFAULT_REGION` 環境変数を上書きします (AWS CLIにのみに適用されます)。

AWS CLI バージョン 1 と AWS CLI バージョン 2 の変更のブレーク

このセクションでは、AWS CLI バージョン 1 と AWS CLI バージョン 2 の動作のすべての変更について説明します。これらの変更では、バージョン 1 と同じ動作をバージョン 2 で実行するために、スクリプトまたはコマンドを更新する必要がある場合があります。

トピック

- [テキストファイルのエンコードを設定するために環境変数を追加](#)
- [バイナリパラメータはデフォルトで base64 エンコードされた文字列として渡されるようになりました](#)
- [マルチパートコピーのファイルプロパティとタグの Amazon S3 処理が改善されました](#)
- [パラメータの http://または https:// URLs の自動取得なし](#)
- [デフォルトですべての出力に使用されるページャー](#)
- [タイムスタンプ出力値は 8601 ISO 形式に標準化されています](#)
- [変更のない CloudFormation デプロイの処理を改善](#)
- [us-east-1 リージョンの Amazon S3 エンドポイントのデフォルト動作を変更しました](#)
- [リージョン AWS STS エンドポイントのデフォルト動作を変更](#)
- [ecr get-login を削除し、ecr get-login-password に置き換えました](#)
- [AWS CLI プラグインのバージョン 2 のサポートが変更されています](#)
- [非表示エイリアスのサポートを削除しました](#)
- [api_versions 設定ファイルの設定はサポートされていません](#)
- [AWS CLI バージョン 2 では、署名 v4 のみを使用して Amazon S3 リクエストを認証します](#)
- [AWS CLI バージョン 2 はページングパラメータとより一貫しています](#)
- [AWS CLI バージョン 2 は、すべてのコマンドでより一貫したリターンコードを提供します。](#)

テキストファイルのエンコードを設定するために環境変数を追加

デフォルトでは、[the section called “blob”](#) のテキストファイルはインストールされたロケールと同じエンコードを使用します。AWS CLI バージョン 2 は Python の埋め込みバージョンを使用しているため、PYTHONUTF8および PYTHONIOENCODING環境変数はサポートされていません。テキストファイルのエンコードをロケールと異なるように設定するには、AWS_CLI_FILE_ENCODING 環境変数を使用します。次の例では、Windows UTF-8で を使用してテキストファイルを開く AWS CLI ように を設定します。

```
AWS_CLI_FILE_ENCODING=UTF-8
```

詳細については、「[の環境変数の設定 AWS CLI](#)」を参照してください。

バイナリパラメータはデフォルトで base64 エンコードされた文字列として渡されるようになりました

では AWS CLI、一部のコマンドには [base64](#) でエンコードされた文字列が必要で、他のコマンドには UTF-8 でエンコードされたバイト文字列が必要でした。AWS CLI バージョン 1 では、エンコードされた 2 つの文字列タイプ間でデータを渡すには、多くの場合、中間処理が必要です。AWS CLI バージョン 2 では、バイナリパラメータの処理の一貫性が向上し、あるコマンドから別のコマンドに値をより確実に渡すことができます。

デフォルトでは、AWS CLI バージョン 2 はすべてのバイナリ入力パラメータとバイナリ出力パラメータを base64 でエンコードされた文字列 blobs (バイナリラージオブジェクト) として渡します。詳細については、「[the section called “blob”](#)」を参照してください。

AWS CLI バージョン 1 の動作に戻すには、[cli_binary_format](#) ファイル設定または `--cli-binary-format` パラメータを使用します。

マルチパートコピーのファイルプロパティとタグの Amazon S3 処理が改善されました

`aws s3` 名前空間で AWS CLI バージョン 1 コマンドを使用して S3 バケットの場所から別の場所にファイルをコピーし、そのオペレーションで [マルチパートコピー](#) を使用する場合、ソースオブジェクトからのファイルプロパティはコピー先オブジェクトにコピーされません。

デフォルトでは、AWS CLI バージョン 2 の対応するコマンドは、すべてのタグと一部のプロパティをソースからコピー先に転送します。AWS CLI バージョン 1 と比較すると、Amazon S3 エンドポイントへの呼び出しが増える AWS API 可能性があります。AWS CLI バージョン 2 の `s3` コマンドのデフォルトの動作を変更するには、`--copy-props` パラメータを使用します。

詳細については、「[the section called “マルチパートコピーのファイルプロパティとタグ”](#)」を参照してください。

パラメータの `http://` または `https://` URLs の自動取得なし

AWS CLI バージョン 2 は、パラメータ値が `http://` または `https://` で始まる場合、GET オペレーションを実行せず `https://`、返されたコンテンツをパラメータ値として使用しません。その結果、関連するコマンドラインオプション `cli_follow_urlparam` が AWS CLI バージョン 2 から削除されます。

を取得し、URL 内容をパラメータ値に URL 渡す必要がある場合は、`curl` または同様のツールを使用して、の内容を URL ローカルファイルにダウンロードすることをお勧めします。次に、`file://` 構文を使用してそのファイルの内容を読み込み、パラメータの値として使用します。

例えば、次のコマンドでは、`http://www.example.com` で見つかったページの内容を取得し、その内容をパラメータとして渡そうとしなくなりました。代わりに、リテラルテキスト文字列 `https://example.com` をパラメータとして渡します。

```
$ aws ssm put-parameter \  
  --value http://www.example.com \  
  --name prod.microservice1.db.secret \  
  --type String 2
```

ウェブの内容を取得してパラメータURLとして使用する必要がある場合は、バージョン 2 で以下を実行できます。

```
$ curl https://my.example.com/mypolicyfile.json -o mypolicyfile.json  
$ aws iam put-role-policy \  
  --policy-document file:///./mypolicyfile.json \  
  --role-name MyRole \  
  --policy-name MyReadOnlyPolicy
```

前の例では、`-o` パラメータは、ソースファイルと同じ名前で現在のフォルダにファイルを保存するよう `curl` に指示します。2 番目のコマンドは、ダウンロードしたファイルの内容を取得し、その内容を `--policy-document` の値として渡します。

デフォルトですべての出力に使用されるページャー

デフォルトでは、AWS CLI バージョン 2 はオペレーティングシステムのデフォルトのページャープログラムを通じてすべての出力を返します。このプログラムは Linux および macOS では [less](#) プログラム、Windows では [more](#) プログラムです。これにより、出力を一度に 1 ページずつ表示することで、サービスからの大量の出力内を移動できます。

AWS CLI バージョン 2 は、別のページングプログラムを使用するか、まったく使用しないように設定できます。詳細については、「[the section called “クライアント側のページャー”](#)」を参照してください。

タイムスタンプ出力値は 8601 ISO 形式に標準化されています

デフォルトでは、AWS CLI バージョン 2 はすべてのタイムスタンプレスポンス値を [ISO 8601 形式](#) で返します。AWS CLI バージョン 1 では、コマンドは HTTP API レスポンスによって返された任意の形式のタイムスタンプ値を返しました。これはサービスごとに異なる場合があります。

HTTP API レスポンスによって返される形式のタイムスタンプを表示するには、`config` ファイルの `wire` 値を使用します。詳細については、「[cli_timestamp_format](#)」を参照してください。

変更のない CloudFormation デプロイの処理を改善

AWS CLI バージョン 1 では、デフォルトで変更のない AWS CloudFormation テンプレートをデプロイすると、失敗したエラーコード AWS CLI を返します。これをエラーと見なさずにスクリプトを続行すると、問題が発生する可能性があります。AWS CLI をバージョン 1 で回避するには `--no-fail-on-empty-changeset`、を返すフラグを追加します。

これは一般的なユースケースであるため、デプロイによる変更がなく、オペレーションが空の変更セットを返す場合、AWS CLI バージョン 2 はデフォルトでの正常な終了コードを返すこととなります。

元の動作に戻すには、フラグ `--fail-on-empty-changeset` を追加します。

us-east-1 リージョンの Amazon S3 エンドポイントのデフォルト動作を変更しました

us-east-1 リージョンを使用するように AWS CLI バージョン 1 を設定すると、AWS CLI は us-east-1 リージョンで物理的にホストされているグローバル `s3.amazonaws.com` エンドポイントを使用します。AWS CLI バージョン 2 は、そのリージョンが指定され `s3.us-east-1.amazonaws.com` ている場合、真のリージョンエンドポイントを使用します。AWS CLI バージョン 2 に強制的にグローバルエンドポイントを使用するには、コマンドのリージョンを に設定します `aws-global`。

リージョン AWS STS エンドポイントのデフォルト動作を変更

デフォルトでは、AWS CLI バージョン 2 は、現在設定されている のすべての AWS Security Token Service (AWS STS) API リクエストをリージョンエンドポイントに送信します AWS リージョン。

デフォルトでは、AWS CLI バージョン 1 はグローバル AWS STS エンドポイントに AWS STS リクエストを送信します。このバージョン 1 のデフォルト動作は、[sts_regional_endpoints](#) 設定を使用して制御できます。

ecr get-login を削除し、ecr get-login-password に置き換えました

AWS CLI バージョン 2 は、コマンドを、コンテナ認証 `aws ecr get-login` との自動統合を改善する `aws ecr get-login-password` コマンドに置き換えます。

`aws ecr get-login-password` コマンドでは、プロセスリスト、シエル履歴、またはその他のログファイル内の認証情報が公開されるリスクが減ります。また、`docker login` コマンドとの互換性が向上し、オートメーションが向上します。

`aws ecr get-login-password` コマンドは、AWS CLI バージョン 1.17.10 以降、および AWS CLI バージョン 2 で使用できます。以前の `aws ecr get-login` コマンドは、下位互換性のために AWS CLI バージョン 1 でも引き続き使用できます。

`aws ecr get-login-password` コマンドを使用すると、パスワードを取得する以下のコードを置き換えることができます。

```
$ (aws ecr get-login --no-include-email)
```

パスワードをシェルの履歴またはログに公開するリスクを減らすには、代わりに以下の例のコマンドを使用します。この例では、パスワードは `docker login` コマンドに直接パイプされ、そこで `--password-stdin` オプションによってパスワードパラメータに割り当てられます。

```
$ aws ecr get-login-password | docker login --username AWS --password-stdin MY-REGISTRY-URL
```

詳細については、AWS CLI バージョン 2 リファレンスガイドの「[aws ecr get-login-password](#)」を参照してください。

AWS CLI プラグインのバージョン 2 のサポートが変更されています

AWS CLI バージョン 2 のプラグインサポートは完全に暫定的であり、ユーザーが AWS CLI バージョン 1 から安定した更新されたプラグインインターフェイスがリリースされるまで移行するのを支援することを目的としています。特定のプラグイン、または AWS CLI プラグインインターフェイスが AWS CLI バージョン 2 の将来のバージョンでサポートされる保証はありません。プラグインを使用する場合は、アップグレード時に の特定のバージョンにロック AWS CLI し、プラグインの機能をテストしてください。

プラグインサポートを有効にするには、`[plugins]` に `~/.aws/config` セクションを作成します。

```
[plugins]
cli_legacy_plugin_path = <path-to-plugins>/python3.7/site-packages
<plugin-name> = <plugin-module>
```

`[plugins]` セクションで、`cli_legacy_plugin_path` 変数を定義し、その値を、プラグインモジュールがある Python サイトパッケージのパスに設定します。次に、プラグインの名前 (`plugin-name`)、およびプラグインのソースコードを含む Python モジュールのファイル名

(plugin-module) を指定して、プラグインを設定できます。は、プラグインをインポート plugin-module して awscli_initialize 関数を呼び出すことで、各プラグインを AWS CLI ロードします。

非表示エイリアスのサポートを削除しました

AWS CLI バージョン 2 では、バージョン 1 でサポートされている以下の非表示エイリアスがサポートされなくなりました。

次の表では、最初の列に、バージョン 2 AWS CLI を含むすべてのバージョンで動作するサービス、コマンド、およびパラメータが表示されます。2 番目の列には、AWS CLI バージョン 2 で機能しなくなったエイリアスが表示されます。

作業サービス、コマンド、パラメータ	廃止されたエイリアス
cognito-identity create-identity-pool open-id-connect-provider-arns	open-id-connect-provider-arns
storagegateway describe-tapes tape-arns	tape-arns
storagegateway.describe-tape-archives.tape-arns	tape-arns
storagegatewaydescribe-vtl-devices。vtl-device-arns	vtl-device-arns
storagegateway.describe-cached-iscsi-volumes.volume-arns	volume-arns
storagegateway.describe-stored-iscsi-volumes.volume-arns	volume-arns
route53domains.view-billing.start-time	start
deploy.create-deployment-group.ec2-tag-set	ec-2-tag-set
deploy.list-application-revisions.s3-bucket	s-3-bucket
deploy.list-application-revisions.s3-key-prefix	s-3-key-prefix
deploy.update-deployment-group.ec2-tag-set	ec-2-tag-set
iam.enable-mfa-device.authentication-code1	authentication-code-1
iam.enable-mfa-device.authentication-code2	authentication-code-2

作業サービス、コマンド、パラメータ	廃止されたエイリアス
iam.resync-mfa-device.authentication-code1	authentication-code-1
iam.resync-mfa-device.authentication-code2	authentication-code-2
importexport.get-shipping-label.street1	street-1
importexport.get-shipping-label.street2	street-2
importexport.get-shipping-label.street3	street-3
lambda.publish-version.code-sha256	code-sha-256
lightsailimport-key-pair..public-key-base64	public-key-base-64
opsworks.register-volume.ec2-volume-id	ec-2-volume-id

api_versions 設定ファイルの設定はサポートされていません

AWS CLI バージョン 2 では、api_versions 設定ファイル設定 APIs を使用した以前のバージョンの AWS サービスの呼び出しはサポートされていません。すべての AWS CLI コマンド APIs が、エンドポイントで現在サポートされているサービスの最新バージョンを呼び出すようになりました。

AWS CLI バージョン 2 では、署名 v4 のみを使用して Amazon S3 リクエストを認証します

AWS CLI バージョン 2 は、Amazon S3 エンドポイントに送信されたサービスリクエストを暗号化して認証するための以前の署名アルゴリズムをサポートしていません。この署名は、すべての Amazon S3 リクエストで自動的に行われます。また、[署名バージョン 4 の署名プロセス](#)のみがサポートされています。署名バージョンを設定することはできません。すべての Amazon S3 バケット署名で SigV4 のみが使用され、最大有効期限は 1 週間URLs になりました。

AWS CLI バージョン 2 はページングパラメータとより一貫しています

AWS CLI バージョン 1 では、コマンドラインでページ分割パラメータを指定すると、自動ページ分割は期待どおりにオフになります。ただし、--cli-input-json パラメータを指定したファイルを使用してページ分割パラメータを指定する場合、自動ページ分割がオフになっていないため、予期しない出力が発生する可能性があります。AWS CLI バージョン 2 では、パラメータを指定する方法に関係なく、自動ページ分割がオフになります。

AWS CLI バージョン 2 は、すべてのコマンドでより一貫したリターンコードを提供します。

AWS CLI バージョン 2 はすべてのコマンドで一貫性が高く、AWS CLI バージョン 1 と比較して適切な終了コードを適切に返します。また、終了コード 252、253、254 を追加しました。終了コードの詳細については、「[the section called “リターンコード”](#)」を参照してください。

AWS CLI バージョン 1 がリターンコード値をどのように使用するかに依存している場合は、終了コードを確認して、期待する値を取得していることを確認することをお勧めします。

AWS CLI バージョン 1 から AWS CLI バージョン 2 をインストールする

このトピックでは、AWS CLI バージョン 1 から AWS CLI バージョン 2 に移行する手順について説明します。

AWS CLI バージョン 1 と 2 は同じawsコマンド名を使用します。両方のバージョンがインストールされている場合、コンピュータによって検索パスで最初に見つかったバージョンが使用されます。AWS CLI バージョン 1 を以前にインストールした場合は、AWS CLI バージョン 2 を使用するには、次のいずれかを実行することをお勧めします。

- 推奨 – [AWS CLI バージョン 1 をアンインストールし、AWS CLI バージョン 2 のみを使用します](#)。
- [両方のバージョンをインストールするには](#)、オペレーティングシステムの機能を使用して、2 つの aws コマンドのいずれかに対して異なる名前でもシンボリックリンクまたはエイリアスを作成します。

バージョン 1 とバージョン 2 との間の重要な変更については、「[the section called “新機能と変更点”](#)」を参照してください。

バージョン 1 をバージョン 2 に置き換える

AWS CLI バージョン 1 をバージョン 2 AWS CLI に置き換えるには、次の手順を実行します。

AWS CLI バージョン 1 を AWS CLI バージョン 2 に置き換えるには

1. 「[the section called “新機能と変更点”](#)」で、バージョン 1 とバージョン 2 との間の重要な変更を確認して、移行用の既存のスクリプトを準備します。

2. [AWS CLI バージョン 1 のインストール、更新、アンインストールのオペレーティングシステムのアンインストール手順に従って、AWS CLI バージョン 1 をアンインストール](#)します。
3. 次のコマンドを使用して、AWS CLI が完全にアンインストールされていることを確認します。

```
$ aws --version
```

出力に基づいて、次のいずれかを実行します。

- バージョンが返されません：AWS CLI バージョン 1 が正常にアンインストールされ、次のステップに進むことができます。
 - バージョンが返されます：AWS CLI バージョン 1 はまだインストールされています。トラブルシューティングステップについては、「[the section called “aws --version” コマンドは、AWS CLI](#)」を参照してください。バージョンが出力されなくなるまで、トラブルシューティングステップを実行します。
4. 「」のオペレーティングシステムの適切なインストール手順に従って、AWS CLI バージョン 2 をインストールします。[の最新バージョンへのインストールまたは更新 AWS CLI](#)。

Side-by-side インストール

両方のバージョンをインストールするには、オペレーティングシステムの機能を使用して、2 つの aws コマンドのいずれかに対して異なる名前でもシンボリックリンクまたはエイリアスを作成します。

1. 「」のオペレーティングシステムの適切なインストール手順に従って、AWS CLI バージョン 2 をインストールします。[の最新バージョンへのインストールまたは更新 AWS CLI](#)。
2. オペレーティングシステムの機能を使用して、を使用するなど、2 つの aws コマンドのいずれかに異なる名前のシンボリックリンクまたはエイリアスを作成します。aws2 AWS CLI バージョン 2 の場合。AWS CLI バージョン 2 のシンボリックリンクの例を次に示します。を置き換える **PATH** インストール場所を指定します。

Linux and macOS

Linux および macOS では、[シンボリックリンク](#)または[エイリアス](#)を使用できます。

```
$ alias aws2='PATH'
```


Windows command prompt

Windows の [DOSKEY](#)。

```
C:\> doskey aws2=PATH
```

AWS CLI バージョン 2 のアンインストール

このトピックでは、AWS Command Line Interface バージョン 2 (AWS CLI バージョン 2) をアンインストールする方法について説明します。

AWS CLI バージョン 2 のアンインストール手順 :

Linux

をアンインストールするには、のインストールに使用したのと同じ方法に従います AWS CLI。

Command line installer

AWS CLI バージョン 2 をアンインストールするには、次のコマンドを実行します。

1. シンボリックリンクとインストールパスを見つけます。

- `which` コマンドを使用して、シンボリックリンクを検索します。 `--bin-dir` パラメータで使用したパスが表示されます。

```
$ which aws
/usr/local/bin/aws
```

- `ls` コマンドを使用して、シンボリックリンクが指すディレクトリを検索します。これにより、 `--install-dir` パラメータで使用したパスが表示されます。

```
$ ls -l /usr/local/bin/aws
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/
local/aws-cli/v2/current/bin/aws
```

2. `--bin-dir` ディレクトリ内の 2 つのシンボリックリンクを削除します。ユーザーにこれらのディレクトリへの書き込みアクセス許可がある場合は、`sudo` を使用する必要はありません。

```
$ sudo rm /usr/local/bin/aws
$ sudo rm /usr/local/bin/aws_completer
```

3. `--install-dir` ディレクトリを削除します。このディレクトリへの書き込みアクセス許可がユーザーにある場合は、`sudo` を使用する必要はありません。

```
$ sudo rm -rf /usr/local/aws-cli
```

- (オプション) `.aws` フォルダ内の共有 AWS SDK情報と AWS CLI 設定情報を削除します。

Warning

これらの設定と認証情報の設定は、すべての AWS SDKs とで共有されます AWS CLI。このフォルダを削除すると、システム上に残っている AWS SDKsによってアクセスできなくなります。

`.aws` フォルダのデフォルトの場所はプラットフォームによって異なります。デフォルトではフォルダは `~/.aws/` にあります。ユーザーにこのディレクトリへの書き込み許可がある場合、`sudo` を使用する必要はありません。

```
$ sudo rm -rf ~/.aws/
```

Snap

AWS CLI では、の公式 AWS サポートバージョンを提供しています `snap`。 `snap` をインストールするために `sudo` を使用した場合は AWS CLI、次の手順を実行します。

- に対して次の `snap remove` コマンドを実行します AWS CLI。

```
$ snap remove aws-cli --classic
```

`sudo` をインストールしていた場合は AWS CLI、コマンドに追加する必要があります。

```
$ sudo snap remove aws-cli --classic
```

- すべてのファイルを削除するには、コマンドプロンプトウィンドウまたはコンピュータの再起動が必要になる場合があります。
- (オプション) `.aws` フォルダ内の共有 AWS SDK情報と AWS CLI 設定情報を削除します。

⚠ Warning

これらの設定と認証情報の設定は、すべての AWS SDKs とで共有されます AWS CLI。このフォルダを削除すると、システム上に残っている AWS SDKsによってアクセスできなくなります。

.aws フォルダのデフォルトの場所はプラットフォームによって異なります。デフォルトではフォルダは `~/.aws/` にあります。このディレクトリへの書き込みアクセス許可がある場合は、`sudo` を使用する必要はありません。

```
$ sudo rm -r ~/.aws/
```

macOS

AWS CLI バージョン 2 をアンインストールするには、インストールに使用したパスを置き換えて、次のコマンドを実行します。このコマンド例では、デフォルトのインストールパスを使用しています。

1. メインプログラムとコンプリータへのシンボリックリンクを含むフォルダを見つけます。

```
$ which aws
/usr/local/bin/aws
```

2. この情報を使用して次のコマンドを実行し、シンボリックリンクが指すインストールフォルダを見つけます。

```
$ ls -l /usr/local/bin/aws
lrwxrwxrwx 1 ec2-user ec2-user 49 Oct 22 09:49 /usr/local/bin/aws -> /usr/local/
aws-cli/aws
```

3. 最初のフォルダの 2 つのシンボリックリンクを削除します。ユーザーにこれらのフォルダに対する書き込みアクセス許可が既にある場合は、`sudo` を使用する必要はありません。

```
$ sudo rm /usr/local/bin/aws
$ sudo rm /usr/local/bin/aws_completer
```

4. メインインストールフォルダを削除します。sudo を使用して、/usr/local フォルダに対する書き込みアクセス許可を取得します。

```
$ sudo rm -rf /usr/local/aws-cli
```

5. (オプション) .awsフォルダ内の共有 AWS SDK情報と AWS CLI 設定情報を削除します。

Warning

これらの設定と認証情報設定は、すべての AWS SDKsと で共有されます AWS CLI。このフォルダを削除すると、システム上に残っている AWS SDKsによってアクセスできなくなります。

.aws フォルダのデフォルトの場所はプラットフォームによって異なります。デフォルトではフォルダは にあります。~/*.aws/*。ユーザーにこのディレクトリへの書き込み許可がある場合、 を使用する必要はありませんsudo。

```
$ sudo rm -rf ~/.aws/
```

Windows

1. 次のいずれかを実行して、[プログラムと機能] を開きます。
 - [コントロールパネル] を開き、[プログラムと機能] を選択します。
 - コマンドプロンプトを開き、次のコマンドを入力します。

```
C:\> appwiz.cpl
```

2. [AWS Command Line Interface] という名前のエントリを選択後、[アンインストール] を選択してアンインストーラを起動します。
3. をアンインストールすることを確認します AWS CLI。
4. (オプション) .awsフォルダ内の共有 AWS SDK情報と AWS CLI 設定情報を削除します。

⚠ Warning

これらの設定と認証情報設定は、すべての AWS SDKs とで共有されます AWS CLI。このフォルダを削除すると、システム上に残っている AWS SDKs によってアクセスできなくなります。

.aws フォルダのデフォルトの場所はプラットフォームによって異なります。デフォルトではフォルダは `%UserProfile%\aws` にあります。

```
$ rmdir %UserProfile%\aws
```

AWS CLI インストールおよびアンインストールエラーのトラブルシューティング

のインストールまたはアンインストール後に問題が発生した場合は AWS CLI、トラブルシューティングの手順 [エラーのトラブルシューティング](#) については、「」を参照してください。最も関連性の高いトラブルシューティングステップについては、「[the section called “コマンドが見つからないエラー”](#)」、「[the section called “aws --version コマンドが、インストールしたのとは異なるバージョンを返す”](#)」、および「[the section called “aws --version 「」 コマンドは、AWS CLI”](#)」を参照してください。

AWS CLI ユーザーガイドドキュメントの履歴

次の表は、2019 年 1 月以降の AWS Command Line Interface ユーザーガイドへの重要な追加点について説明しています。このドキュメントの更新に関する通知については、RSS フィードにサブスクライブできます。

変更	説明	日付
認証情報と認証に関する情報を更新しました。	認証情報と認証方法の説明と例を更新しました。これには、関連する開始方法ページと設定ページの更新が含まれます。このドキュメントの増大に対処するため、認証情報関連のトピックは新しい「 認証とアクセス認証情報 」セクションに移動しました。	2023 年 3 月 31 日
AWS IAM Identity Center の自動認証更新によるトークンプロバイダーの設定追加	SSO トークンプロバイダー設定を使用して AWS IAM Identity Center (IAM Identity Center) でユーザーを認証するように AWS CLI を設定する方法について説明します。SSO トークンプロバイダーは更新された認証トークンを自動的に取得できます。	2022 年 12 月 7 日
AWS CLI バージョン 2 の Amazon ECR Public の公式イメージをリリースしました	AWS CLI バージョン 2 でサポートされる Amazon ECR Public の公式イメージは、Linux、macOS、Windows 向けにリリースされています。	2022 年 11 月 18 日
AWS CLI V1 から V2 への移行ガイドを更新しました	重要な変更ガイドを拡張し、AWS CLI バージョン 1 から AWS CLI バージョン 2 へ	2022 年 5 月 13 日

	の移行手順を含めました。インストールの問題に役立つトラブルシューティングページの更新も含まれます。	
ソースから AWS CLI インストーラーをビルドする新しいプロセス。	サポートされているオペレーティングシステムに AWS CLI の最新リリースをインストールまたは更新する方法について説明します。	2022 年 2 月 17 日
AWS CLI V1 および V2 のコンテンツが、それぞれのガイドに分離されました	明瞭さと使いやすさのために、AWS CLI バージョン 1 および AWS CLI バージョン 2 のコンテンツが独自のガイドに分割されました。AWS CLI バージョン 1 については、 AWS CLI バージョン 1 ユーザーガイド	2021 年 11 月 2 日
AWS CLI エイリアス情報を追加	AWS CLI エイリアス情報を追加しました。エイリアスは、頻繁に使用するコマンドやスクリプトを短縮するために AWS Command Line Interface (AWS CLI) で作成できるショートカットです。	2021 年 3 月 11 日
フィルター出力情報を更新しました	フィルターの情報を更新し、独自のページに移動しました。	2021 年 2 月 1 日
ウィザードの情報を追加しました	AWS CLI バージョン 2 のウィザード情報を追加しました。	2020 年 11 月 20 日
自動プロンプトを更新しました	AWS CLI バージョン 2 の自動プロンプト情報を最新の機能で更新しました。	2020 年 11 月 10 日

Amazon S3 スクリプトの例を追加しました	Amazon S3 ライフサイクルのスクリプト例を追加しました。	2020 年 10 月 15 日
Amazon EC2 スクリプトの例を追加しました	Amazon EC2 インスタンスタイプのスクリプトの例を追加しました。	2020 年 10 月 15 日
再試行情報を追加しました	AWS CLI のリトライの機能と動作のリトライページを追加しました。	2020 年 9 月 17 日
サーバー側とクライアント側のページネーションページのページネーションページ	ページネーションの情報を更新し、単一のページに一元化しました。	2020 年 8 月 17 日
S3 コマンドページを更新しました	新しい例とリソースを含めて、高レベルの S3 コマンドページを更新しました。	2020 年 7 月 30 日
更新されたインストール情報	Linux、macOS、Windows のインストール、更新、およびアンインストール情報が更新されました。	2020 年 5 月 19 日
AWS CLI バージョン 2 のテキストファイルエンコーディングに関する情報を追加しました	デフォルトで、AWS CLI バージョン 2 はローカルと同じテキストファイルエンコーディングを使用します。環境変数を使用してテキストファイルのエンコードを設定できるようになりました。	2020 年 5 月 14 日
AWS CLI バージョン 2 の公式 Docker イメージがリリースされました	AWS CLI バージョン 2 の公式サポート Docker イメージは、すべての Linux、macOS、および Windows 向けにリリースされています。	2020 年 3 月 31 日

[AWS CLI バージョン 2 のクライアント側のページャーに関する情報を追加しました](#)

AWS CLI バージョン 2 はデフォルトで、すべてのクライアント側の出力にペーザープログラム `less` を使用しません。

2020 年 2 月 19 日

[AWS Command Line Interface \(AWS CLI\) バージョン 2 が正式にリリース](#)

AWS CLI バージョン 2 は一般公開されており、お客様のインストールに推奨されるバージョンです。

2020 年 2 月 10 日

[AWS CLI バージョン 2 用の macOS インストーラは、Apple パッケージのインストーラ .pkg ファイルになりました。](#)

AWS CLI バージョン 2 用の macOS インストーラは、シェルスクリプトを含む .zip ファイルから完全な macOS インストーラパッケージに更新されました。これにより、インストールがシンプルになり、最新の macOS リリースとの互換性が確保されます。

2020 年 2 月 3 日

[AWS CLI バージョン 2 での S3 および STS リージョンエンドポイントの改善されたデフォルト処理に関するコンテンツを追加しました。](#)

AWS CLI バージョン 2 はデフォルトで、Amazon S3 および AWS STS サービスのリクエストをグローバルエンドポイントではなく、現在設定されているリージョンのエンドポイントに転送するようになりました。

2020 年 1 月 13 日

[AWS CLI バージョン 2 のデベロッパープレビューリリース](#)

AWS CLI バージョン 2 のプレビューリリースを発表。バージョン 2 のインストール手順を追加しました。移行トピックを追加して、バージョン 1 と 2 の違いについて説明します。

2019 年 11 月 7 日

AWS CLI 名前付きプロファイルに AWS IAM Identity Center のサポートを追加しました	AWS CLI バージョン 2 は、IAM Identity Center に直接ログインし、後続の AWS コマンドでの使用のために AWS CLI 一時認証情報を取得できる名前付きプロファイルの作成のサポートを追加します。	2019 年 11 月 7 日
新しい MFA セクション	多要素認証とロールを使用して CLI にアクセスする方法について説明した新しいセクションを追加しました。	2019 年 5 月 3 日
「CLI の使用」セクションを更新	使用方法の説明と手順を大きく改善および追加しました。	2019 年 3 月 7 日
「CLI のインストール」セクションを更新	AWS CLI をインストールするための説明と手順の大幅な改善と追加を行いました。	2019 年 3 月 7 日
「CLI の設定」セクションを更新	AWS CLI を設定するための説明と手順の大幅な改善と追加を行いました。	2019 年 3 月 7 日